Curriculumformer: Taming Curriculum Pre-Training for Enhanced 3-D Point Cloud Understanding

Ben Fei, *Graduate Student Member, IEEE*, Tianyue Luo, Weidong Yang, *Member, IEEE*, Liwen Liu, Rui Zhang, *Graduate Student Member, IEEE*, and Ying He, *Member, IEEE*

***Abstract*— Learning universal representations of 3-D point clouds is essential for reducing the need for manual annotation of large-scale and irregular point cloud datasets. The current modus operandi for representative learning is self-supervised learning, which has shown great potential for improving point cloud understanding. Nevertheless, it remains an open problem how to employ auto-encoding for learning universal 3-D representations of irregularly structured point clouds, as previous methods focus on either global shapes or local geometries. To this end, we present a cascaded self-supervised point cloud representation learning framework, dubbed Curriculumformer, aiming to tame curriculum pre-training for enhanced point cloud understanding. Our main idea lies in devising a progressive pre-training strategy, which trains the Transformer in an easy-to-hard manner. Specifically, we first pre-train the Transformer using an upsampling strategy, which allows it to learn global information. Then, we follow up with a completion strategy, which enables the Transformer to gain insight into local geometries. Finally, we propose a Multi-Modal Multi-Modality Contrastive Learning (M4CL) strategy to enhance the ability of representation learning by enriching the Transformer with semantic information. In this way, the pre-trained Transformer can be easily transferred to a wide range of downstream applications. We demonstrate the superior performance of Curriculumformer on various discriminant and generative tasks, outperforming state-of-the-art methods. Moreover, Curriculumformer can also be integrated into other off-the-shelf methods to promote their performance. Our code is available at https://github.com/Fayeben/Curriculumformer.**

***Index Terms*— 3-D representation learning, curriculum learning, point clouds, self-supervised learning, transformer.**

I. Introduction

POINT clouds play an indispensable role in 3-D computer vision [1], [2] and are attracting growing attention due to their flexibility in representing arbitrary geometries and memory efficiency. Similar to 2-D images, there has been significant research in recent years focused on learning representations for 3-D point clouds [3]. The research field of 3-D representation learning shares many similarities with its 2-D counterpart, including the use of auto-encoder architecture [4] and self-reconstruction-based supervision [5]. Recent progresses in both natural language processing (NLP) [6] and 2-D computer vision [7] have promoted numerous developments in 3-D representation learning, including point cloud Transformer (PCT) [8], Point-BERT [5], and Point-M2AE [9]. However, a pressing question remains on how to pre-train networks on large-scale raw data, endowing them with universal representation abilities and benefiting downstream tasks with fine-tuning.

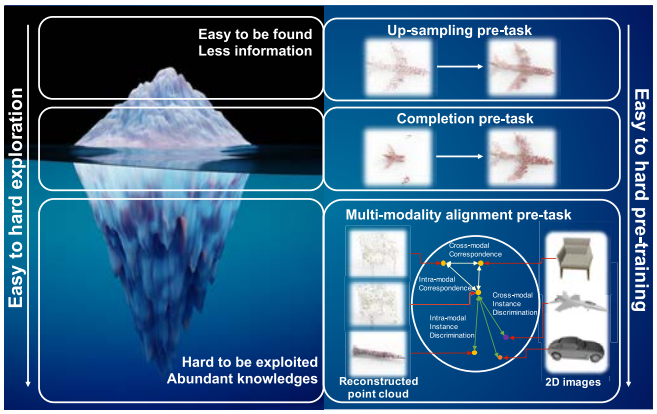

Fig. 1. Illustration of curriculum pre-training. Pre-training the Transformer in an easy-to-hard manner is akin to gradually excavating the tip of an iceberg. When pre-training the Transformer in a facile way, the model acquires less information and weaker representative capabilities. By progressively increasing the difficulty of self-supervised training, the network can obtain abundant knowledge, leading to superior representation learning and benefiting downstream tasks.

The question mentioned above poses two main challenges to enhancing the transfer capability of pre-trained models: fully exploiting global structures and local geometries [4], and boosting pre-training accuracy by integrating semantic information [10]. To address the former challenge, previous methods such as Point-BERT [5] and 3-D auto-encoder [4] have employed self-reconstruction as supervision to focus on either global structures or local parts of the input point clouds. However, a single self-reconstruction architecture often causes models to focus solely on either local parts or global structures, rather than effectively balancing both aspects [11]. To tackle the latter challenge, several methods

Manuscript received 15 June 2023; revised 15 December 2023 and 28 February 2024; accepted 22 May 2024. Date of publication 13 June 2024; date of current version 7 April 2025. This work was supported in part by the National Natural Science Foundation of China under Grant U2033209, and in part by the Ministry of Education, Singapore, under its Academic Research Fund under Grant MOE-T2EP20220-0005 and Grant RT19/22. *(Ben Fei and Tianyue Luo are co-first authors.) (Corresponding authors: Weidong Yang; Ying He.)*

Ben Fei, Tianyue Luo, Weidong Yang, Liwen Liu, and Rui Zhang are with the School of Computer Science, Fudan University, Shanghai 200433, China (e-mail: bfei21@m.fudan.edu.cn; tianyueluo21@m.fudan.edu.cn; wdyang@fudan.edu.cn; liwenliu21@m.fudan.edu.cn; 22210240379@m.fudan.edu.cn).

Ying He is with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: yhe@ntu.edu.sg).

This article has supplementary downloadable material available at https://doi.org/10.1109//TNNLS.2024.3406587, provided by the authors.

Digital Object Identifier 10.1109/TNNLS.2024.3406587

utilize contrastive learning between point clouds and images as a self-supervised pre-training strategy [12]. However, even after augmentation, point clouds and images are still easily aligned, weakening the representative learning ability of the pre-trained model.

This article aims to learn a powerful representation of point clouds that does not require manually annotated supervision. To address the aforementioned challenges, we focus on the pre-training stage and propose a novel curriculum pre-training framework named Curriculumformer. Our framework consists of two essential components: a progressive pre-training tactic and a multi-modal multi-modality contrastive learning (M4CL) module (Fig. 1). Specifically, the progressively harder pre-training setting can be divided into the upsampling pre-task, the completion pre-task, and the M4CL module. The upsampling strategy guides the network in upsampling a sparse point cloud into a denser one, thereby enabling the models to learn global shapes [13]. This initial pre-training stage serves as a warm-up, allowing the model to develop a basic understanding of the underlying structure of the input data. Then the completion pre-task follows, which focuses on completing partial or missing information in the input data, allowing the model to gain insight into the local geometries [1]. The advanced course in Curriculumformer is the M4CL task, which concentrates on learning semantic information at the object level [14], [15]. Through the cascaded pre-training strategy of Curriculumformer, the obtained model can learn hierarchical information from both global structures and local geometries.

To enhance the accuracy of the self-reconstruction-based strategy, we must reconsider the previously mentioned M4CL strategy, which was developed with a "Kill Two Birds with One Stone" purpose. In detail, the M4CL strategy involves embedding both the reconstructed point cloud and the rendered 2-D image into a feature space that is close to each other. In real-world applications such as robotics and autonomous driving, it has been shown that models that understand the 3-D–2-D correspondence can greatly facilitate a full understanding of 3-D world [16]. The M4CL module follows a joint objective of embedding augmented versions of the same point cloud into a close feature space while maintaining 3-D–2-D correspondence between them and the rendered 2-D image of the original 3-D point cloud, which can be treated as the third stage and the most difficult course of pre-training. Unlike existing methods that employ data augmentation of 3-D point cloud [12], M4CL uses a reconstructed point cloud as the output of a Transformer-based 3-D branch in 3-D–2-D correspondence. This not only enables the model to align the reconstructed point cloud with greater accuracy to learn more meaningful representatives but also enhances the difficulty of the last pre-training phase. By leveraging the merits of our curriculum pre-training framework and the M4CL module, we obtain a powerful pre-trained model whose trained encoder can be transferred to various downstream tasks. Our experiments demonstrate the superior performance of Curriculumformer, which outperforms widely used methods on several benchmarks.

The contributions of this article are summarized as follows.

1) We introduce a novel curriculum pre-training framework dubbed *Curriculumformer*. Unlike previous single-task pre-training strategies, *Curriculumformer* designs an asymmetrical encoder–decoder Transformer architecture that is pre-trained in an easy-to-hard manner. This allows the model to learn local and global patterns step-by-step, leading to enhanced point cloud understanding.
2) The M4CL module, devised as the hardest stage in the pre-training, aims to align the reconstructed point cloud and the corresponding image while fully exploiting the representative capability of the Transformer backbone.
3) With the aid of the curriculum pre-training tactic and the M4CL module, the well-trained encoder can be easily transferred to a variety of downstream applications. We demonstrate the superiority of *Curriculumformer* in five different downstream tasks, achieving the most competitive performance compared to other pre-training or train-from-scratch methods. Our curriculum pre-training tactic can also promote the performance of the existing self-supervised methods.

