PointCluster: Deep Clustering of 3-D Point Clouds With Semantic Pseudo-Labeling

Xiu Liu, Xinxin Han, Huan Xia, Kang Li, Haochen Zhao, Jia Jia, Gang Zhen, Linzhi Su, Fengjun Zhao, and Xin Cao

***Abstract*— Point cloud classification is a fundamental problem in 3-D point cloud analysis. However, most existing methods are supervised, which requires costly and laborious annotations of large-scale point cloud datasets. This severely limits the practical applicability of point clouds. Therefore, exploring point cloud clustering methods, which can group point clouds into semantically meaningful clusters in an unsupervised manner, is of great importance. However, this remains a formidable challenge for humans. Here, we present PointCluster, a novel framework for deep clustering of 3-D point clouds. To enable accurate and reliable self-supervision for the clustering process, the framework introduces two semantic pseudo-labeling algorithms: prototype pseudo-labeling and reliable pseudo-labeling. We devise a three-step training process for the clustering network. First, we adopt a cross-modal representation learning approach to optimize the feature model. Second, we freeze the network parameters of the feature model and apply the prototype pseudo-labeling algorithm to optimize the clustering heads separately. Third, we use the reliable pseudo-labeling algorithm to jointly train the feature model and the clustering head in a semi-supervised manner, which enhances the overall clustering performance. The experimental results demonstrate that PointCluster achieves the state-of-the-art clustering results on public datasets such as ShapeNet. Moreover, our method narrows the gap between unsupervised point cloud clustering and supervised point cloud classification, offering a new perspective for the point cloud classification task.**

***Index Terms*— Point cloud clustering, self-supervised learning, unsupervised point cloud classification.**

I. Introduction

VARIOUS forms of 3-D data can be easily collected with the advancement of 3-D acquisition technology. These include meshes, point clouds, depth images, and others. These data formats provide a more realistic and natural representation of objects and scenes in 3-D space than 2-D images. As a consequence, point cloud processing has attracted considerable attention in various emerging fields, such as autonomous driving, robotics, and virtual reality.

Point cloud classification, a fundamental but critical step in many point cloud analysis applications, aims to assign predetermined semantic tags (such as aircraft, tables, and lights) to the cluttered point cloud. The rapid advancement of deep neural networks (DNNs) has facilitated a breakthrough in point cloud processing. The pioneer work, PointNet [1], directly applied neural networks to handle discrete and irregular point cloud data for classification and other tasks. A large amount of research on 3-D point cloud analysis using DNNs has emerged since then. Recent advances in neural networks specifically designed for raw 3-D point clouds have achieved considerable progress in various point cloud processing tasks [2], [3], [4], [5], [6], [7], [8]. However, these techniques rely heavily on large manually annotated datasets [9], [10], [11] to train the network, which poses significant challenges for subsequent applications of point clouds. The process of manually annotating datasets is often time-consuming, arduous, and error-prone. Moreover, the network models trained on manually labeled data may have limited generalization capacity. Therefore, it is essential to develop neural networks that can learn discriminative feature representations of point clouds without human supervision and cluster them into semantically meaningful clusters. This problem remains challenging for humans at present.

Motivated by the above goal, unsupervised representation learning for point clouds has drawn extensive attention recently. The approaches aim to learn robust and discriminative feature representations without the need for labeled samples. A wide range of self-supervised learning tasks, also known as pretext tasks, are proposed in [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], and [22] to achieve this. The learned feature representations can be transferred to diverse downstream tasks as prior or auxiliary information, thereby alleviating the time-consuming and laborious manual labeling work. The existing unsupervised representation learning approaches for point clouds can be broadly classified into three categories according to the type of pretext tasks: generation-based methods [16], [17], [18], [19], context-based methods [12], [13], [14], [15], and multiple modality-based methods [20], [21], [22]. Recent advances in unsupervised point cloud representation learning have exhibited remarkable efficacy. The unsupervised representation learning approach,

Manuscript received 28 May 2023; revised 26 September 2023 and 3 March 2024; accepted 20 April 2024. Date of publication 25 April 2024; date of current version 6 May 2024. This work was supported in part by the Key Research and Development Program of Shaanxi Province under Grant 2024SF-YBXM-681, Grant 2019GY215, and Grant 2021ZDLSF06-04; and in part by the National Natural Science Foundation of China under Grant 61701403 and Grant 61806164. *(Xiu Liu and Xinxin Han contributed equally to this work.) (Corresponding authors: Linzhi Su; Xin Cao.)*

Xiu Liu, Xinxin Han, Huan Xia, Kang Li, Linzhi Su, Fengjun Zhao, and Xin Cao are with the School of Information Science and Technology, Northwest University and National and Local Joint Engineering Research Center for Cultural Heritage Digitization, Xi'an, Shaanxi 710127, China (e-mail: sulinzhi029@163.com; caoxin918@hotmail.com).

Haochen Zhao, Jia Jia, and Gang Zhen are with the No. 1 Department of Conservation and Restoration, Shaanxi Institute for the Preservation of Cultural Heritage, Xi'an, Shaanxi 710075, China.

Digital Object Identifier 10.1109/TGRS.2024.3393911

however, is only the first part of the two-stage training pipeline. In the second stage, the downstream task still requires manual annotations for supervised training. Without ground-truth labels, downstream tasks cannot be accomplished.

To overcome the challenges mentioned above, point cloud clustering aims to automatically group 3-D point cloud samples into semantically meaningful clusters without relying on ground-truth labels. As a classic problem in machine learning, clustering serves as a vital preprocessing step for various applications, such as anomaly detection, robotic vision, and autonomous driving. Many traditional clustering methods, such as K-means++ [23], SC [24], and ac [25], have been widely studied and applied to data clustering. However, these methods depend on predefined distance metrics, which can be challenging to determine for complex point cloud datasets, leading to suboptimal performance. Deep clustering of 3-D point clouds remains largely unexplored to the best of our knowledge, with only a few related works. By assigning pseudo-labels to point cloud samples with clustering algorithms, some unsupervised representation learning methods [26], [27] integrate traditional clustering methods, such as K-means++ [23], to learn useful feature representations for downstream tasks. Therefore, there is an urgent need for a superior 3-D point cloud clustering approach that can automatically group point clouds into semantically meaningful clusters without labeled samples.

Deep clustering has drawn significant attention in the field of 2-D image analysis in recent years. Several methods [28], [29], [30], [56], [57] have demonstrated their effectiveness in grouping image datasets into meaningful clusters without the need for human annotations. One of the successful approaches is SPICE [30], which achieves significant improvement in the performance of image clustering. Inspired by this achievement, we present PointCluster, a novel framework for deep clustering of 3-D point clouds. Two crucial aspects must be considered for clustering: instance-level similarity and cluster-level difference. Thus, samples in the same cluster should be as similar as possible, while samples in different clusters should be clearly distinguished. The core idea of our approach is to use a pseudo-labeling algorithm to leverage both the instance-level similarity and the cluster-level difference to achieve accurate and reliable self-supervision of network training. As illustrated in Fig. 1, our method consists of two components: the feature model, which measures the instance-level similarity, and the clustering head, which identifies the cluster-level difference. We design a three-step training strategy for point cloud clustering.

1) We adopt a representation learning approach to optimize the feature model and extract semantically meaningful features.
2) After freezing the network parameters of the feature model trained in the previous step, we propose a prototype pseudo-labeling algorithm to consider both the instance-level similarity and the cluster-level difference and train the clustering head separately under the expectation–maximization (EM) framework to predict the cluster semantics.
3) To further enhance the clustering performance, we propose a reliable pseudo-labeling algorithm to filter out a subset of reliable pseudo-labels and jointly optimize the feature model and the clustering head in a semi-supervised manner.

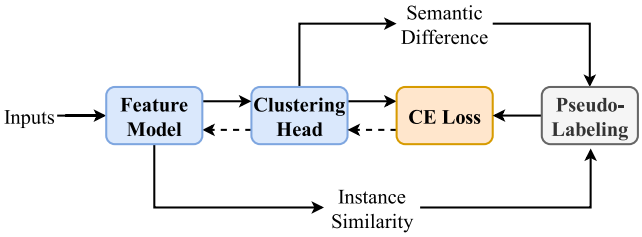

Fig. 1. Proposed PointCluster framework synergizes both the instance-level similarity and the cluster-level differences using the pseudo-labeling algorithm to train the clustering network.

Extensive experiments on public datasets demonstrate that our approach achieves the state-of-the-art clustering results and even outperforms some classical supervised point cloud learning methods. We summarize the main contributions of this article as follows.

1) We introduce PointCluster, a novel framework for deep clustering of 3-D point clouds, which achieves accurate and reliable self-supervision for clustering by synergizing the instance-level feature similarity and the cluster-level difference.
2) We propose two semantic pseudo-labeling algorithms: prototype pseudo-labeling and reliable pseudo-labeling. The prototype pseudo-labeling algorithm identifies prototypes for training the clustering head in an EM framework, and the reliable pseudo-labeling algorithm filters reliable samples for the joint training stage to further enhance the clustering performance.
3) Extensive experiments on various benchmark datasets demonstrate that our PointCluster outperforms existing state-of-the-art clustering methods and significantly narrows the performance gap between point cloud clustering and supervised point cloud classification.