II. Related Work

A. Curriculum Learning

Curriculum learning [17] is a learning paradigm inspired by human learning that involves starting from simple tasks and gradually increasing to more difficult ones. Many studies show that this training approach leads to better generalization and faster convergence in various tasks in 2-D vision and NLP [18], [19]. Several methods feed training samples to the network from easy to hard to improve the neural network's performance for image classification [18], question-answer [20], and machine translation [21]. Recent studies focus on exploiting the curriculum learning tactic at the task level, demonstrating that a set of well-designed curriculums can benefit learning complex knowledge. For example, curriculum learning for natural answer generation (CL-NAG) [19] progressively increases the degrees of parallelism from easy to hard, resulting in remarkable accuracy improvements over previous non-autoregressive methods in neural machine translation. Another example is multi model curriculum learning (MMCL) [22], which utilizes curriculum learning to transfer well-learned knowledge to a target task.

Curriculum learning is now emerging as a promising approach in the field of 3-D vision. For instance, a differentiable template matching strategy with curriculum learning has enabled an automatic 3-D annotation model [23]. Additionally, curriculum learning has been integrated into 3-D self-supervised learning, as seen in the case of PointSmile, which incorporates curriculum data augmentation (CDA) of point clouds. PointSmile utilizes CDA to learn from easy-to-hard samples, dynamically affecting the latent space to create better embeddings. However, PointSmile only applies curriculum learning at the data level, whereas our Curriculumformer takes a step further by incorporating curriculum learning at the task level.

B. Representation Learning on Point Clouds

Due to irregular structure and permutation invariance during point cloud manipulation, learning point cloud representation is more challenging than other modalities [24], [25]. Pioneered

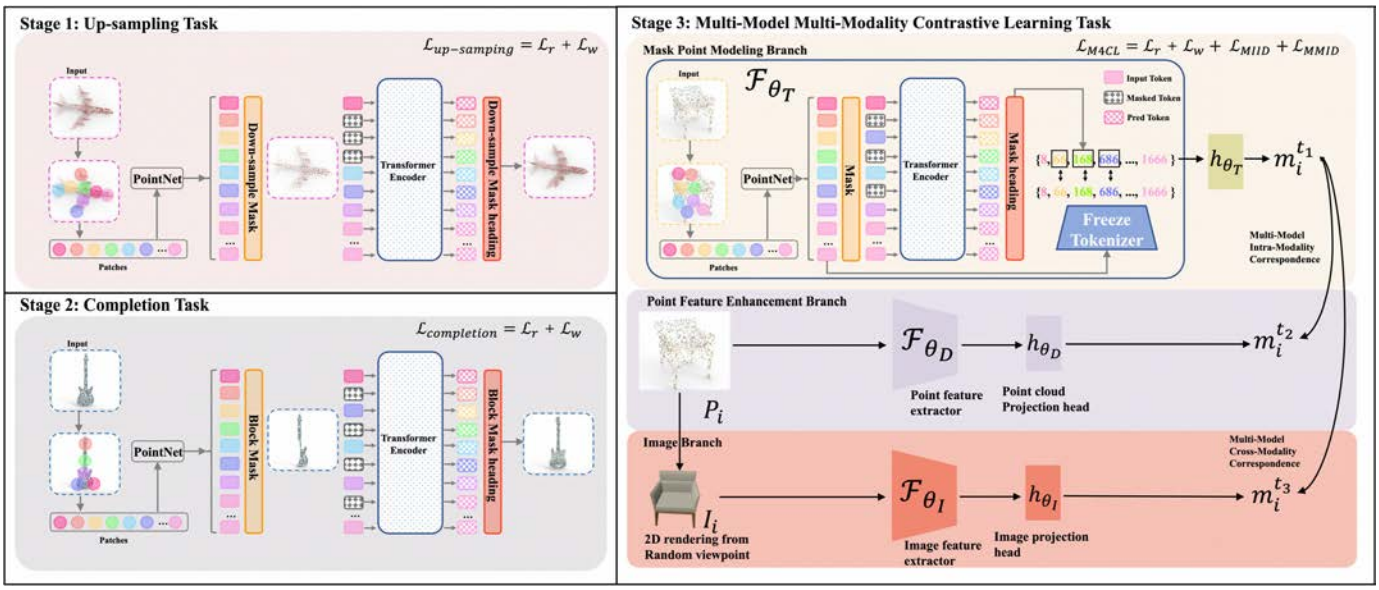

Fig. 2. Overview of the pre-training strategy of the Curriculumformer. The elementary courses consist of three pre-training objectives: 1) DMPM; 2) BMPM; and 3) M4CL strategy.

TABLE I

SHAPE CLASSIFICATION ON MODELNET40. [ST] AND [T] ARE DENOTED AS THE STANDARD TRANSFORMERS AND TRANSFORMER-BASED METHODS, RESPECTIVELY

	Methods	Accuracy
Supervised	PointNet [26]	89.2
	PointNet++ [27]	90.7
	PointWeb [43]	92.3
	SpiderCNN [44]	92.4
	PointCNN [28]	92.5
	KPConv [45]	92.9
	DGCNN [41]	92.9
	RS-CNN [46]	92.9
	DensePoint [47]	93.2
	[T]PCT [8]	93.2
	[T]PVT [48]	93.6
	[T]PointTransformer [49]	93.7
	[ST]Transformer [5]	91.4
Self-supervised	OcCo [35]	93.0
	STRL [50]	93.1
	[ST]Transformer-OcCo [5]	92.1
	PointNet + CrossPoint [12]	92.8
	DGCNN + CrossPoint [12]	93.0
	[ST]Point-BERT [5]	93.2
	[ST]**Curriculumformer**	**93.6**
	[T]MaskPoint [51]	93.8
	[T]**Curriculumformer + MaskPoint**	**94.0**
	[T]Point-MAE [11]	93.8
	[T]**Curriculumformer + Point-MAE**	**94.0**
	[T]Point-M2AE [9]	94.0
	[T]**Curriculumformer + Point-M2AE**	**94.1**

TABLE II

COMPARISON OF SHAPE CLASSIFICATION PERFORMANCE ON SCANOBJECTNN. THE ACCURACY (%) ON THREE SPLITS SETTINGS OF SCANOBJECTNN IS LISTED

Methods	OBJ-BG	OBJ-ONLY	PB-T50-RS
PointNet [26]	73.3	79.2	68.0
PointNet++ [27]	82.3	84.3	77.9
DGCNN [41]	82.8	86.2	78.1
PointCNN [28]	86.1	85.5	78.5
SpiderCNN [44]	77.1	79.5	73.7
BGA-DGCNN [53]	-	-	79.7
BGA-PN++ [53]	-	-	80.2
Transformer [5]	79.9	80.6	77.2
Transformer+OcCo [5]	84.9	85.5	78.8
[ST]Point-BERT [5]	87.4	88.1	83.0
[ST]**Curriculumformer**	**89.5**	**88.5**	**84.5**
[T]MaskPoint [51]	89.7	89.3	84.6
[T]**Curriculumformer + MaskPoint**	**90.0**	**89.5**	**84.9**
[T]Point-MAE [11]	90.02	88.29	85.18
[T]**Curriculumformer + Point-MAE**	**90.71**	**88.98**	**85.32**
[T]Point-M2AE [9]	91.22	88.81	86.43
[T]**Curriculumformer + Point-M2AE**	**91.74**	**89.16**	**86.54**

by PointNet [26] and PointNet++ [27], recent contributions directly process point cloud without any pre-preparation and have facilitated boosting many advances in point cloud-based tasks, including 3-D object classification [27], 3-D object detection [28], and 3-D point cloud synthesis [3]. Whereas, the performance of such representation learning works still largely relies on the annotated point cloud. Furthermore, cTree [29] has been introduced to learn point cloud representations in a label-efficient setting such as few-shot learning. By comparison, our Curriculumformer pays attention to learning transferable point cloud representations with an absence of annotations, benefiting various downstream tasks.

C. Self-Supervised Learning on 3-D Vision

In recent years, 3-D representation learning has been widely studied, yet without annotations [30], [31]. Most of the studies mainly devise various pre-tasks to recover the

TABLE III
COMPARISON OF FEW-SHOT CLASSIFICATION PERFORMANCE ON MODELNET40. FOR A FAIR COMPARISON, THE AVERAGE ACCURACY (%) AND STANDARD DEVIATION (%) OF TEN EXPERIMENTS ARE REPORTED