II. Related Works

In this section, we provide a brief overview of the related research in two domains: supervised learning on 3-D point clouds and unsupervised representation learning.

A. Supervised Learning on 3-D Point Clouds

The 3-D point clouds are a distinctive data type that differs significantly from 2-D images. The irregularity and discreteness of 3-D point clouds in 3-D space pose challenges for learning point cloud representations and performing downstream tasks. PointNet [1], a pioneering work in this field, directly operates on raw 3-D point clouds without any preprocessing. Its network design employs shared multilayer perceptrons (MLPs) to learn features for each point independently and fuses these features via pooling operations

to learn the point cloud representation. However, it cannot capture the local structure of point clouds, which is essential. Since then, the field of 3-D point cloud analysis has undergone rapid development [4], [5], [7], [15], [34]. PAConv [6] is one of the most outstanding convolution-based methods, which is a general convolution operation for 3-D point cloud processing that handles irregular and unordered point clouds effectively. It constructs convolution kernels by dynamically assembling basic weight matrices, whose coefficients are adaptively learned based on point positions. It can be integrated into simple MLP-based point cloud pipelines without changing the network configuration and achieve highly competitive performance. Attention-based methods have also demonstrated excellent ability in exploring the local structure of point clouds [7], [17], [52]. Point Transformer v2 [52] proposes group vector attention, improved position encoding, and partition-based pooling to enhance point cloud learning, which overcomes the limitations of previous attention-based methods. A recent study [8] presents a new perspective that replaces network architectures with "complex" local geometric extractors with a pure residual MLP network equipped with a lightweight geometric affine module to capture the features of 3-D point clouds. It outperforms many advanced methods in accuracy (ACC) and boasts significant advantages in training speed. Although the above methods have achieved remarkable results in downstream tasks such as point cloud classification, they rely on large amounts of manually annotated datasets to train neural networks. In contrast to these methods, our research aims to explore an effective unsupervised point cloud clustering method.

B. Unsupervised Representation Learning

Owing to the high cost of large-scale manually annotated datasets, unsupervised representation learning has attracted increasing attention in recent years. Its aim is to learn robust and transferable feature representations from unlabeled data, reducing the dependence of downstream tasks on labels. A variety of unsupervised representation learning techniques for 3-D point clouds have been explored. Early unsupervised learning works can be typically divided into two categories: autoencoder (AE)-based methods [16], [37], [38] and generative adversarial network (GAN)-based methods [19], [39], [40]. These methods learn feature representations by accurately reconstructing input data with different network architectures. However, most of them do not utilize the local geometric information effectively, leading to suboptimal performance in downstream tasks such as classification. Recently, contrastive learning has emerged as an effective way to learn unsupervised representations [13], [14], [15], [22], [41], [53]. Contrastive learning generates positive and negative sample pairs for each instance. It then trains the encoder by using contrastive loss to maximize the similarity of the representations for positive pairs and minimize it for negative pairs. Liu et al. [42] propose a novel point discriminative learning strategy for unsupervised representation learning on 3-D point clouds. This strategy enforces the network to generate consistent features for points within the same local shape region and distinct features for points from different local shape regions or randomly sampled noise points. CrossPoint [22] devise a cross-modal contrastive learning approach to learn more general and transferable 3-D point cloud representations. It enables a 3-D–2-D correspondence of objects in the invariant space while promoting invariance to transformations in the point cloud modality. Some recent studies have begun to focus on the masked AE framework [50], [54], [55] by masking a part of a point cloud randomly and training the AE to reconstruct the masked part, accomplishing unsupervised pretraining of point clouds. Zhang et al. [54] propose a novel pretraining framework of multiscale masked AE, which enables hierarchical learning of 3-D point cloud feature representations. Point-M2AE generates powerful 3-D representations by encoding multiscale point clouds and reconstructing the masked coordinates based on a global-to-local upsampling scheme. Recent work [55] leverages rich 2-D knowledge learned from pretrained models to guide the learning of 3-D features from point cloud data. Specifically, the authors employ two image-to-point learning schemes in their work, namely, 2-D guided masking before the encoder and 2-D semantic reconstruction after the decoder, to enhance the quality of 3-D feature learning. With 2-D guidance, I2P-MAE learns excellent 3-D representations and reduces the demand for large-scale 3-D data. This work achieves state-of-the-art performance in 3-D representation learning. In fact, our proposed clustering strategy is related to unsupervised representation learning methods. To obtain discriminative feature representations for subsequent clustering processes, we use the contrastive learning paradigm to train the feature model during the initial training phase. Our framework is flexible and can accommodate any other representation learning method.

III. Method

In this work, we investigate a point cloud dataset $\mathcal{D} = \{P_i\}_{i=1}^{N}$ where each point cloud $P_i \in \mathbb{R}^{n\times 3}$. Our aim is to develop a novel unsupervised clustering network for 3-D point clouds, which can automatically cluster N point clouds in the dataset $\mathcal{D}$ into K categories without any human supervision. To achieve this, we propose PointCluster, a general framework for deep clustering of 3-D point clouds, as illustrated in Fig. 2. From a conceptual perspective, the framework consists of the following two components: a feature model that measures the instance-level similarity and a clustering head that identifies the cluster-level differences. Given an input point cloud sample P_i, the feature model F extracts the features $f_i = F(P_i; \theta_P)$, which the clustering head C maps to the probability distribution $p_i = C(f_i; \theta_C)$ over K categories. Here, θ_p and θ_C are the trainable parameters of F and C, respectively. Furthermore, we present two types of semantic pseudo-labeling algorithms, which facilitate the generated pseudo-labels to provide accurate and reliable self-supervision for the clustering process.

Specifically, the training process of our clustering framework involves three steps. To effectively exploit the shape features of the samples, we first construct positive and negative sample pairs and then train the feature model F using a novel cross-modal contrastive learning method. Subsequently, we freeze the network parameters of the feature model and train the clustering head C with our proposed prototype pseudo-labeling algorithm. Finally, we use our reliable

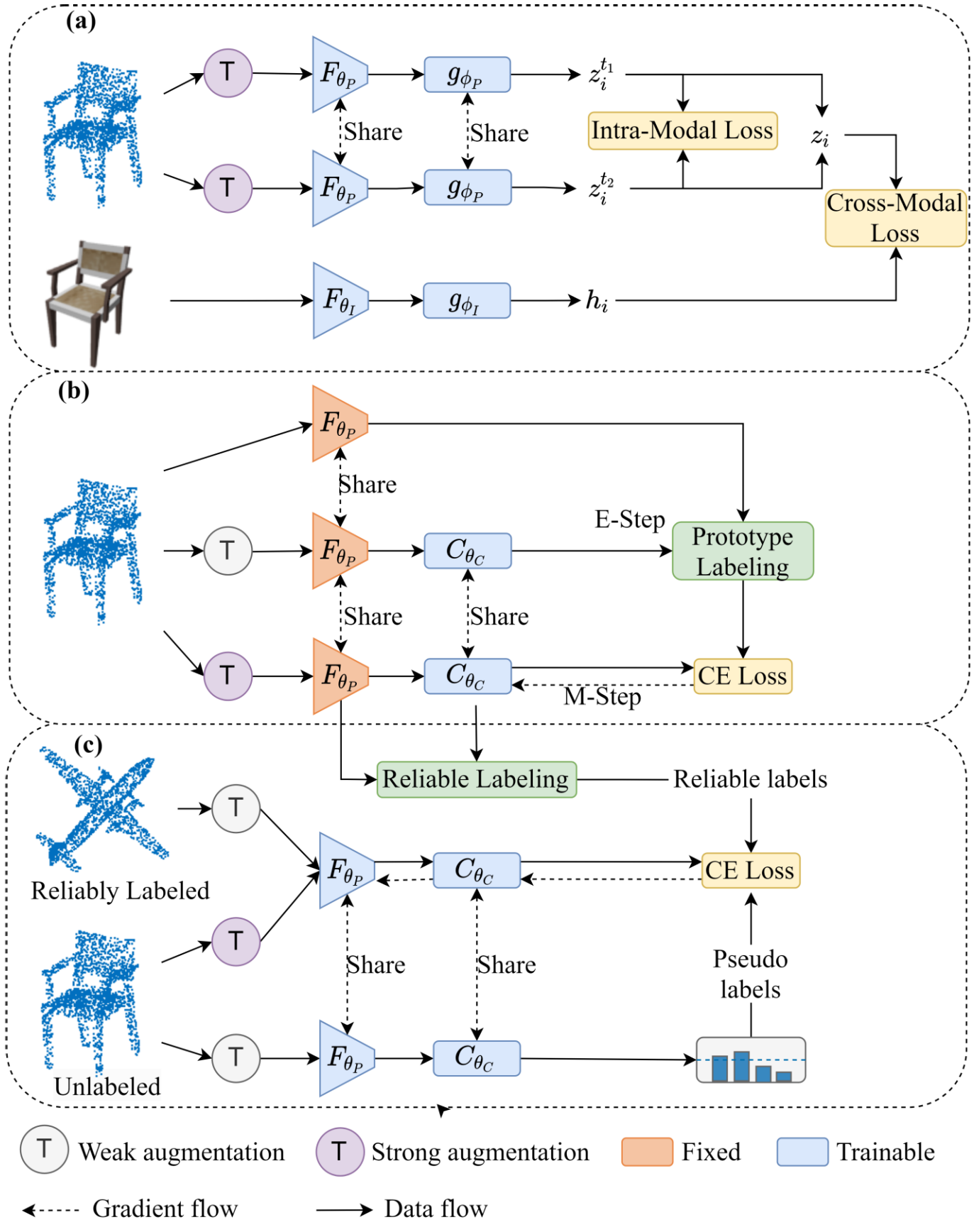

Fig. 2. Pipeline of our framework. (a) Train the feature model by using unsupervised representation learning approach CrossPoint [22]. (b) Train the clustering head while freezing the network parameters of the optimized feature model. (c) Jointly train the feature model and the clustering head in a semi-supervised manner.

pseudo-labeling algorithm to select reliable pseudo-labels and jointly optimize the whole network in a semi-supervised manner, which further enhances the clustering performance. In this section, we will provide a detailed description of each training stage in sequence.