Methods	5-way		10-way	
	10-shot	20-shot	10-shot	20-shot
DGCNN [41]	91.8 ± 3.7	93.4 ± 3.2	86.3 ± 6.2	90.9 ± 5.1
DGCNN + OcCo [35]	90.9 ± 4.8	93.5 ± 4.4	84.6 ± 4.7	90.2 ± 2.2
CrossPoint + PointNet [12]	92.5 ± 3.0	94.9 ± 2.1	83.6 ± 5.3	87.9 ± 4.2
CrossPoint + DGCNN [12]	91.9 ± 3.3	93.9 ± 3.1	86.4 ± 5.4	91.3 ± 4.6
Transformer [5]	87.8 ± 5.2	93.3 ± 4.3	84.6 ± 5.5	89.4 ± 6.3
[ST]Transformer +OcCo [5]	94.0 ± 3.6	95.9 ± 2.3	89.4 ± 5.1	92.4 ± 4.6
[ST]Point-BERT [5]	94.6 ± 3.1	96.3 ± 2.7	**92.3 ± 4.5**	92.7 ± 5.1
[ST]**Curriculumformer**	**97.0 ± 1.6**	**97.8 ± 1.7**	92.1 ± 5.0	**94.8 ± 3.8**
[T]MaskPoint [51]	95.0 ± 3.7	97.2 ± 1.7	91.4 ± 4.0	93.4 ± 3.5
[T]**Curriculumformer + MaskPoint**	**97.4 ± 2.0**	**98.4 ± 0.8**	**93.2 ± 4.0**	**94.9 ± 3.4**
[T]Point-MAE [11]	96.3 ± 2.5	97.8 ± 1.8	92.6 ± 4.1	95.0 ± 3.0
[T]**Curriculumformer + Point-MAE**	**97.0 ± 2.4**	**98.0 ± 1.3**	**93.1 ± 4.1**	**95.1 ± 3.4**
[T]Point-M2AE [9]	96.8 ± 1.8	**98.3 ± 1.4**	92.3 ± 4.5	95.0 ± 3.0
[T]**Curriculumformer + Point-M2AE**	**97.0 ± 1.6**	97.6 ± 1.58	**92.7 ± 4.1**	**95.4 ± 2.8**

augmented input point cloud on the basis of latent vectors, including rotation [32], deformation [33], rearranged parts [34], and occlusion [35]. On the other hand, contrastive learning can be regarded as a powerful tactic to learn representations. For instance, PointContrast [36] employs contrastive learning on features of the same point cloud from different views to learn discriminative 3-D representations. Further, DepthContrast [37] devises the contrast for depth maps undergoing various augmentations. CrossPoint [12] performs cross-modality contrastive learning between rendered images and point clouds to obtain abundant self-supervised signals. For the first time, Point-BERT [5] introduces BERT into 3-D point clouds pre-training and achieves remarkable results on several downstream tasks. GrowSP [38] proposes a straightforward strategy that enables the progressive growth of superpoints throughout the training process, facilitating the gradual acquisition of meaningful semantic elements. Moreover, PointDC [39] is a self-supervised framework designed for 3-D semantic segmentation. It consists of two main components: cross-modal distillation and super-voxel clustering. However, a single pre-task might not unleash the full potential of pre-trained models, and our Curriculumformer is proposed to tackle this issue in a curriculum learning framework.

III. CURRICULUMFORMER

In this article, we revisit self-supervised point cloud representation learning by presenting a curriculum pre-training objective. We begin by introducing the preliminaries of the proposed method (see Section III-A). Following that, we present curriculum pre-training formulated in both upsampling and completion pre-tasks (see Section III-B) and M4CL (see Section III-C) settings. At last, we present our overall training objective (see Section III-D). Overview of the devised framework is depicted in Fig. 2.

A. Preliminaries

In the former two stages of pre-training, the given dataset $\mathcal{S} = \{\mathbf{P}_i\}_{i=1}^{|\mathcal{S}|}$ will undergo downsampling masked points modeling (DMPM) and block masked points modeling (BMPM) to obtain spare point clouds $\mathbf{U}_i$ and partial ones $\mathbf{B}_i$, respectively. The objective of these two pre-tasks is to upsample and complete the masked point clouds in a cascaded way. In the last stage of pre-training, given a dataset $\mathcal{S} = \{(\mathbf{P}_i, \mathbf{I}_i)\}_{i=1}^{|\mathcal{S}|}$ with $\mathbf{P}_i \in \mathbb{R}^{N\times 3}$ and $\mathbf{I}_i \in \mathbb{R}^{H\times W\times 3}$, where $\mathbf{I}_i$ denotes a rendered 2-D image modality of point cloud $\mathbf{P}_i$ [40]. Curriculumformer trains a Transformer-based point cloud rebuilder $\mathcal{F}_{\theta_T}(\cdot)$, a point cloud feature learner $\mathcal{F}_{\theta_D}(\cdot)$, and an image feature learner $\mathcal{F}_{\theta_I}(\cdot)$ in a self-supervised manner, where multi-layer perceptron (MLP) projection heads $h_{\phi_T}(.)$, $h_{\phi_D}(.)$, and $h_{\phi_I}(.)$ are devised for reconstructed point cloud, point cloud, and image, respectively. After these three stages of pre-training, the Transformer backbone could be effectively and efficiently transferred to downstream tasks.

B. Curriculum Pre-Training Framework

Given a point cloud, the tokenization of each point in a naive approach is computationally expensive, as it treats each point as a separate token. This is because the point-wise reconstruction task involves quadratic complexity of self-attention in Transformers. Following the patch embedding approach used in Vision Transformers [7], we group each point cloud into multiple local patches or sub-clouds, which is a simple and efficient strategy. Specifically, to obtain point patch embeddings, we follow Point-BERT [5] to train a discrete variational auto-encoder (dVAE) to get point embeddings as input of the Transformer. Specifically, when given input point cloud, it is first divided into l local patches with c_i as center points, which will be projected into point embeddings e_i through a PointNet [26]. After that, positional embeddings PE_i of each patch are obtained via applying an MLP on the corresponding center point c_i. The input embeddings are defined as x_i, regarded as a union of point embeddings e_i, and positional embeddings PE_i, which will be fed into the Transformer backbone. Encouraged by Devlin et al. [6], class token [**cls**] is appended to input spaces. In this way, the input sequence of the Transformer can be written as $H^0 = \mathrm{Concat}\{\mathbf{cls}, x_i\}$. The output of final layer $H^F = \{h^F_{\mathrm{cls}}, h^F_1, \ldots, h^F_l\}$ represents the combination of global feature and encoded representation of input point patches.

In detail, the masked positions are denoted as $\mathbf{M} \in \{1, \ldots, k\}^{\lfloor rk \rfloor}$ with r as the ratio of the mask, which is set to 0.25–0.45 in all pre-training stages. Following that, all masked point embeddings are replaced with learnable pre-defined mask embeddings $E[\mathbf{M}]$ and maintain their positional embeddings. At last, the input embeddings $\mathrm{O}^{\mathbf{M}} = \{x_i | i \notin \mathbf{M}\}_{i=1}^{l} \cup \{E[\mathbf{M}] + \mathrm{PE}_i | i \in \mathbf{M}\}_{i=1}^{l}$ can be regarded as the input of Transformer encoder.

TABLE IV

COMPARISON OF PART SEGMENTATION PERFORMANCE ON THE SHAPENETPART. THE MIOU ACROSS ALL INSTANCE MIOU (%) AND THE IOU (%) FOR EACH CATEGORIES ARE COMPARED

Methods	mIoU_C	mIoU_I	Aero	Bag	Cap	Car	Chair	Ear	Guitar	Knife	Lamp	Lap	Motor	Mug	Pistol	Rock	Skate	table
PointNet [26]	80.4	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PointNet++ [27]	81.9	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
DGCNN [41]	82.3	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
Transformer [5]	83.4	85.1	82.9	85.4	87.7	78.8	90.5	90.8	91.1	87.7	85.3	95.6	73.9	94.9	83.5	61.2	74.9	80.6
Transformer+OcCo [5]	83.4	85.1	83.3	85.2	88.3	79.9	90.7	74.1	91.9	87.6	84.7	95.4	75.5	94.4	84.1	63.1	75.7	80.8
Point-BERT [5]	84.1	85.6	**84.3**	**84.8**	**88.0**	**79.8**	91.0	**81.7**	91.6	**87.9**	85.2	**95.6**	75.6	**94.7**	**84.3**	**63.4**	**76.3**	81.5
Curriculumformer + Point-BERT	**84.5**	**85.9**	**84.3**	82.2	87.6	79.7	**91.1**	73.7	**91.8**	87.8	**85.3**	95.2	**75.8**	94.4	84.2	62.6	**76.3**	**82.5**
MaskPoint [51]	83.3	**85.4**	**83.9**	**84.0**	87.5	79.4	**90.8**	78.3	**91.9**	**87.5**	85.1	**95.5**	74.2	94.2	**84.6**	59.9	**75.3**	81.4
Curriculumformer + MaskPoint	**83.5**	**85.4**	83.7	81.9	**87.8**	**79.5**	90.6	**79.2**	91.5	87.5	**85.6**	95.3	**75.2**	**94.7**	83.7	**63.6**	75.2	**81.5**
Point-MAE [11]	84.1	**86.1**	84.3	**85.0**	88.3	80.5	91.3	**78.5**	**92.1**	87.4	**86.1**	96.1	75.2	94.6	84.7	**63.5**	**77.1**	**82.4**
Curriculumformer + Point-MAE	**84.4**	**86.1**	**85.2**	84.8	**88.9**	**81.0**	**91.6**	75.4	91.9	**87.8**	85.9	**96.2**	**76.8**	**95.0**	**85.0**	61.7	77.0	81.6
Point-M2AE [9]	84.7	**86.5**	85.2	85.1	**89.4**	**81.5**	**91.9**	**76.9**	92.1	88.2	86.2	96.1	**77.8**	95.0	84.6	**66.5**	76.1	**82.7**
Curriculumformer + Point-M2AE	**85.0**	86.4	**85.6**	**86.7**	88.9	81.2	91.7	73.6	**92.2**	**88.3**	**86.3**	**96.3**	77.4	**95.3**	**85.8**	65.6	**77.1**	82.0

1) Upsampling Pre-Task: On the one hand, the goal of upsampling task is to upsample a dense object on the basis of a sparse one. The pre-trained dVAE encodes each local patch into discrete point tokens, standing for geometric information. After that, DMPM is leveraged to mask point patches into a sparse point cloud, which is shown in Stage 1 in Fig. 2. Therefore, those informative tokens can be directly applied as surrogate supervision signals to pre-train the backbone. With the objective of un-sampling, the Transformer will be enriched with global information due to this pre-text task focuses more on the whole shape. Noted that curriculum learning in this article aims to pre-train the Transformer from easy to hard. As shown in the Supplementary Material, the DMPM strategy is the easiest to be trained, where pre-training loss converged fastly while the accuracy of pre-training is the highest. Therefore, the DMPM strategy is carried out first to endow the Transformer with a certain ability of representative learning.

2) Completion Pre-Task: On the contrary, the target of point cloud completion pre-text task is to endow the Transformer to infer local geometric details of missing parts given the remaining ones. Unlike the DMPM pre-text task, point patches will go through BMPM module, from which partial point clouds could be obtained. The reason for the BMPM strategy in point cloud completion pre-text task can be found in the Supplementary Material, from which we can conclude that the BMPM tactic is much more difficult than the DMPM tactic. If we train the Transformer with BMPM and DMPM in a cascaded way, the Transformer will be initialized from a better state, with an even faster converge speed and lower loss of pre-training.

C. Multi-Modal Multi-Modality Contrastive Learning

Furthermore, after being trained with upsampling and completion pre-text tasks, the model is strengthening at exploiting both local and global information. However, the supervision of reconstruction might not be enough to recover a high-fidelity point cloud. Therefore, by the merits of the modality of images and feature alignment from the same point cloud, the accuracy of reconstruction can be enhanced greatly, which could also be regarded as the last course of curriculum pre-training due to the hardness of this task (see Fig. 3).

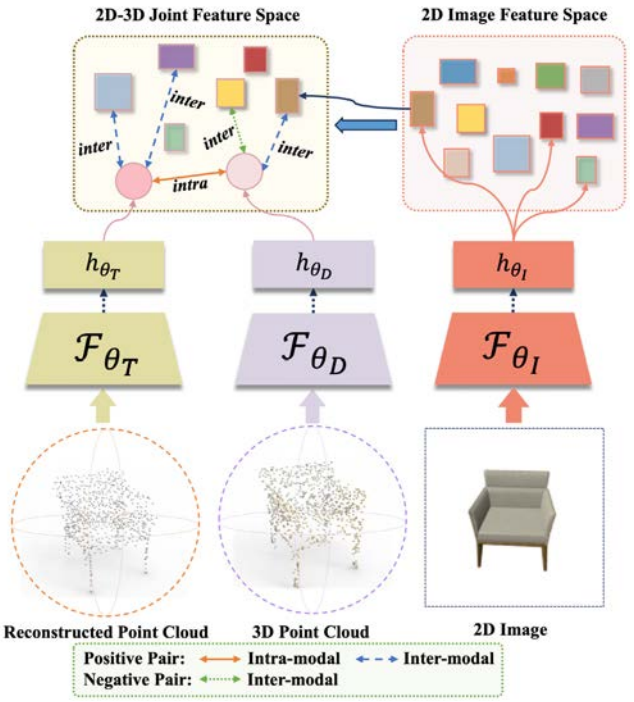

Fig. 3. Overview of M4CL strategy of our Curriculumformer.

1) Multi-Modal Intra-Modality Instance Discrimination: To promote the reconstruction accuracy and exploit the cross-model learning, we propose multi-model intra-modality instance discrimination (MIID). Different from intra-modal instance discrimination in CrossPoint [12], which enforces invariance to a collection of point cloud geometric transformations to feed augmented point cloud into dynamic graph convolutional neural network (DGCNN) [41], we utilize the Transformer and DGCNN as our two 3-D backbones. In our setting, given an input point cloud $\mathbf{P}_i$, point cloud does not need to be transformed, while the recovered point cloud $\mathbf{P}_i^T$ from the Transformer can be naturally regarded as a transformed point cloud.

Furthermore, DGCNN-based [41] point cloud feature extractor $\mathcal{F}_{\theta_D}$ maps $\mathbf{P}_i$ to a feature embedding space, while Transformer $\mathcal{F}_{\theta_T}$ recover the incomplete point cloud and then

map the reconstructed $\mathbf{P}_i^T$ into the feature embedding space. After that, the resulting feature vectors will be projected to an invariant space $\mathbb{R}^d$, where the contrastive loss can be applied, leveraging the projection head h_{ϕ_D} and h_{ϕ_T}, respectively. The projected vectors of $\mathbf{P}_i^D$ and $\mathbf{P}_i^T$ are denoted as $m_i^{t_1}$ and $m_i^{t_2}$, respectively. The target is to maximize the similarity between $m_i^{t_1}$ and $m_i^{t_2}$ while minimizing the similarity among other projected vectors in the mini-batch. The NT-Xent loss presented in SimCLR [42] is utilized as an instance discriminator in MIID, where the loss function for positive pair can be calculated as follows:

$$l(i,t_1,t_2) = -\log \frac{\exp\left(\frac{s\left(\mathbf{m}_i^{t_1},\mathbf{m}_i^{t_2}\right)}{\tau}\right)}{\sum_{\substack{k=1\\k\neq i}}^{N}\exp\left(\frac{s\left(\mathbf{m}_i^{t_1},\mathbf{m}_k^{t_1}\right)}{\tau}\right)+\sum_{k=1}^{N}\exp\left(\frac{s\left(\mathbf{m}_i^{t_1},\mathbf{m}_k^{t_2}\right)}{\tau}\right)} \tag{1}$$

where N stands for the size of mini-batch, τ denotes as the temperature co-efficient, and $s(\cdot)$ is the cosine similarity function. Therefore, the MIID loss function L_{MIID} can be formulated as

$$\mathcal{L}_{\text{MIID}} = \frac{1}{2N}\sum_{i=1}^{N}[l(i,t_1,t_2)+l(i,t_2,t_1)]. \tag{2}$$

2) Multi-Modal Multi-Modality Instance Discrimination: Apart from feature alignment between the reconstructed point cloud and original point cloud modality, an additional contrastive objective between point clouds and images is also proposed to learn discriminative information, resulting in higher accuracy of the reconstruction and enhancing representation learning capability of point clouds. Therefore, specifically, the rendered 2-D image $\mathbf{I}_i$ of $\mathbf{P}_i$ is first embedded to a feature space by utilizing ResNet34-based [52] backbone $\mathcal{F}_{\theta_I}$, followed by projection of feature vectors into invariant space $\mathbb{R}^d$ through image projection head h_{ϕ_I}. In this way, the projected image feature can be written as $m_i^{t_3}$. Further, multi-modal multi-modality instance discrimination (MMID) is performed between the modalities of point clouds and images for an enhanced point cloud understanding.

The aim is to maximize the similarity between $m_i^{t_1}$ and $m_i^{t_3}$ in invariance space since they come from the same objects. Cross-modality alignment enables the network to learn from harder positive and negative data, thus enhancing representation ability compared to solely learning from intra-modality alignment. Overall, the loss function for the positive pair is computed as follows:

$$c(i,t_1,t_3) = -\log \frac{\exp\left(\frac{s\left(\mathbf{m}_i^{t_1},\mathbf{m}_i^{t_3}\right)}{\tau}\right)}{\sum_{\substack{k=1\\k\neq i}}^{N}\exp\left(\frac{s\left(\mathbf{m}_i^{t_1},\mathbf{m}_k^{t_1}\right)}{\tau}\right)+\sum_{k=1}^{N}\exp\left(\frac{s\left(\mathbf{m}_i^{t_1},\mathbf{m}_k^{t_3}\right)}{\tau}\right)} \tag{3}$$

where s, N, τ represent for same parameters as in (1). The MMID loss function L_{MMID} is then described as

$$\mathcal{L}_{\text{MMID}} = \frac{1}{2N}\sum_{i=1}^{N}[c(i,t_1,t_3)+c(i,t_3,t_1)]. \tag{4}$$

D. Overall Objectives

During the pre-training phases with upsampling and completion pre-text tasks, reconstruction loss and contrastive loss are leveraged at the same time.

To increase the difficulty of our Curriculumformer's pre-training with reconstruction loss, we applied a well-defined auxiliary task of mixed token prediction, inspired by CutMix [67]. The normalization process has removed the absolute position information of each sub-cloud, allowing for easy generation of virtual samples through the combination of two sub-cloud groups without the need for complex patch alignment techniques [68]. Additionally, as part of the pre-training process, each virtual sample is expected to predict the tokens produced by the original corresponding sub-cloud, which facilitates the DMPM and BMPM tasks. We generate the same number of virtual samples as real samples to increase the difficulty of pre-training tasks. By doing so, the training potential of Transformers in low-data regimes is enhanced.