A. Feature Model Training With Cross-Modal Contrastive Learning

To obtain a feature model that can accurately measure instance-level similarity, we adopt the cross-modal contrastive learning method CrossPoint [22] in this training stage, as shown in Fig. 2(a). This method fuses the intramodal and cross-modal contrastive learning objectives to learn representation features.

1) Intramodal Contrastive Learning on Point Clouds: Intramodal contrastive learning is accomplished by taking the transformation invariance of the point cloud as an efficient pretext task. For each point cloud in the dataset, we generate two augmented point clouds by applying different transformations to the same sample and regard them as a positive sample pair, while the rest are negative sample pairs. The objective of intramodal contrastive learning is to maximize the similarity between positive pairs and minimize the similarity between negative pairs.

Formally, given a 3-D point cloud P_i as input and a set of point cloud geometric transformations T, we randomly combine the transformations from T to generate two distinct transformations, t_1 and t_2. Then, we apply them to the point cloud P_i, producing two augmented point clouds, $P_i^{t_1}$ and $P_i^{t_2}$. The feature model F_{θ_P} maps the two point clouds to a feature space, and the projection head $g_{\emptyset_P}$ further projects the features to an invariant space $\mathbb{R}^d$, where instance-level contrastive learning is performed. The resulting projected vectors are denoted as $z_i^{t_1}$ and $z_i^{t_2}$, where $z_i^t = g_{\emptyset_P}(F_{\theta_P}(P_i^t))$. Here, the feature model F_{θ_P} uses DGCNN [3], and the projection head $g_{\emptyset_P}$ is a nonlinear MLP. The contrastive loss function employed in this approach is the NT-Xent loss from SimCLR [43]. For the positive sample pair of examples $z_i^{t_1}$ and $z_i^{t_2}$, the loss function can be computed as

$$\mathcal{L}(i,t_1,t_2) = -\log \frac{\exp\left(s\left(z_i^{t_2}, z_i^{t_2}\right)/\mathcal{T}\right)}{\sum_{\substack{k=1\\k\neq i}}^{M}\exp\left(s\left(z_i^{t_1}, z_k^{t1}\right)/\mathcal{T}\right)+\sum_{k=1}^{M}\exp\left(s\left(z_i^{t_1}, z_k^{t_2}\right)/\mathcal{T}\right)} \tag{1}$$

where M is the mini-batch size, $\mathcal{T}$ is the temperature parameter, and $s(.)$ denotes the cosine similarity function. The intramodal contrastive loss function $\mathcal{L}_{\text{imid}}$ for a mini-batch of point clouds can be expressed as

$$\mathcal{L}_{\text{imid}} = \frac{1}{2M}\sum_{i=1}^{M}\left[\mathcal{L}(i,t_1,t_2)+\mathcal{L}(i,t_2,t_1)\right]. \tag{2}$$

2) Cross-Modal Contrastive Learning on Point Clouds: By forcing 3-D point clouds to correlate to their rendered 2-D images, cross-modal contrastive learning facilitates effective representation learning on point clouds in an auxiliary manner. In addition to the 3-D point cloud P_i, we also use its corresponding rendered 2-D image I_i as another input modality. The image feature model F_{θ_I} is first applied to map I_i into a feature space, and then, the projection head $g_{\emptyset_I}$ is used to map the features into an invariance space. The projected vector h_i can be obtained by $h_i = g_{\emptyset_I}(F_{\theta_I}(I_i))$. We adopt the classic ResNet [44] as the image feature model F_{θ_I}. For better understanding of 3-D point clouds from the image pattern, we compute the mean of the projected vectors $z_i^{t_1}$ and $z_i^{t_2}$ to obtain the projected prototype vector z_i of P_i. The contrastive loss function $\mathcal{C}(i,z,h)$ for the positive pair of examples z_i and h_i is defined as follows:

$$\mathcal{C}(i,z,h) = -\log\left(\frac{\exp\left(s(z_i,h_i)/\mathcal{T}\right)}{\sum_{\substack{k=1\\k\neq i}}^{M}\exp\left(s(z_i,z_k)/\mathcal{T}\right)+\sum_{k=1}^{M}\exp\left(s(z_i,h_k)/\mathcal{T}\right)}\right). \tag{3}$$

As in the preceding formula, s, M, and $\mathcal{T}$ denote the same variables. We define the cross-modal loss function $\mathcal{L}_{\text{cmid}}$ for a mini-batch as follows:

$$\mathcal{L}_{\text{cmid}} = \frac{1}{2M}\sum_{i=1}^{M}\left[\mathcal{C}(i,z,h)+\mathcal{C}(i,h,z)\right]. \tag{4}$$

The final loss function $\mathcal{L}$ is obtained by simply adding $\mathcal{L}_{\text{imid}}$ and $\mathcal{L}_{\text{cmid}}$, which enhances the representation learning ability by incorporating the transformation invariance within the point cloud modality and the 2-D–3-D cross-modal feature correspondence.

It should be noted that the trained point cloud feature model F_{θ_P} will be transferred to the subsequent stage and used continuously. Moreover, the cross-modal contrast learning method adopted in this article is flexible and can be replaced by any other unsupervised representation learning approach.

B. Clustering Head Training With Prototype Pseudo-Labeling

In this training stage, we fix the network parameters of the feature model optimized in the previous stage and optimize the clustering head separately. Specifically, the input of the current stage comprises the point cloud dataset $\mathcal{D}$ and the feature model F_{θ_P} optimized in the previous stage, with the goal of optimizing the clustering head and predicting the clustering labels $\{y_i\}_{i=1}^N$ for point cloud samples. The clustering head C here is a simple several-layer MLP that maps the features f_i to the probabilities, $p_i = C(f_i;\theta_C)$, where $f_i = F(P_i;\theta_P)$. Under the supervised setting clustering heads can be optimized by minimizing the cross entropy (CE). However, since we do not have the ground-truth labels of point cloud samples, it is crucial to find ways to produce valuable supervisory information for training the clustering heads. To tackle this problem, we propose a prototype pseudo-labeling algorithm that alternatively estimates pseudo-labels for a batch of point cloud samples and optimizes the network parameters of the clustering head.

Specifically, we perform two steps iteratively until convergence under the EM expectation strategy: the expectation step calculates $\{y_i\}$ given the clustering model parameters θ_C and

the maximization step updates θ_C given $\{y_i\}$. As shown in Fig. 2(b), we design the training stage into three branches inspired by contrastive learning as follows.

1) The first branch takes the primitive point clouds as inputs and extracts the embedding features f_i using the feature model F_{θ_P}.
2) The second branch takes the weakly transformed point cloud samples as inputs, extracts their embedding features using the same feature model, and then maps the features to the probabilities p_i over K clusters, which are subsequently combined with f_i to compute the pseudo labels y_i by the prototype pseudo-labeling algorithm.
3) The final branch takes the strongly transformed point cloud samples as inputs and trains the clustering head C with supervision using the generated pseudo-labels.

The feature model F_{θ_P} and the clustering head C share weights across the three branches. We elaborate the training procedure in more detail next.

1) Prototype Pseudo-Labeling (E-Step): Given a batch of point cloud samples $\mathcal{D}_b$, the top branch first extracts the embedding features $\mathcal{F} = [f_1, f_2, \ldots, f_M]^T \in \mathbb{R}^{M\times D}$, where M is the batch size and D is the feature dimension. The middle branch calculates the probability distribution $\mathcal{P} = [p_1, p_2, \ldots, p_M]^T \in \mathbb{R}^{M\times K}$ over the corresponding K clusters for the weakly transformed samples $\alpha(\mathcal{D}_b)$, where α denotes the weak transformation function for the point cloud sample.