DMPM task aims to reconstruct the point tokens that correspond to the downsample-masked locations, while BMPM tries to recover partial-masked locations. Therefore, the pre-training objectives of both two pre-text tasks are to maximize the log-likelihood of the correct point tokens v_i when presented with the masked input embeddings $\mathrm{U}^{\mathcal{M}}$

$$\mathcal{L}_r = \max \sum_{\mathrm{U}\in D}\mathbb{E}_{\mathcal{M}}\left[\sum_{i\in\mathcal{M}}\log \mathcal{P}\left(v_i\middle|\mathrm{U}^{\mathcal{M}}\right)\right]. \tag{5}$$

The DMPM and BMPM tasks encourage the prediction of the masked geometric structure of point clouds. However, relying solely on these tasks for transformer training results in inadequate comprehension of the point clouds' higher-level semantics. Hence, to enhance the transformer networks' acquisition of high-level semantics, we utilized MoCo [69], a popular contrastive learning technique. Combined with the point patch mixing technique, the model is encouraged to focus on the high-level semantics of point clouds through the optimization of contrastive loss, ensuring that the features of virtual samples are closely aligned with those of the original samples. Consider a mixed sample feature w derived from two other samples' features, denoted as g_1^+ and $g_2^+(\{g_i\}$. The contrastive loss, which depends on the mixing ratio r, is expressed as

$$\mathcal{L}_w = -r\log\frac{\exp\left(wg_1^+/\tau\right)}{\sum_{i=0}^{K}\exp(wg_i/\tau)} - (1-r)\log\frac{\exp\left(wg_2^+/\tau\right)}{\sum_{i=0}^{G}\exp(wg_i/\tau)} \tag{6}$$

where τ refers to temperature and G is the memory bank size. By combining DMPM and BMPM objectives with contrastive loss, our Curriculumformer can capture both local geometric

TABLE V

COMPARISON OF CURRICULUMFORMER FINE-TUNED ON SHAPENET55, SHAPENET34, AND SHAPENETUNSEEN21 AND OTHER NETWORKS REGARDING CD-$\ell_1 \times 10^3$, CD-$\ell_2 \times 10^3$, AND THE AVERAGE F-SCORE@1%. THREE DIFFICULT DEGREES INCLUDING CD-*S*, CD-*M*, AND CD-*H* ARE LEVERAGED TO VALIDATE THE COMPLETION PERFORMANCE, STANDING FOR THE *Simple*, *Moderate*, AND *Hard* SETTINGS

Methods	ShapeNet55					ShapeNet34					ShapeNetUnseen21				
	CD-*S* (CD-ℓ_1/ CD-ℓ_2)	CD-*M* (CD-ℓ_1/ CD-ℓ_2)	CD-*H* (CD-ℓ_1/ CD-ℓ_2)	CD-*Avg.* (CD-ℓ_1/ CD-ℓ_2)	F-Socre -Avg	CD-*S* (CD-ℓ_1/ CD-ℓ_2)	CD-*M* (CD-ℓ_1/ CD-ℓ_2)	CD-*H* (CD-ℓ_1/ CD-ℓ_2)	CD-*Avg.* (CD-ℓ_1/ CD-ℓ_2)	F-Socre -Avg	CD-*S* (CD-ℓ_1/ CD-ℓ_2)	CD-*M* (CD-ℓ_1/ CD-ℓ_2)	CD-*H* (CD-ℓ_1/ CD-ℓ_2)	CD-*Avg.* (CD-ℓ_1/ CD-ℓ_2)	F-Socre -Avg
ASFMNet [54]	19.138 1.308	20.172 1.517	23.513 2.282	20.941 1.702	0.247	18.350 1.189	19.123 1.343	21.913 1.909	19.795 1.480	0.268	21.591 1.995	23.006 2.342	27.682 3.660	24.075 2.666	0.216
TopNet [55]	27.233 2.483	28.749 2.848	33.986 4.642	29.989 3.324	0.110	22.382 1.606	23.271 1.793	26.020 2.432	23.891 1.944	0.154	26.775 2.499	28.312 2.928	33.121 4.407	29.403 3.278	0.103
GRNet [56]	19.159 1.137	20.645 1.489	24.034 2.394	21.279 1.673	0.239	18.809 1.102	20.034 1.366	22.989 2.089	20.611 1.519	0.247	21.245 1.552	23.753 2.281	49.427 4.169	24.808 2.667	0.208
FoldingNet [57]	25.203 2.095	26.596 2.410	30.424 3.333	27.408 2.613	0.091	23.556 1.859	24.466 2.059	27.584 2.759	25.202 2.226	0.137	28.356 2.887	29.833 3.290	35.356 4.968	31.182 3.715	0.088
CRN [58]	21.207 1.502	22.364 1.801	25.849 2.726	23.140 2.010	0.205	20.304 1.362	21.216 1.594	24.159 2.318	21.893 1.758	0.221	24.247 2.237	26.076 2.840	31.771 4.833	27.365 3.303	0.177
PCN [59]	22.990 1.811	23.976 2.062	27.360 2.937	24.775 2.270	0.167	21.433 1.551	22.304 1.753	25.086 2.426	22.941 1.910	0.192	27.593 2.983	28.989 3.442	34.598 5.558	30.393 3.994	0.128
ECG [60]	16.710 1.167	18.727 1.545	23.480 2.555	19.639 1.756	0.321	13.122 0.735	14.628 0.996	18.461 1.696	15.404 1.142	**0.496**	15.282 1.255	17.595 1.759	23.535 3.267	18.804 2.094	**0.460**
PoinTr [61]	12.491 0.698	14.182 1.049	18.811 2.022	15.161 1.256	0.446	12.006 0.632	13.393 0.910	17.365 1.697	14.255 1.080	0.459	13.290 0.838	15.522 1.376	21.881 3.070	16.898 1.761	0.421
SnowflakeNet [62]	13.568 0.680	15.380 0.979	19.412 1.754	16.120 1.138	0.362	13.612 0.693	15.272 0.968	19.385 1.727	16.090 1.129	0.370	15.162 0.974	17.720 1.491	23.986 3.022	18.956 1.829	0.331
Curriculumformer	10.042 0.583	11.781 0.925	15.864 1.844	12.562 1.117	0.429	9.657 0.512	11.215 0.800	14.662 1.498	11.845 0.937	0.443	10.266 0.631	12.506 1.114	17.476 2.338	13.416 1.361	0.412
Curriculumformer + MaskPoint	10.208 0.601	12.077 0.984	16.327 1.982	12.871 1.189	0.422	9.575 0.500	11.011 0.762	**14.328** **1.436**	11.638 **0.899**	0.442	**10.136** 0.609	**12.264** 1.067	**17.152** 2.280	**13.184** 1.319	0.409
Curriculumformer + Point-MAE	10.117 0.592	12.085 1.002	16.458 2.034	12.887 1.209	0.425	9.717 0.520	11.242 0.800	14.747 1.546	11.902 0.955	0.440	10.334 0.648	12.503 1.116	17.509 2.353	13.449 1.372	0.407
Curriculumformer + Point-M2AE	**9.281** **0.476**	**10.711** **0.739**	**14.046** **1.427**	**11.344** **0.880**	**0.452**	**9.509** **0.491**	**10.971** **0.754**	14.410 1.469	**11.630** 0.905	0.441	10.415 **0.507**	12.550 **1.054**	17.555 **2.263**	13.507 **1.275**	0.388

TABLE VI

COMPARISON OF CURRICULUMFORMER FINE-TUNED ON MVP DATASET WITH OTHER TRAIN-FROM-SCRATCH NETWORKS REGARDING CD-$\ell_1 \times 10^3$, CD-$\ell_2 \times 10^3$, AND AVERAGE F-SCORE@1%