After obtaining $\mathcal{P}$ and $\mathcal{F}$, the top (M/K) confident samples and their corresponding embedding features are selected to estimate the cluster prototypes for each cluster. Then, the indices of these cluster prototypes are assigned to their neighboring samples as the pseudo labels. Formally, we take the kth cluster as an example to illustrate the top (M/K) confident point clouds for each cluster

$$\xi_k = \left\{ f_i | i \in \text{argtopk}\left(\mathcal{P}:,k,\frac{M}{K}\right)\ \forall i = 1,2,\ldots,M \right\} \quad (5)$$

where $\mathcal{P}:,k$ denotes the kth column of probability matrix $\mathcal{P}$ and argtopk$(\mathcal{P}:,k,(M/K))$ returns the indices of the top (M/K) confident samples in column $\mathcal{P}:,k$. Therefore, the cluster prototypes $\{\gamma_k\}_{k=1}^K$ in the feature space can be computed as follows:

$$\gamma_k = \frac{K}{M}\sum_{f_i\in\xi_k} f_i \quad \forall k = 1,2,\ldots,K. \quad (6)$$

Next, by computing the cosine similarity between the features f_i and the cluster center γ_k, we select the (M/K) samples closest to the cluster center γ_k as $\mathcal{D}^k$ and assign them the same pseudo label $y_i = k \forall P_i \in \mathcal{D}^k$. Moreover, we use overlapping assignments for pseudo labels, which means that a point cloud sample may correspond to more than one pseudo label. Consequently, we obtain a batch of point cloud samples with semantic pseudo labels as follows:

$$\mathcal{D}^s = \left\{(P_i, y_i) | \forall P_i \in \mathcal{D}^k, k = 1,2,\ldots,K\right\}. \quad (7)$$

A toy example of the prototype pseudo-labeling process is illustrated in Fig. 3. The input consists of the predicted probabilities of a batch of ten point cloud samples over three clusters. First, for each cluster, the top three confident samples are selected. Then, in the feature space, the cluster prototypes for each cluster are estimated based on the features of the selected samples. Finally, the top three samples closest to each cluster prototype are assigned the index of the corresponding prototype as pseudo-labels. We set the pseudo-labels of other unassigned samples to -1 and do not use them for training the clustering head.

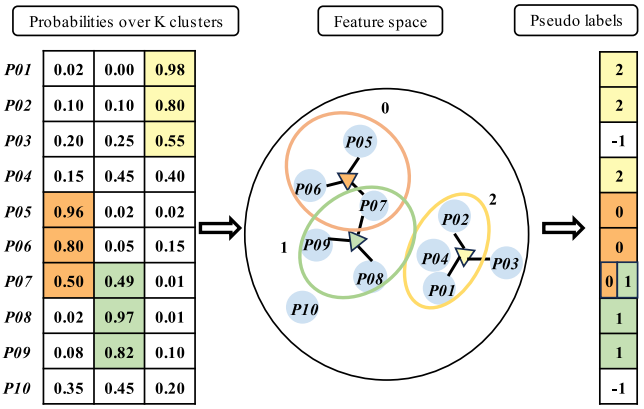

Fig. 3. Toy example of the prototype pseudo-labeling process. First, given the predicted probabilities of ten point cloud samples over three clusters as input, the top three confident samples are selected for each cluster and marked with different colors (orange, green, and yellow) in the figure to represent different clusters. Then, in the feature space, the cluster prototypes for each cluster (denoted by different colored triangles) are estimated based on the features of the selected samples. Finally, the top three samples closest to each cluster prototype (samples in the same ellipse) are assigned the index of the corresponding prototype as pseudo-labels. The pseudo-labels of other unlabeled samples are -1 and will not be used for training the clustering head.

2) Clustering Head Training (M-Step): Given the labeled point cloud samples $\mathcal{D}^s$, we can optimize the clustering heads under supervision. Specifically, we generate the strongly transformed version $\beta(\mathcal{D}^s)$ of these point clouds, where β denotes the strong transformation function for point clouds. Then, we compute the corresponding probabilities over the K clusters and train the clustering head C using the CE loss function as follows:

$$\mathcal{L}_{\text{clu}} = \frac{1}{M}\sum_{i=1}^{M}\mathcal{L}_{\text{ce}}\left(y_i, p_i'\right) \quad (8)$$

where $p_i' = \text{softmax}(p_i)$, $p_i = C(F(\beta(P_i);\theta_P);\theta_C)$, and $\mathcal{L}_{\text{ce}}$ denotes the CE loss function. The above process for training the clustering head is summarized in Algorithm 1.

Furthermore, we explain some settings in the above process. On the one hand, we use double Softmax functions in (8) to reduce the update speed of network parameters because the initial pseudo-labels generated are not accurate enough. p_i is also the output of a Softmax function. On the other hand, the clustering head exhibits a high degree of randomness in its initialization parameters, which may degrade the clustering performance. To address this issue, we simultaneously train multiple clustering heads independently. Consequently, we fix the network parameters of the feature model and train multiple lightweight clustering heads with low computational cost in the current training stage. The clustering head with the minimum $\mathcal{L}_{\text{clu}}$ loss over the whole dataset is selected as the best for point cloud clustering.

Algorithm 1 Clustering Head Training

Input: $\mathcal{D} = \{P_i\}_{i=1}^{N}$, F_{θ_P}, K, M, T, α, β.
Initialization: Keep the network parameters F_{θ_P} fixed, set $t = 0$, and initialize θ_C.
while $t < T$ **do**
  for $b = 1, 2, \ldots, \lfloor \frac{N}{M} \rfloor$ **do**
    E-step:
    Take M samples from $\mathcal{D}$ as $\mathcal{D}_b$;
    Extract representation features $\mathcal{F} = F(\mathcal{D}_b; \theta_P)$
    Compute probabilities $\mathcal{P} = C(F(\alpha(\mathcal{D}_b); \theta_P); \theta_C)$;
    Construct labeled point cloud set $\mathcal{D}^s$ with (5), (6), and (7);
    M-step:
    Compute probabilities $\mathcal{P} = C(F(\beta(\mathcal{D}_b); \theta_P); \theta_C)$
    Optimize θ_C by minimizing (8);
  end
$t = t + 1$
end
Select the optimal clustering head with the minimum loss;
foreach $P_i \in \mathcal{D}$ **do**
  $p_i = C(F(P_i; \theta_P); \theta_C)$;
  $y_i = argmax_k(pi)$;
end
Output: Cluster label y_i of $P_i \in \mathcal{D}$.

C. Joint Training With Reliable Pseudo-Labeling

In the previous two stages, we separately optimize the feature model and the clustering head. However, this strategy is often suboptimal, despite that it achieves effective clustering results. Therefore, we optimize the two components jointly in this stage. To address the challenge of the pseudo labels obtained in the previous step being not entirely accurate, we propose a reliable pseudo-labeling algorithm to select the pseudo labels with high confidence as supervision information for further enhancing the clustering performance.

1) Reliable Pseudo-Labeling: We assume that a sample's pseudo label is reliable if it agrees with the pseudo labels of many neighboring samples in the embedding space. In detail, given the embedding features and the pseudo labels $\{(P_i, f_i, y_i)\}_{i=1}^{N}$ obtained in Section III-B, we compute the cosine similarity between representation features and select N_s nearest samples for each point cloud P_i. We represent the labels of these N_s nearest samples by ς_i and then compute the consistency ratio r_i of the point cloud P_i as follows:

$$r_i = \frac{1}{N_s} \sum_{y \in \varsigma_i} \mathbb{L}(y = y_i) \tag{9}$$

where $\mathbb{L}(y = y_i)$ is one when y equals y_i and zero otherwise. If the consistency ratio r_i of P_i exceeds a predefined threshold λ, the sample (P_i, y_i) is considered as reliably labeled, and otherwise, the corresponding pseudo label is ignored. Through this strategy, we obtain the following point cloud dataset with reliable pseudo-labels as follows:

$$\mathcal{D}^r = \{(P_i, y_i) | r_i > \lambda \ \forall i = 1, 2, \ldots, N\}. \tag{10}$$

2) Joint Training: After obtaining the sample set $\mathcal{D}^r$ with reliable pseudo-labels, the unsupervised clustering task can be converted into a semi-supervised learning paradigm. In this stage, we use a baseline semi-supervised learning method [45] with the DGCNN backbone. On the one hand, we train the feature model and the clustering head by using the samples with reliable labels. On the other hand, different transformations of the same sample should produce the consistent predictions. To this end, as shown in Fig. 2(c), we use the confidently predicted label of the weakly transformed point cloud as the pseudo-label for its strongly transformed version to further optimize the network. Formally, the consistency pseudo label y_j^u of an unlabeled point cloud P_j is computed as

$$y_j^u = \begin{cases} \text{argmax}(p_j), & \text{if } \max(p_j) \geq \eta \\ -1, & \text{otherwise} \end{cases} \tag{11}$$

where $p_j = C(F(\alpha(P_j); \theta_P); \theta_C)$ and η denotes the confidence threshold. During this stage, the entire network is trained with the following loss function:

$$\begin{aligned} &\mathcal{L}_{\text{joint}} \\ &= \frac{1}{L} \sum_{i=1}^{L} \mathcal{L}_{\text{ce}}(y_i, C(F(\alpha(P_i); \theta_P); \theta_C) \\ &\quad + \frac{1}{U} \sum_{j=1}^{U} \mathbb{L}(y_j^u \geq 0) \mathcal{L}_{\text{ce}}(y_j^u, C(F(\beta(P_j); \theta_P); \theta_C)) \end{aligned} \tag{12}$$

where the samples in the first item are from the sample subset $\mathcal{D}^r$ with reliable pseudo-labels. The samples in the second item are from the whole dataset $\mathcal{D}$. L and U denote the number of reliably labeled and unlabeled samples within a batch, respectively. Note that any other effective semi-supervised learning algorithm can be applied in this work.

The entire process of our approach is summarized above. In summary, we propose a general point cloud clustering framework PointCluster that completes the clustering task by gradually optimizing the feature model, the clustering head, and the entire network end-to-end.

IV. Experiments and Results

A. Datasets and Evaluation Metrics

During the training stage, we follow the premise of balanced datasets with roughly equal sample quantities per class, as commonly done in 2-D image clustering methods [56], [57]. Furthermore, we train and evaluate the network proposed in this article using the entire dataset, without partitioning it into separate training and testing datasets. It should be noted that the testing dataset is not required to follow the balance assumption, and we have the option to perform the training and testing stage on distinct training and testing datasets. To create the balanced datasets and conduct extensive clustering experiments, we collect ten categories of point clouds (with the most point cloud samples) from two widely used benchmark datasets: ShapeNet [10] and ModelNet40 [9]. Table I summarizes the necessary details for each of our balanced point cloud datasets. Each point cloud sample is reserved with 2048 points as input, and only the 3-D coordinate information of sampling points is used in the experiments. In the first training stage, the feature model is pretrained only

TABLE I
NECESSARY DETAILS OF SELECTED DATASETS

Dataset	Number	Classes(K)
ShapeNet	14890	10
ModelNet40	4350	10

based on the ShapeNet [10] datasets and their corresponding rendered RGB images from [22].

This study adopts three standard clustering performance metrics reported in [56], [57], [58], and [59] to evaluate the performance of the clustering method: clustering ACC [46], normalized mutual information (NMI) [47], and adjusted rand index (ARI) [48]. These values range from 0 to 1, and more precisely, higher scores of these indicators imply more reliable clustering results.

B. Implementation Details

We adopt the general DGCNN as our feature extraction model throughout the training process for a fair comparison with alternative approaches. The dimension of the learned representation features is 2048. The framework constructs the clustering head as a four-layer MLP with dimensions [2048, 512, 256, K], where the dimension K of the output layer denotes the number of clusters. The number of clusters is predefined as the number of categories in the target dataset in this study. We devise our weak augmentation strategy by combining random cropping, translation, and normalization sequentially. For strong augmentation, we combine random cutout, rotation, scaling, jitter, and other effects.

For the first training stage, we follow the training settings in [22]. Specifically, we use the Adam optimizer with a weight decay of 1×10^{-4} and an initial learning rate of 1×10^{-3}, cosine annealing [49] as a learning rate scheduler. We train the model for 250 epochs. For the second training stage, we also employ the Adam optimizer with a constant learning rate of 0.001 and a batch size M of 16. We train eight clustering heads independently and simultaneously and select the one with the minimum loss as the best clustering head. We set N_s to 100 and λ to 0.9 to select reliable pseudo labels. In the joint training stage, we use an SGD optimizer with a momentum of 0.9 and an initial learning rate of $1 \times e^{-4}$. The batch size is 16, consisting of eight reliably labeled samples and eight unlabeled samples. The confidence threshold η is 0.95, consistent with the setting in [45].

C. Comparison of Clustering Performance

Since deep clustering of 3-D point clouds is scarcely studied, we compare our approach with several traditional clustering algorithms and other closely related methods. Specifically, we first use traditional clustering algorithms, including K-means++ [23], SC [24], and AC [25], to contrast with our approach. In addition, we apply several unsupervised representation learning methods, including STRL [14], Point-MAE [50], Point-M2AE [54], and I2P-MAE [55], to learn the representation features of point clouds in target datasets; then, we cluster the point clouds using K-means++ as a postprocessing step to obtain clustering results. We also compare our approach with these representation-based clustering methods. Furthermore, to demonstrate how the joint training stage enhances the clustering performance, we denote the PointCluster without the joint training phase as PointCluster*, where the superscript * implies separate training.

Table II reports the quantitative clustering results on the whole dataset using the above clustering methods. The results show that our PointCluster significantly outperforms the other approaches on all three evaluation metrics. Moreover, several trends can be observed in Table II. First, compared to traditional clustering algorithms (i.e., [23], [25]), representation-based clustering methods(i.e., [38], [14]) perform better, which indicates that representation learning plays a vital role in point cloud clustering. Second, even though increasingly advanced unsupervised representation learning algorithms can learn excellent representation features, the clustering results obtained by applying K-means++ subsequently are significantly worse than our approach. Our proposed learning-based point cloud clustering framework can group point clouds into correct clusters more accurately based on representation features than the traditional clustering algorithm. Our PointCluster* performs better than the other methods, even without the third joint training stage. All of these results demonstrate the efficacy of our approach in deep clustering of 3-D point clouds.

Table III provides more detailed comparison results on the ShapeNet and ModelNet40 datasets. For a fair comparison, we compare our method with the classical supervised learning method DGCNN, which is used as the backbone network in the proposed framework. It is clear that the clustering ACC of our PointCluster is slightly lower than that of the supervised method DGCNN, with only a 2% (93.2% versus 95.1%) ACC gap on ShapeNet and 1.5% (97.0% versus 98.5%) on ModelNet40. Therefore, our method significantly narrows the gap between unsupervised clustering and supervised classification in the point cloud domain. On the other hand, unlike k-means++, which infers the cluster labels from cluster centers, our approach adopts nonlinear clustering heads to predict clustering labels after learning representation features. It can be seen that our PointCluster outperforms the representation-based clustering algorithm CrossPoint + Kmeans++ on all three standard metrics, which demonstrates the superiority of our proposed clustering framework.

D. Visualization

1) Visualization of Semantic Clusters: We visualize the semantic clusters on ShapeNet learned by our PointCluster. We randomly sample seven point clouds from each of the ten clusters as presented in Fig. 4. The point clouds in each column are assigned to the same cluster. Samples with red borders are classified incorrectly and should belong to different clusters. For example, the last point cloud in the second column should be in the watercraft category, but it is categorized as a car. In addition, some categories, such as table, chair, and bench, are easily confused. In the chair category, the fifth point cloud should be categorized as a bench. In the table category,

TABLE II
QUANTITATIVE COMPARISON WITH COMPETITIVE POINT CLOUD CLUSTERING METHODS

Method	ShapeNet			ModelNet40		
	ACC	NMI	ARI	ACC	NMI	ARI
K-means++	0.5039	0.4704	0.3099	0.1324	0.019	0.003
SC	0.2975	0.3170	0.1001	0.1391	0.0316	0.0061
AC	0.5776	0.5144	0.3698	0.1007	0.0267	0.0033
STRL	0.7133	0.6755	0.5483	0.8856	0.8406	0.8025
Point-MAE	0.7222	0.6352	0.5292	0.7713	0.7874	0.6843
Point-M2AE	0.7793	0.7054	0.6038	0.8179	0.8059	0.7356
I2P-MAE	0.7952	0.7328	0.6779	0.8528	0.8343	0.8367
PointCluster*	0.9137	0.8324	0.8230	0.9106	0.8485	0.8187
PointCluster	**0.9317**	**0.8675**	**0.8586**	**0.9698**	**0.9495**	**0.9386**

TABLE III
MORE DETAILED COMPARISON RESULTS

Method	ShapeNet			ModelNet40		
	ACC	NMI	ARI	ACC	NMI	ARI
Supervised	0.9514	0.8865	0.9053	0.9854	0.9782	0.9839
CrossPoint +K-means++	0.7383	0.6864	0.5986	0.7622	0.8025	0.7048
PointCluster*	0.9137	0.8324	0.8230	0.9106	0.8485	0.8187
PointCluster	**0.9317**	**0.8675**	**0.8586**	**0.9698**	**0.9495**	**0.9386**

TABLE IV
ABLATION STUDIES OF POINTCLUSTER* ON THE SHAPENET DATASET

Variants	ACC	NMI	ARI
Non-overlap	0.8905	0.8059	0.7978
Joint-SH	0.7257	0.6417	0.5586
Joint-MH	0.8014	0.7015	0.6381
CE	0.6977	0.6454	0.5539
TCE	0.7320	0.6846	0.5973
PointCluster*	**0.9137**	**0.8324**	**0.8230**

the last point cloud should also be categorized as a bench. Overall, we observe that for ShapeNet, the cluster assignment acquired by our method mostly matches natural clusters The visual results demonstrate that our method learns semantically meaningful clusters and performs effectively.

2) Visualization of the Representation Features: To ensure a fair comparison, we use t-SNE to visualize the representation features of point cloud samples learned by the clustering methods in Table II. Fig. 5 shows the visualization results. It is clear that all the compared methods are able to provide good discrimination for the majority of classes. However, the class boundaries for some classes are not precise or compact. Overall, we did not observe a very significant difference in the quality of the representation features learned by these different methods. As illustrated in Table II, our proposed method exhibits superior clustering performance. This suggests that, besides learning well-separated representation features, accurate cluster assignments are crucial for clustering performance. Our novel clustering framework outperforms traditional post-processing steps such as K-means++ in grouping point clouds into correct clusters accurately. This further confirms the effectiveness of our deep clustering method for point clouds.