Methods	Airplane	Bed	Bench	Bookshelf	Bus	Cabinet	Car	Chair	Guitar	Lamp	Motorbike	Pistol	Skateboard	Sofa	Table	Watercraft	Avg.
	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2	F-Score/ CD-ℓ_1/ CD-ℓ_2
ASFMNet [54]	0.857 6.530 0.234	0.438 16.810 1.567	0.674 10.496 0.617	0.461 14.105 1.087	0.673 9.240 0.345	0.486 12.285 0.597	0.533 10.861 0.405	0.517 13.799 0.879	0.886 5.571 0.109	0.593 14.510 1.336	0.626 9.868 0.328	0.749 8.456 0.354	0.822 7.263 0.357	0.473 13.210 0.685	0.587 12.687 0.972	0.632 11.351 0.669	0.605 11.484 0.691
GRNet [56]	0.825 7.134 0.336	0.467 16.535 1.302	0.645 11.252 0.610	0.504 14.026 0.890	0.679 9.769 0.374	0.550 12.202 0.558	0.613 10.466 0.366	0.501 14.705 0.915	0.815 6.684 0.163	0.536 15.735 1.431	0.668 9.492 0.302	0.740 8.396 0.276	0.778 8.719 0.701	0.497 13.531 0.670	0.579 13.149 0.902	0.635 10.833 0.530	0.609 11.817 0.679
CRN [58]	0.891 5.495 0.190	0.527 18.480 2.139	0.704 11.440 0.853	0.626 12.509 1.018	0.718 9.507 0.405	0.674 10.064 0.404	0.734 8.641 0.264	0.656 11.367 0.625	0.644 12.745 0.899	0.665 12.111 0.954	0.623 11.346 0.525	0.594 16.773 1.974	0.700 11.478 0.758	0.639 10.825 0.488	0.726 10.389 0.724	0.732 9.186 0.436	0.696 10.579 0.651
TopNet [55]	0.747 7.979 0.286	0.327 16.529 1.023	0.595 11.043 0.493	0.347 14.796 0.779	0.618 9.797 0.316	0.420 12.617 0.508	0.511 11.053 0.370	0.342 14.895 0.741	0.812 6.760 0.153	0.336 16.765 1.189	0.532 10.868 0.372	0.638 9.492 0.321	0.754 7.875 0.218	0.364 13.988 0.599	0.501 12.929 0.674	0.485 12.537 0.637	0.492 12.357 0.584
FoldingNet [57]	0.739 8.144 0.339	0.309 15.940 1.022	0.604 10.948 0.636	0.407 12.359 0.561	0.705 8.650 0.255	0.496 11.246 0.412	0.572 10.180 0.314	0.334 14.668 0.754	0.848 6.255 0.120	0.320 17.878 1.536	0.511 10.928 0.358	0.706 8.576 0.268	0.782 9.418 1.267	0.392 13.041 0.570	0.518 11.966 0.572	0.515 11.923 0.563	0.516 11.881 0.615
PCN [59]	0.785 7.951 0.374	0.334 20.999 1.938	0.559 14.019 0.987	0.461 15.830 1.178	0.746 8.920 0.323	0.570 11.997 0.543	0.641 10.358 0.373	0.387 17.882 1.240	0.879 5.643 0.120	0.357 22.093 2.363	0.607 10.963 0.431	0.705 9.885 0.558	0.816 7.676 0.347	0.453 14.694 0.795	0.538 14.841 1.119	0.529 13.921 0.856	0.559 13.598 0.902
ECG [60]	0.843 6.189 0.185	0.644 11.956 0.859	0.804 7.850 0.375	0.707 9.870 0.520	0.814 7.055 0.216	0.681 9.635 0.385	0.726 8.180 0.243	0.682 10.168 0.522	0.779 6.734 0.179	0.714 10.711 0.881	0.784 7.448 0.209	0.841 6.552 0.235	0.900 5.542 0.156	0.649 10.002 0.429	0.759 9.032 0.506	0.750 8.450 0.394	0.740 8.753 0.418
PoinTr [61]	0.916 4.806 0.134	0.654 11.921 0.818	0.853 6.948 0.272	0.728 9.438 0.439	0.842 6.781 0.193	0.729 9.343 0.374	0.751 8.293 0.240	0.725 9.369 0.398	0.952 4.089 0.056	0.737 9.465 0.615	0.796 7.447 0.188	0.868 6.176 0.185	0.924 5.141 0.107	0.717 9.490 0.381	0.795 8.469 0.444	0.791 7.533 0.267	0.784 8.070 0.338
SnowflakeNet [62]	0.934 4.393 0.113	0.709 10.750 0.736	0.874 6.431 0.264	0.787 8.383 0.380	0.848 6.735 0.206	0.763 9.056 0.403	**0.766** 8.052 0.238	0.762 8.772 0.405	0.969 3.454 0.042	0.782 9.159 0.690	0.833 6.905 0.167	0.893 5.722 0.170	0.943 4.458 0.121	0.742 9.035 0.388	0.840 7.551 0.391	0.797 7.404 0.285	0.813 7.597 0.338
Curriculumformer	**0.949** **3.675** **0.070**	0.743 **8.641** 0.443	**0.900** 5.403 0.173	0.820 **7.275** 0.287	0.851 **6.119** 0.180	0.754 **8.406** 0.330	0.751 **7.540** **0.208**	0.803 **7.062** **0.260**	**0.980** **2.971** **0.032**	**0.866** **5.714** 0.250	0.844 6.268 **0.135**	0.902 5.001 0.120	0.950 4.048 0.084	0.759 **7.788** 0.286	0.854 **6.542** 0.279	**0.834** **5.966** 0.190	0.834 **6.369** 0.221
Curriculumformer + MaskPoint	0.943 3.885 **0.070**	0.722 8.690 **0.423**	0.883 5.599 0.172	0.786 7.584 0.295	0.824 6.428 **0.175**	0.709 8.697 **0.310**	0.716 7.964 0.229	0.768 7.371 0.269	0.970 3.236 0.040	0.843 5.899 **0.232**	0.825 6.499 0.143	0.891 5.194 0.122	0.935 4.277 0.087	0.728 7.982 **0.282**	0.829 6.686 **0.256**	0.822 6.106 **0.182**	0.825 6.381 **0.205**
Curriculumformer + Point-MAE	0.940 3.882 0.071	0.717 8.904 0.457	0.882 5.725 0.196	0.790 7.523 **0.267**	0.824 6.441 0.183	0.717 8.690 0.321	0.724 7.915 0.230	0.777 7.396 0.277	0.971 3.273 0.042	0.848 5.940 0.241	0.823 6.532 0.146	0.888 5.220 0.119	0.934 4.323 0.094	0.725 8.088 0.294	0.831 6.709 0.262	0.818 6.175 0.196	0.826 6.421 0.212
Curriculumformer + Point-M2AE	**0.949** 3.709 0.081	**0.745** 9.123 0.532	**0.900** **5.390** **0.166**	**0.821** 7.369 0.290	**0.861** 6.254 0.194	**0.771** 8.531 0.363	0.764 7.566 0.216	**0.809** 7.190 0.278	0.976 3.125 0.037	0.864 5.970 0.292	**0.851** **6.247** **0.135**	**0.905** **4.953** **0.113**	**0.959** **3.887** **0.062**	**0.770** 7.932 0.300	**0.857** 6.816 0.347	**0.837** 6.096 0.201	**0.840** 6.493 0.243

structures and high-level semantic patterns simultaneously, enabling critical point cloud representation learning.

Therefore, the overall objective for the former two stages is

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_w. \tag{7}$$

For the last stage of pre-training, the final objective should additionally include two M4CL losses mentioned in Section III-C, which could be formulated as follows:

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_w + \mathcal{L}_{\text{MIID}} + \mathcal{L}_{\text{MMID}}. \tag{8}$$

IV. Experiments

In this section, the transfer ability of Curriculumformer is evaluated on five widely used downstream applications, including classification, segmentation, and completion. We first show the implementation details of the pre-training and fine-tuning stages and then present the performance of Curriculumformer on the five downstream tasks.

A. Pre-Training Setups

1) Data Setups: ShapeNet [70], which contains more than 50 000 CAD models from 55 object categories, is utilized to pre-train our model. 1024 points from each 3-D instance are sampled and divided into 64 point patches, where each patch contains 32 points. A PointNet [26] is employed to project each patch into point embeddings, which is utilized as input of Curriculumformer. In the last pre-train stage, the rendered RGB images are collected from [40], owing 43 783 images from 13 classes. When given a point cloud, a 2-D image is randomly chosen among all rendered images, rendered from an arbitrary viewpoint. Further, we conduct data augmentation on the rendered image, including random crop, color jittering, and random horizontal flips.

2) Pre-Training Setups: In the pre-training experiments, the depth of the Transformer is set to 12, while the feature dimension is set to 384. The head's number is set to 6. A random depth of rate 0.1 is utilized in the encoder of the Transformer. In the pre-training stage, the weight of the Tokenizer learned by dVAE is fixed. Input embedding points are masked according to the pre-training stage. Then, the model is trained to predict the expected point labels for these masked tokens. On the MoCo, the size of the repository is set to 16 384, the temperature is set to 0.07, and the weight momentum is set to 0.999. AdamW optimizer is utilized with a 0.0005 initial learning rate and a 0.05 weight decay. Our model is pre-trained with 20, 20, and 10 epochs with 128 batch sizes for three pre-training stages, respectively. The total pre-training time is 19 h. On the other hand, the other existing self-supervised methods all need 300 epochs to pre-train the models. Noted that, in the last stage, the architecture of $\mathcal{F}_{\theta_D}$ is provided by Afham et al. [12], while the network $\mathcal{F}_{\theta_I}$ is from TorchVision [71] accompanied by an MLP serving as the head. $\mathcal{F}_{\theta_D}$ and $\mathcal{F}_{\theta_I}$ will be randomly initialized and trained during the last stage. Moreover, our experiments are mostly conducted on NVIDIA GeForce RTX 3090 GPU. Part segmentation and indoor segmentation experiments are performed on NVIDIA TITAN RTX.

B. Fine-Tuning Setups

1) Shape Classification: The Curriculumformer is fine-tuned on two datasets to validate the performance of shape classification. The most widely utilized ModelNet40 [72] contains synthetic 3-D objects of 40 classes, where 9843 samples are used for training with 2468 samples utilized for validation. Further, the more difficult dataset ScanObjectNN [53] consists of 11 416 samples for training and 2902 samples for validation from 15 classes, all of them are collected from the noisy real-world scans, determining the domain gaps with the pre-trained ShapeNet [72]. Specifically, ScanObjectNN is divided into three settings: OBJ-BG, OBJ-ONLY, and PB-T50-RS. In detail, 1024 and 2048 points are evenly sampled from each 3-D object of ModelNet40 and ScanObjectNN, respectively. The fine-tuning on the two datasets adopts the same settings, where the network is fine-tuned for 300 epochs together with a 32 batch size, a 5×10^{-4} learning rate, and a 5×10^{-2} weight decay. The other hyperparameters are kept the same as in the stage of pre-training.

2) Part Segmentation: In this experiment, ShapeNet-Part [73] dataset consists of 16 881 synthetic 3-D objects of 16 classes and 50 part classes. The training set and validation set contain 14 007 and 2874 instances, respectively. 2048 points are evenly sampled from each instance as the input of Curriculumformer, and the part categories will be predicted. Curriculumformer is fine-tuned for 300 epochs with a 16 batch size, a 2×10^{-4} learning rate, and a 0.1 weight decay. Other settings of training just keep the same with the experiments of shape classification.

3) Few-Shot Classification: Following the previous works [35], [37], "K-way N-shot" settings are carried out on ModelNet40 [5] for few-shot classification. And K out of

Fig. 4. Visualization of part segmentation results. Different colors can be ascribed to different parts of the objects. The top row depicts results predicted by Curriculumformer, while the bottom row shows the ground truth.

TABLE VII
Semantic Segmentation Results Are Presented for Area 5 of the S3DIS Dataset. The Evaluation Metrics Include mAcc and mIoU Across All Categories. Two Types of Input Features Are Utilized: xyz, Which Represents Point Cloud Coordinates, and xyz + rgb, Which Incorporates Both Coordinates and RGB Color Information

Methods	Input	mAcc (%)	mIoU (%)
PointNet [26]	xyz + rgb	49.0	41.1
PointNet++ [27]	xyz + rgb	67.1	53.5
PointCNN [28]	xyz + rgb	63.9	57.3
PCT [8]	xyz + rgb	67.7	61.3
Transformer [5]	xyz	68.6	60.0
Point-BERT [5]	xyz	69.7	60.5
Point-MAE [11]	xyz	69.9	60.8
Curriculumformer	xyz	**70.6**	**62.2**

Fig. 5. Visualization comparison of point cloud completion on PCN, MVP, ShapeNet55, and ShapeNetUnseen21 datasets.

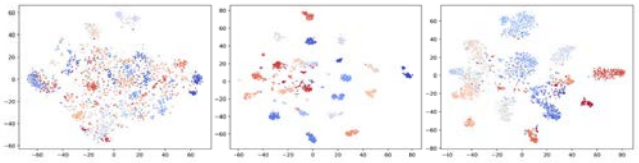

Fig. 6. Visualization of feature distributions. We visualize the features of test sets in ModelNet40 using t-distributed stochastic neighbor embedding (t-SNE). (a) Curriculumformer pre-trained on ShapeNet, (b) fine-tuning learned encoder of Curriculumformer on ModelNet40, and (c) ScanObjectNN.

40 categories and $N + 20$ 3-D objects per category are randomly selected, where N is for training and 20 for testing. The Curriculumformer is evaluated on four few-shot scenarios: five-way ten-shot, five-way 20-shot, ten-way ten-shot, and ten-way 20-shot, respectively. Further, ten independent runs under each setting are performed, and average accuracy together with standard deviations are reported to minimize the influence of the variance of random sampling. The fine-tuning settings are still just same as 3-D shape classification, but the epochs decrease to 150 epochs.

4) Point Cloud Completion: We utilize a standard Transformer encoder and a powerful Transformer-based decoder devised in SnowflakeNet [62]. We fine-tune our model on the point cloud completion benchmarks for 200 epochs.

PCN dataset [59], as the widely used benchmark, consists of 30 974 objects from eight classes. For each 3-D instance, 16 384 points are evenly sampled from the shape surface, and eight partial point clouds are collected via back-projecting 2.5-D depth images from eight views into 3-D. We utilize the same train/test split settings of PCN. MVP dataset [74] is a newly proposed dataset, where partial 3-D shapes are obtained from 26 uniformly distributed camera poses for each CAD model chosen from ShapeNet. The ground truth is collected through Poisson disk sampling (PDS) methods. And the multi-view partial point cloud dataset consists of 16 categories, with 62 400 pairs for training and 41 600 pairs for testing. The ShapeNet55 dataset, proposed by Yu et al. [61], leverages all objects from 55 categories in ShapeNet to ensure the diversity of objects. At the same time, ShapeNet34 dataset [61] is used to validate the generalization capabilities. ShapeNet34 can be divided into 34 seen classes and 21 unseen classes. Our model is fine-tuned on the seen classes and tested on both seen and unseen classes to verify the generalization performance. In detail, 8192 points are uniformly sampled from the object surface to obtain complete ground truth for each object in ShapeNet34/55 [61]. The viewpoints are arbitrarily set to enrich diversity, and n furthest points from each viewpoint are discarded to get the partial input. In the fine-tuning stage, n is randomly chosen from 2048 to 6144 (25%–75% of ground

TABLE VIII
3-D OBJECT DETECTION RESULTS ARE REPORTED ON THE VALIDATION SET OF SCANNET V2. OUR PRE-TRAINING MODEL AND POINT-BERT ADOPT 3DETR AS THE BACKBONE ARCHITECTURE. IN CONTRAST, OTHER METHODS UTILIZE VOTENET AS THE BACKBONE FOR FINE-TUNING. ONLY GEOMETRY INFORMATION IS UTILIZED AS INPUT FOR THE DOWNSTREAM TASK. THE "INPUT" COLUMN INDICATES THE INPUT TYPE DURING THE PRE-TRAINING STAGE, WHERE "xyz" REPRESENTS GEOMETRY INFORMATION

Methods	SSL	Pre-trained Input	AP_{25}	AP_{50}
VoteNet [63]		-	58.6	33.5
STRL [50]	✔	xyz	59.5	38.4
Implicit Autoencoder [64]	✔	xyz	61.5	39.8
RandomRooms [65]	✔	xyz	61.3	36.2
PointContrast [36]	✔	xyz	59.2	38.0
DepthContrast [35]	✔	xyz	61.3	-
3DETR [66]		-	62.1	37.9
Point-BERT [5]	✔	xyz	61.0	38.3
MaskPoint [51]	✔	xyz	63.4	40.6
Point-MAE [11]	✔	xyz	63.0	42.4
Curriculumformer	✔	xyz	**63.5**	**42.5**

truth), and the remaining points will be downsampled to 2048 points as inputs. In the evaluation stage, eight viewpoints are selected, and n is selected as 2048, 4096, or 6144 for convenience. In terms of the value of n, the test samples can be divided into three difficulties, including *simple*, *moderate*, and *hard*. We adopt ℓ_1, ℓ_2 Chamfer Distance, and F-score@1% for evaluation.

5) Indoor Segmentation: The S3DIS dataset, also known as the Stanford Large-Scale 3-D Indoor Spaces dataset [75], offers instance-level semantic segmentation for six expansive indoor areas. These areas consist of a total of 271 rooms and encompass 13 distinct semantic categories. Consistent with established conventions, we designated area 5 specifically for testing purposes, while utilizing the remaining areas for training our models.

6) Indoor Detection: The benchmark widely recognized for 3-D object detection is ScanNet V2 [76], which comprises 1513 indoor scenes and encompasses 18 distinct object classes. To ensure consistency, we adopt the evaluation procedure established by VoteNet [63], which calculates the mean average precision for two threshold values: 0.25 (mAP@0.25) and 0.5 (mAP@0.5). These metrics allow us to effectively evaluate the performance of our Curriculumformer.

C. Downstream Tasks

1) 3-D Object Classification: The Curriculumformer is fine-tuned the widely used ModelNet40 [72] and the challenging real-world ScanObjectNN [53]. Following Point-BERT [5], the voting tactic is utilized for fair comparison on ModelNet40 and ScanObjectNN. As shown in Table I, Curriculumformer achieves 93.6% accuracy on ModelNet40, surpassing Point-BERT fine-tuned by +0.4%. For ScanObjectNN in Table II, Curriculumformer performs better than the state-of-the-art Point-BERT by a great margin, +2.07%, +0.35%, and +1.45%, respectively, under the three settings, demonstrating the superiorities of Curriculumformer under complex circumstances. Moreover, due to the real-world scans of ScanObjectNN having large domain gaps with the pre-trained synthetic datasets, Curriculumformer also achieves a satisfactory transfer performance to understand point clouds from another domain.

Moreover, motivated by recent advancements in the field, such as the utilization of different Transformer backbones in approaches like MaskPoint [51], Point-MAE [11], and Point-M2AE [9], we aim to ensure a fair comparison by pre-training these methods within our Curriculumformer framework. Leveraging the benefits of our pre-training strategy, we are able to further enhance the performance of MaskPoint, Point-MAE, and Point-M2AE by 0.2, 0.2, and 0.1, respectively, on the ModelNet40 dataset. On the ScanObjectNN dataset, the effectiveness of our pre-training strategy in improving the performance of various backbones is demonstrated in Table II. Consequently, when using the same backbone, our pre-training strategy enables the acquisition of more comprehensive representations compared to relying solely on mask strategies.