E. Empirical Analysis

In this section, we conduct a series of ablation studies to evaluate the effectiveness of various components and settings in our framework. All experiments in this section are performed on the ShapeNet dataset.

1) Effectiveness of Overlap Assignment: As shown in Table IV, we replace the original settings in the PointCluster* with each item individually for experimental comparison. We first evaluate the effect of overlap and nonoverlap assignment on the clustering performance of PointCluster* when assigning pseudo-labels in the second stage. As demonstrated in Table IV, overlap assignment outperforms nonoverlap assignment. When the same sample is close to multiple clustering centers, forcing only one pseudo-label to be assigned to the instance by nonoverlap allocation may introduce additional local inconsistency, which hinders model training.

2) Effectiveness of Three-Step Training Strategy: During the second training stage, we freeze the network parameters of the feature model and only optimize the clustering heads independently. To study the effect of our three-step training strategy, we propose two alternative training approaches for comparison, jointly training the feature model and a single cluster head (Joint-SH) and jointly training the feature model and multiple cluster heads (Joint-MH). As illustrated in Table IV, both of these two training strategies lead to a significant drop in clustering performance. At the beginning of training, our clustering head cannot make relatively correct predictions, and the incorrect labels deteriorate the feature model, resulting in lower quality representation features. The clustering head is even less likely to make accurate predictions, entering a vicious cycle. As a result, it is crucial to optimize the clustering head independently while freezing the network parameters of the feature model during this training stage.

3) Effectiveness of Double Softmax Functions: To evaluate the effect of applying the double Softmax functions before calculating the CE loss in (8), we conduct experiments with

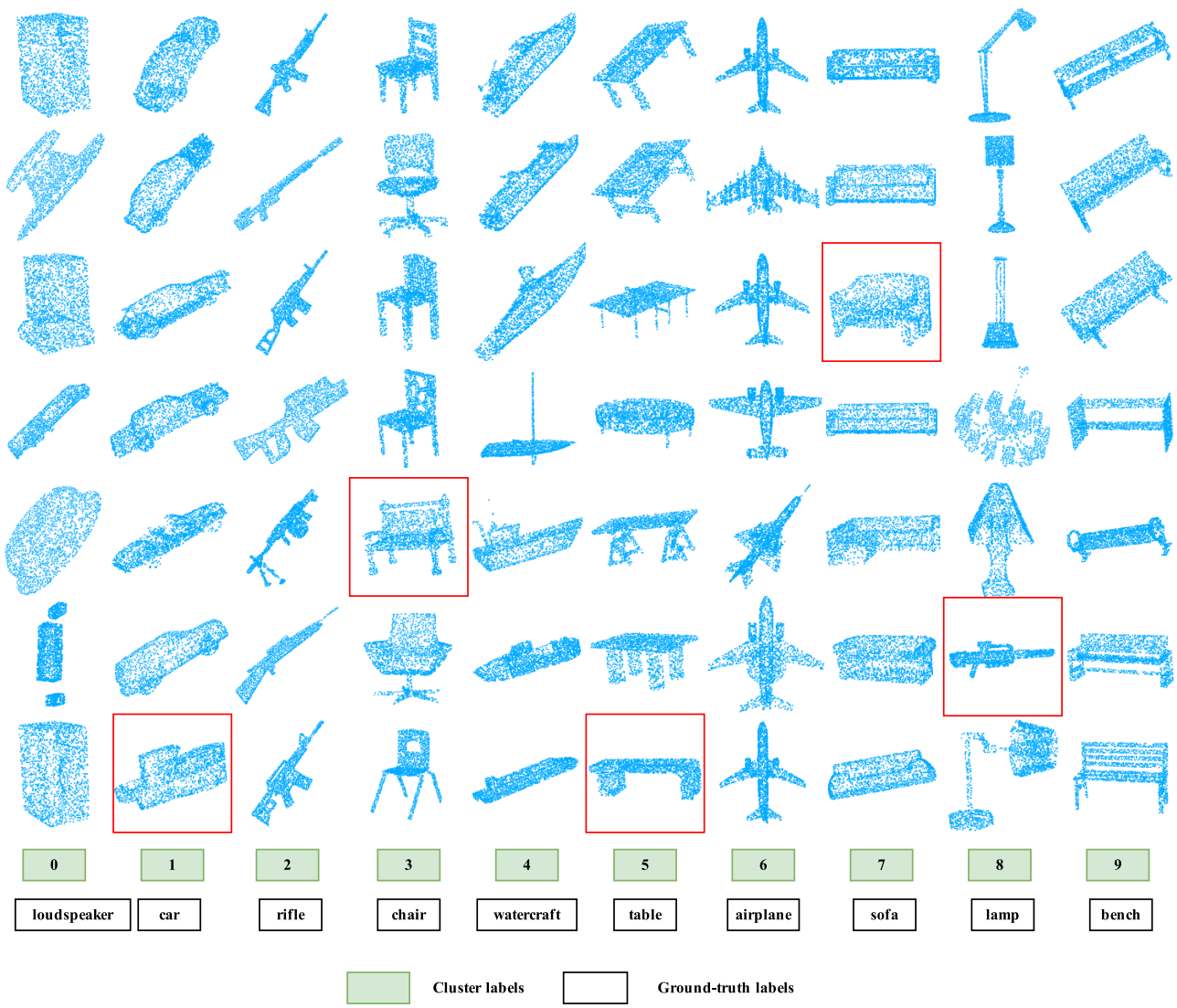

Fig. 4. Visualization of learned semantic clusters on ShapeNet. We randomly sample seven point clouds from all ten clusters and show them above. The point clouds in each col are assigned to the same cluster, while samples in red borders are categorized incorrectly and should belong to different clusters.

standard CE and TCE [51] loss with temperature parameters. We conduct experiments at different temperature values for TCE and find that the best performance is achieved at 0.2. As shown in Table IV, compared to CE and TCE, applying double Softmax functions in the method produces the best clustering performance.

4) Effect of Data Augmentation: We evaluate the effects of different data augmentation strategies on PointCluster*, as shown in Table V. Specifically, augment1 and augment2 refer to data augmentation of the middle and bottom branches in Fig. 2(b). The results show that the clustering performance is the best when the middle branch uses weak augmentation, and the bottom branch uses strong augmentation. This can be explained by the fact that the middle branch's prediction results are employed to assign pseudo-labels, which may introduce incorrect labels if a strong augmentation is applied. Moreover, our PointCluster* performs better when the bottom branch applies strong augmentation, as the feature model is encouraged to provide consistent predictions for richer augmentations. Overall, the data augmentation strategy here has a small impact on the clustering performance because the feature model is already equipped with the transformation invariance ability after the pretraining in the first stage.

TABLE V
QUANTITATIVE COMPARISON OF POINTCLUSTER* WITH DIFFERENT DATA AUGMENTATION STRATEGIES ON THE SHAPENET DATASET

Augment1	Augment2	ACC	NMI	ARI
Weak	Weak	0.9005	0.8150	0.8079
Strong	Weak	0.8592	0.8276	0.7790
Strong	Strong	0.9014	0.8035	0.7938
Weak	**Strong**	**0.9137**	**0.8324**	**0.8230**

5) Effectiveness of Optimal Clustering Head Selection: In the second training stage, we independently optimize several clustering heads in parallel to choose the best clustering head for point cloud clustering. However, the unsupervised setting does not provide any manual annotations to determine the superior clustering head. Therefore, we select the clustering head that has the lowest clustering loss on the entire dataset

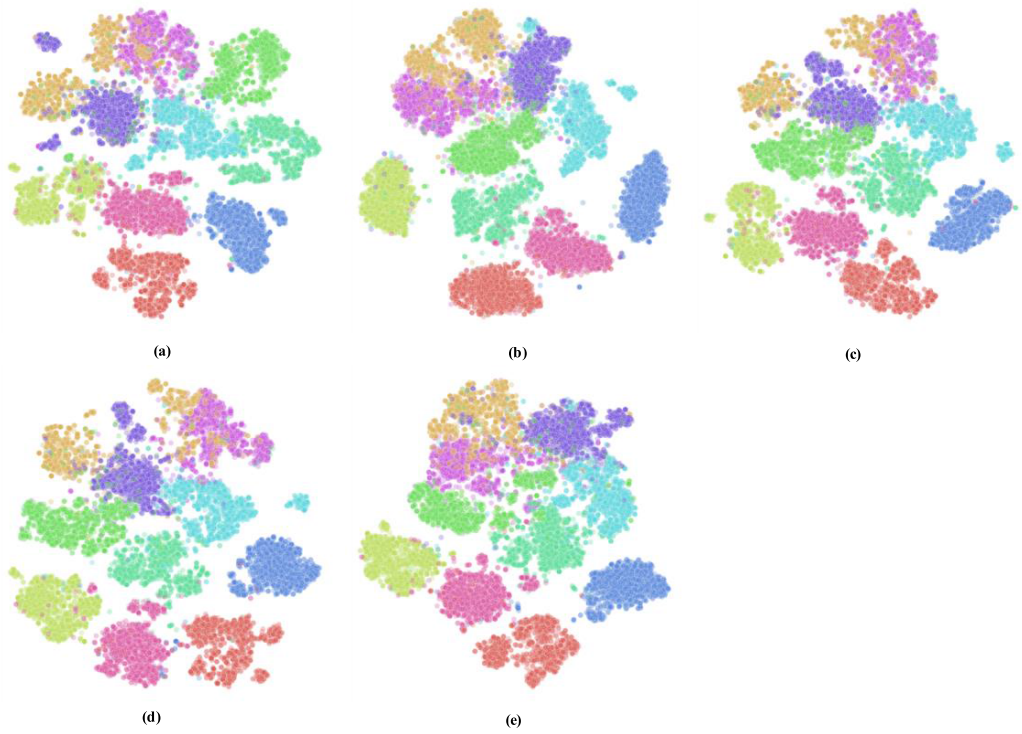

Fig. 5. Visualization of the representation features on ShapeNet dataset learned by different clustering methods. (a) STRL. (b) Point-MAE. (c) Point-M2AE. (d) I2P-MAE. (e) PointCluster*.

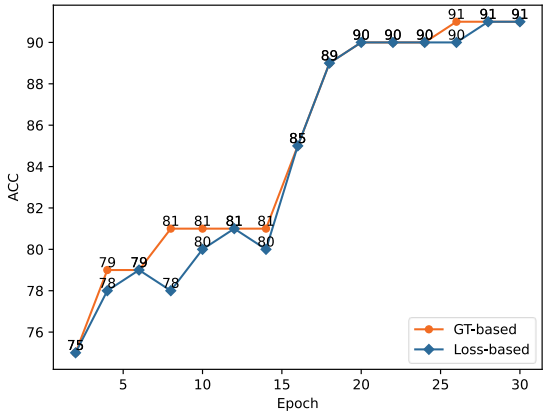

Fig. 6. Clustering head selection. The blue diamond marks the selected best clustering head with the minimum clustering loss, and the orange circle denotes the best clustering head evaluated with the ground truth.

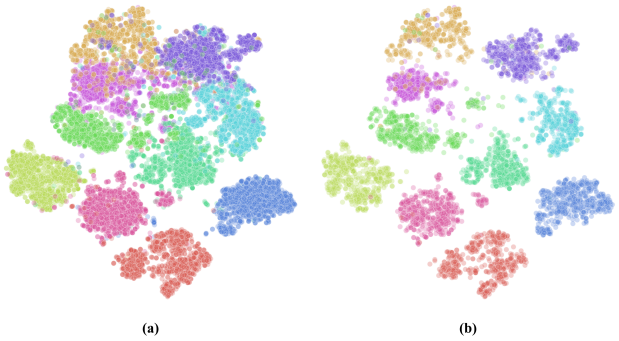

Fig. 7. Visualization of the representation features of point cloud samples in ShapeNet. (a) ACC of all samples is 0.913 and the ACC of the jointly trained model using all pseudo-labeled samples is 0.908. (b) ACC of the selected reliable samples is 0.979 and the ACC of the jointly trained model using the reliable samples is 0.931.

as the best one. Lower clustering loss implies better clustering performance. As shown in Fig. 6, the best clustering heads selected by ground-truth labels are remarkably close to those chosen by clustering losses, which proves the validity of the loss metric we adopted. By doing so, the poorly performing clustering heads can be filtered out. As a result, our PointCluster has a lower standard deviation and is more robust to unstable clustering performance caused by clustering head initialization.

6) Effectiveness of Reliable Pseudo-Labeling: We use the reliable pseudo-labeling algorithm to select the reliable subset of samples before the joint training stage. To demonstrate the necessity of this algorithm, we use t-SNE to visualize the representation features of point cloud samples in the ShapeNet dataset, as shown in Fig. 7. Fig. 7(a) shows the prediction results of PointCluster* for all point clouds, achieving an ACC of 91.3%. Several locally inconsistent samples can be seen in this figure. If we use these samples directly for joint training, the ACC will not be boosted. Fig. 7(b) shows the reliable samples selected by our reliable pseudo-labeling algorithm, which have almost no locally inconsistent samples and a clustering ACC of 97.9%. Using reliable samples for joint training, the ACC of PointCluster increases to 93.1%.

7) Hyperparameter Analysis: To explore the effect of different hyperparameters on our method, we perform ablation studies on two key hyperparameters, λ and N_s, which are both used to select reliable pseudo-labels. First, we set N_s to 100 and adjust λ from 0.6 to 0.95. Table VI shows the evaluation results, where Num and ACC_{SEL} represent the number and ACC of selected reliable labels, respectively.

TABLE VI
ABLATION STUDIES OF PARAMETER λ ON THE SHAPENET DATASET

λ	Num	ACC_{SEL}	ACC	NMI	ARI
0.6	9480	0.9499	0.9203	0.8445	0.8424
0.7	7950	0.9563	0.9237	0.8487	0.8426
0.8	6320	0.9705	0.9285	0.8549	0.8482
0.9	5280	0.9792	**0.9317**	**0.8675**	**0.8586**
0.95	2680	0.9876	0.9222	0.8504	0.8391

TABLE VII
ABLATION STUDIES OF PARAMETER N_s ON THE SHAPENET DATASET

N_s	Num	ACC_{SEL}	ACC	NMI	ARI
10	8640	0.9658	0.9229	0.8360	0.8210
50	6040	0.9757	0.9286	0.8478	0.8443
100	5280	0.9792	**0.9317**	**0.8675**	**0.8586**
200	2310	0.9904	0.9232	0.8449	0.8389

The table indicates that as λ decreases, more pseudo-labels are selected through the reliable pseudo-labeling algorithm, but the ACC of these pseudo-labels also declines. Conversely, as λ increases, the opposite trend is observed. The results indicate that selecting pseudo-labels with high ACC is crucial for the subsequent joint training stage. When the pseudo-labels have low ACC (e.g., when $\lambda = 0.6$ or 0.7), the performance of our method deteriorates, even with a large number of pseudo-labels. This is because incorrect pseudo-label assignments interfere with network training. When λ is 0.9, our method achieves the best performance, which implies that the quantity and quality of pseudo-labels are optimally balanced. Then, we set λ to 0.9 and conduct experiments with different values of N_s, including 10, 50, 100, 200, and 400. The results are shown in Table VII. The table reveals that as N_s decreases, more pseudo-labels are selected through the reliable pseudo-labeling algorithm, but their ACC also declines. Conversely, as N_s increases, the opposite trend is observed. Similarly, when N_s is 100, the quantity and quality of pseudo-labels are optimally balanced, and our method achieves the best performance.

V. DISCUSSION

In this work, our proposed PointCluster outperforms prior point cloud clustering methods by large margins and significantly narrows the performance gap between unsupervised clustering and supervised classification in the 3-D point cloud domain. Despite its effectiveness, our PointCluster is not without some limitations. First, current deep clustering algorithms often require the knowledge of the cluster number K. However, in practice, we may not know prior information about the target dataset, such as the number of categories. This makes it difficult to provide an appropriate clustering number K for subsequent network training. Therefore, how to automatically group all samples in the target dataset into appropriate clusters when the clustering number K is unknown remains an open challenge. Second, our proposed method imposes a constraint condition. Similar to deep image clustering algorithms, it assumes that the training dataset is roughly balanced, meaning that the number of samples in each category is comparable. However, the point cloud datasets from the real world may not satisfy this requirement. Therefore, more point cloud clustering approaches should be proposed for unbalanced datasets. Finally, our PointCluster achieves the remarkable clustering performance by progressively optimizing the feature model, the clustering head, and the entire network end-to-end. The training process with three stages is computationally complicated. In the future, how to design simple and effective one-stage deep clustering methods for 3-D point clouds would be a promising direction.

VI. CONCLUSION

In this article, we have presented PointCluster, a novel and general framework for deep clustering of 3-D point clouds, which can group point clouds into semantically meaningful clusters without relying on any human annotations. The clustering network is composed of two components: the feature model, which measures the instance-level similarity among point clouds, and the clustering head, which identifies the cluster-level difference. We adopt an unsupervised representation learning approach to train the feature model and a prototype pseudo-labeling algorithm to train the clustering head. Then, we jointly train them using a reliable pseudo-labeling algorithm to further boost the clustering performance. Extensive experiments on various datasets show that our PointCluster outperforms existing state-of-the-art clustering methods by a large margin and narrows down the performance gap between point cloud clustering and supervised point cloud classification. To the best of our knowledge, deep clustering of 3-D point clouds is scarcely studied. We believe that our clustering framework can offer a new perspective for the point cloud classification task and facilitate a wider range of point cloud learning tasks.

ACKNOWLEDGMENT

The authors would like to acknowledge Dr. Chuang Niu from Rensselaer Polytechnic Institute for providing technical guidance and expertise that greatly assisted our research.

REFERENCES

[1] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[2] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 820–830.

[3] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[4] H. Ran, W. Zhuo, J. Liu, and L. Lu, "Learning inner-group relations on point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15457–15467.

[5] T. Xiang, C. Zhang, Y. Song, J. Yu, and W. Cai, "Walk in the cloud: Learning curves for point clouds shape analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 895–904.

[6] M. Xu, R. Ding, H. Zhao, and X. Qi, "PAConv: Position adaptive convolution with dynamic kernel assembling on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3172–3181.

[7] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16239–16248.

[8] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," 2022, *arXiv:2202.07123*.

[9] Z. Wu et al., "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[10] A. X. Chang et al., "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[11] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1588–1597.

[12] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "PointContrast: Unsupervised pre-training for 3D point cloud understanding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 574–591.

[13] A. Xiao, J. Huang, D. Guan, X. Zhang, S. Lu, and L. Shao, "Unsupervised point cloud representation learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 11321–11339, Sep. 2023.

[14] S. Huang, Y. Xie, S.-C. Zhu, and Y. Zhu, "Spatio-temporal self-supervised representation learning for 3D point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6515–6525.

[15] Y. Rao, J. Lu, and J. Zhou, "PointGLR: Unsupervised structural representation learning of 3D point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2193–2207, Feb. 2023.

[16] J. Li, B. M. Chen, and G. H. Lee, "SO-Net: Self-organizing network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9397–9406.

[17] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-BERT: Pre-training 3D point cloud transformers with masked point modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 19291–19300.

[18] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via occlusion completion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9762–9772.

[19] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, Jul. 2018, pp. 40–49.

[20] L. Jing, L. Zhang, and Y. Tian, "Self-supervised feature learning by cross-modality and cross-view correspondences," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 1581–1891.

[21] L. Xue et al., "ULIP: Learning a unified representation of language, images, and point clouds for 3D understanding," 2022, *arXiv:2212.05171*.

[22] M. Afham, I. Dissanayake, D. Dissanayake, A. Dharmasiri, K. Thilakarathna, and R. Rodrigo, "CrossPoint: Self-supervised cross-modal contrastive learning for 3D point cloud understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9892–9902.

[23] J. MacQueen, "Some methods for classification and analysis of multivariate observation," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, 1967, pp. 281–297.

[24] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst., Natural Synth.*, 2001, pp. 849–856.

[25] P. Franti, O. Virmajoki, and V. Hautamaki, "Fast agglomerative clustering using a k-nearest neighbor graph," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1875–1881, Nov. 2006.

[26] L. Zhang and Z. Zhu, "Unsupervised feature learning for point cloud understanding by contrasting and clustering using graph convolutional neural networks," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 395–404.

[27] K. Hassani and M. Haley, "Unsupervised multi-task feature learning on point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8159–8170.

[28] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 132–149.

[29] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool, "SCAN: Learning to classify images without labels," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 268–285.

[30] C. Niu, H. Shan, and G. Wang, "SPICE: Semantic pseudo-labeling for image clustering," *IEEE Trans. Image Process.*, vol. 31, pp. 7264–7278, 2022.

[31] R. Roveri, L. Rahmann, A. C. Öztireli, and M. Gross, "A network architecture for point cloud classification via automatic depth images generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4176–4184.

[32] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.

[33] C. Wang, M. Cheng, F. Sohel, M. Bennamoun, and J. Li, "NormalNet: A voxel-based CNN for 3D object classification and retrieval," *Neurocomputing*, vol. 323, pp. 139–147, Jan. 2019.

[34] R. Zhang et al., "Parameter is not all you need: Starting from non-parametric networks for 3D point cloud analysis," 2023, *arXiv:2303.08134*.

[35] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. NIPS*, Dec. 2017, pp. 5099–5108.

[36] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8887–8896.

[37] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 484–499.

[38] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 206–215.

[39] J. Wu et al., "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 82–90.

[40] C.-L. Li, M. Zaheer, Y. Zhang, B. Poczos, and R. Salakhutdinov, "Point cloud GAN," 2018, *arXiv:1810.05795*.

[41] Z. Qi et al., "Contrast with reconstruct: Contrastive 3D representation learning guided by generative pretraining," 2023, *arXiv:2302.02318*.

[42] F. Liu, G. Lin, C.-S. Foo, C. K. Joshi, and J. Lin, "Point discriminative learning for data-efficient 3D point cloud analysis," 2021, *arXiv:2108.02104*.

[43] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[45] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2020, pp. 596–608.

[46] T. Li and C. Ding, "The relationships among various nonnegative matrix factorization methods for clustering," in *Proc. 6th Int. Conf. Data Mining (ICDM)*, Dec. 2006, pp. 362–371.

[47] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Machince Learn. Res.*, vol. 3, pp. 583–617, Jun. 2002.

[48] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985.

[49] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.

[50] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2022, pp. 604–621.

[51] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[52] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling," in *Proc. NIPS*, vol. 35, Dec. 2022, pp. 33330–33342.

[53] Y. Wu et al., "Self-supervised intra-modal and cross-modal contrastive learning for point cloud understanding," *IEEE Trans. Multimedia*, pp. 1–13, 2023.

[54] R. Zhang et al., "Point-M2AE: Multi-scale masked autoencoders for hierarchical point cloud pre-training," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 27061–27074.

[55] R. Zhang, L. Wang, Y. Qiao, P. Gao, and H. Li, "Learning 3D representations from 2D pre-trained models via image-to-point masked autoencoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 21769–21780.
[56] Y. Li, M. Yang, D. Peng, T. Li, J. Huang, and X. Peng, "Twin contrastive learning for online clustering," *Int. J. Comput. Vis.*, vol. 130, no. 9, pp. 2205–2221, Sep. 2022.
[57] Z. Huang, J. Chen, J. Zhang, and H. Shan, "Learning representation for clustering via prototype scattering and positive sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7509–7524, Jun. 2023.
[58] S. Zhou et al., "A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions," 2022, *arXiv:2206.07579*.
[59] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A survey of clustering with deep learning: From the perspective of network architecture," *IEEE Access*, vol. 6, pp. 39501–39514, 2018.

Xiu Liu received the bachelor's degree in computer science and technology from Xi'an University of Posts and Telecommunications, Xi'an, China, in 2021. She is currently pursuing the master's degree in software engineering with the School of Information Science and Technology, Northwest University, Xi'an.

Her research interests include deep learning and point cloud processing.

Xinxin Han received the B.S. degree from Xi'an University of Architecture and Technology, Xi'an, China, in 2022. She is currently pursuing the M.S. degree with the School of Information Science and Technology, Northwest University, Xi'an.

Her main research interests include point cloud processing and 3-D recognition.

Huan Xia received the B.S. degree from the North China University of Technology, Beijing, China, in 2022. He is currently pursuing the M.S. degree with the School of Information Science and Technology, Northwest University, Xi'an, China.

His main research interests include point cloud processing and deep learning

Kang Li received the B.S. and M.S. degrees in computer science and technology and the Ph.D. degree in computer software from Norwest University, Xi'an, China, in 2006, 2009, and 2013, respectively.

He was a Lecturer, from 2003 to 2010, an Assistant Professor, from 2010 to 2017, and has been a Professor, since 2023, with the School of Computer Science and Technology, Northwest University. His research interests include the development of digitization and modeling cultural heritage, as well as visualization analysis of 3-D data and images.

Haochen Zhao received the master's degree in cultural relics and museums from Northwest University, Xi'an, China in 2020.

He joined Shaanxi Institute for the Preservation of Cultural Heritage, Xi'an, in 2020. His research interests include cultural heritage conservation.

Jia Jia received the master's degree in archeology from the School of Cultural Heritage, Northwest University, Xi'an, China, in 2014. She is currently pursuing the Ph.D. degree with the Department of Cultural Heritage and Museology, Fudan University, Shanghai, China.

She has been working at Shaanxi Institute for the Preservation of Cultural Heritage, Xi'an, since 2016, as a Museologist. Her research interests include the conservation of mural painting, clay sculpture, and masonry.

Gang Zhen received the bachelor's degree in museum from Northwest University, Xi'an, China.

He works at Shaanxi Institute for the Preservation of Cultural Heritage, Xi'an, as a Professor of relics and museology. His research interests include cultural relics conservation and restoration.

Linzhi Su received the B.S. degree in electronic engineering and the Ph.D. degree in circuits and systems from Xidian University, Xi'an, China, in 2011 and 2016, respectively.

Since 2016, he has been a Lecturer with the School of Information Science and Technology, Northwest University, Xi'an. His research interests include deep learning and computed tomography.

Fengjun Zhao received the B.S. degree in electronics engineering and the Ph.D. degree in signal and information processing from Xidian University, Xi'an, China, in 2010 and 2015, respectively.

He was a Lecturer from 2015 and 2019 and has been an Associate Professor since 2019 with the School of Information Science and Technology, Northwest University, Xi'an. His research interests include image analysis and computer-aided diagnosis.

Xin Cao received the B.S. degree in electronic engineering and the Ph.D. degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2011 and 2016, respectively.

He was a Lecturer from 2016 and 2020 and has been an Associate Professor since 2020 with the School of Information Science and Technology, Northwest University, Xi'an. His research interests include point cloud processing and optical molecular imaging.