2) Few-Shot Object Classification: Encouraged by the excellent performance on 3-D object classification, the experiments on few-shot classification [29], [35] are also conducted on ModelNet40 to validate the performance of Curriculumformer under the scenarios of limited fine-tuning samples. As we can see from Table III, Curriculumformer fulfills the best performance for three few-shot settings and outperforms Point-BERT by +2.4%, +1.5%, and +2.1%, respectively. Additionally, our pre-training strategy can further enhance the existing backbones, such as MaskPoint [51], Point-MAE [11], and Point-M2AE [9], which is shown in Table III. Furthermore, Curriculumformer also fulfills more minor deviations than other transformer-based methods, indicating the inherent universal 3-D representations of Curriculumformer for adapting well to the few-shot downstream tasks.

3) 3-D Object Part Segmentation: Apart from the classification tasks, the Curriculumformer is further evaluated for part segmentation on ShapeNetPart [73], which aims to predict per-point labels and needs a detailed understanding of local information. A straightforward segmentation head is employed to verify the effectiveness of our proposed pre-training tactic for well-capturing both high-level semantics and fine-grained local details. As reported in Table IV, Curriculumformer fulfills the best 85.9% instance mean intersection over union (mIoU), outperforming the second-best Point-BERT by +0.3%. Furthermore, the visualization of our segmentation performance is shown in Fig. 4, demonstrating that Curriculumformer can obtain satisfactory predictions with a very narrow difference from the ground truth. However, when combining our Curriculumformer and newly proposed methods, as shown in Table IV, it is found that the improvements of these backbones on the part segmentation task are marginal. The main reason can be ascribed to that our newly proposed upsampling and M4CL tasks all focus on global representations of objects compared to these methods, while the part segmentation task needs local representations. Therefore, when integrating the pre-training strategy focusing

TABLE IX

ABLATION STUDY ON VARIOUS PRE-TRAINING TASKS. WE VALIDATE THE EFFECTS OF DIFFERENT PRE-TRAINING DESIGNS AND REPORT THE CLASSIFICATION ACCURACY (%) AND PART SEGMENTATION (MIOU) AFTER FINE-TUNING ON MODELNET40 AND SHAPENETPART

	One-stage Up-sampling	Two-stage Completion	Three-stage M4CL	Acc.	mIoU_I
Model1	✔			92.66	85.49
Model2		✔		92.79	85.51
Model3			✔	92.91	85.41
Model4	✔	✔		93.03	85.37
Curriculumformer	✔	✔	✔	**93.59**	**85.90**

TABLE X

ABLATION STUDY ON VARIOUS PRE-TRAINING TASKS. WE VALIDATE THE EFFECTS OF DIFFERENT PRE-TRAINING DESIGNS AND REPORT CD-ℓ_1, CD-ℓ_2, AND F-SCORE@1% AFTER FINE-TUNING ON PCN

	One-stage Up-sampling	Two-stage Completion	Three-stage M4CL	CD-ℓ_1	CD-ℓ_2	F-Score@1%
Model1	✔			8.422	0.352	0.717
Model2		✔		8.346	0.381	0.730
Model3			✔	8.077	0.328	0.734
Model4	✔	✔		7.704	0.303	0.764
Curriculumformer	✔	✔	✔	**7.392**	**0.257**	**0.775**

TABLE XI

ABLATION STUDY ON THE SHUFFLE OF PRE-TRAINING TASKS

	Stage 1	Stage 2	Stage 3	Acc. (MN40)	Acc. (SONN)
Model A	3rd	2nd	1st	93.41	88.5
Model B	2nd	3rd	1st	93.08	87.8
Model C	3rd	1st	2nd	93.39	88.2
Model D	2nd	1st	3rd	93.48	88.6
Model E	1st	3rd	2nd	92.97	87.5
Curriculumformer	1st	2nd	3rd	**93.59**	**89.5**

TABLE XII

PERFORMANCE OF INCORPORATING OTHER PRE-TRAINING METHODS INTO OUR FRAMEWORK

	Stage1	Stage2	Stage3	MN40	SONN (OBJ-ONLY)
Model I	Up-sample	Completion	M4CL	**93.6**	**88.5**
Model II	OcCo	MAE	M4CL	93.2	87.9
Model III	Up-sample	Completion	M4CL-Depth	93.3	88.1
Model IV	OcCo	MAE	M4CL-Depth	92.7	87.5

TABLE XIII

PERFORMANCE OF INCORPORATING CLIP-BASED SSL METHODS INTO OUR FRAMEWORK

	ModelNet40	SONN		
		OBJ-BG	OBJ-ONLY	OBJ-BG
Curriculumformer	93.6	89.5	88.5	84.5
Curriculumformer+PointCLIP (En.)	94.1	91.7	89.9	86.9
Curriculumformer+ULIP	94.3	92.0	90.1	87.3

TABLE XIV

ABLATION STUDIES ON APPLYING TWO STAGES OF PRE-TRAINING SIMULTANEOUSLY. THE PERFORMANCE ON DOWNSTREAM MODELNET40 AND SCANOBJECTNN (OBJ-BG) IS REPORTED

	Up-sampling	Completion	M4CL	Acc. (MN40)	Acc. (SONN)
Model V		◊	◊	93.11	87.9
Model VI	◊		◊	92.83	87.5
Curriculumformer	1st	2nd	3rd	**93.59**	**89.5**

on local representations could further enhance the performance of part segmentation.

4) Point Cloud Completion: The reason because nearly all of the previous self-supervised learning methods solely concentrated on the discriminant capabilities of the representation learned by the network and evaluated it by transferring the pre-trained model to classification applications. The generative capability of the model is rarely studied [9], [77], [78], [79]. Therefore, we verify the transfer learning ability of Curriculumformer to point cloud completion. The Curriculumformer is evaluated on four datasets: PCN [59], MVP [74], ShapeNet55 [61], and ShapeNet34 [61], all of them are proposed to evaluate the performance on point cloud completion. The PCN is a widely used dataset with eight categories, while MVP is presented with more classes and viewpoints. The ShapeNet55 dataset utilizes all categories of the ShapeNet, and ShapeNet34 is usually performed to test the generalization capability. As shown in Fig. 5, the Curriculumformer is able to complete well all partial point clouds from the four datasets, superior to nearly all other supervised methods, such as PCN [59], GRNet [56], TopNet [55], even the state-of-the-art PoinTr [61], and SnowflakeNet [62]. Moreover, the quantitative results are shown in Tables V and VI and Table I in the Supplementary Material, the Curriculumformer fulfills the lowest CD-ℓ_1, CD-ℓ_2, and highest F-score@1% in all datasets, indicating the curriculum pre-training framework endows the Curriculumformer with excellent completion ability under various classes, viewpoints, and defect levels, as well as the generalization ability on unseen objects. Furthermore, our pre-training strategy can also tame MaskPoint [51], Point-MAE [11], and Point-M2AE [9] for the better point cloud completion since they are designed with robust backbones.

5) Indoor 3-D Semantic Segmentation: Moreover, we assess the performance of our proposed Curriculumformer in the context of 3-D semantic segmentation of large-scale scenes. This task is particularly challenging as it necessitates comprehension of both global semantics and local geometric details. The results of our experiment are presented in Table VII. Notably, our Curriculumformer exhibits a significant improvement compared to the Transformer trained from scratch, with a performance gain of 2.9% in mean accuracy (mAcc) and 3.7% in mIoU. This result serves as evidence that our Curriculumformer effectively enhances the Transformer's capabilities in addressing such demanding downstream tasks. Moreover, our Curriculumformer outperforms other self-supervised methods, achieving the highest performance by improving the mAcc and mIoU by 1.0% and 0.2%, respectively, in comparison to the second-best result obtained by Point-MAE. Even when compared to approaches that rely on scene geometric features and colors (as depicted in the top four methods in Table VII), our

Curriculumformer demonstrates comparable or even superior performance.

6) Indoor 3-D Object Detection: Furthermore, we proceed to evaluate the performance of our Curriculumformer on the 3-D object detection task, which necessitates methods with a robust understanding of large-scale scenes. To accomplish this, we conducted an experiment on the widely used real-world dataset, ScanNet V2. The results, presented in Table VIII, are measured in terms of AP_{25} and AP_{50}. Comparing the performance of both the methods trained from scratch and the pre-training methods, our approach achieves the highest AP_{25} and AP_{50} scores. Notably, our model outperforms the second-best method by attaining a 0.2% gain in AP_{25} and a 0.2% gain in AP_{50}.

V. Ablation Study and Analysis

A. Visualization Results

In order to further gain insight into the effectiveness of Curriculumformer, the learned features are visualized through t-SNE [80]. Fig. 6(a) shows our learned features before fine-tuning, where features are well separated, even trained with the absence of annotations, determining it is suitable for the initialization of the model. Fig. 6(b) and (c) gives the visualization of features fine-tuned on ModelNet40 and ScanObjectNN, where features form multiple clusters are quite separate from each other, demonstrating the effectiveness of Curriculumformer.

B. Effects on the Pre-Training Strategy

In order to gain a better understanding of the impact of pre-training tasks on the Transformer model, we conducted a thorough ablation study. The results of this study, as presented in Tables IX and X, demonstrate that Model 1, Model 2, and Model 3 were trained using self-supervised methods on single stages, resulting in relatively lower accuracy when compared to the Curriculumformer. When Model 4 is pre-trained on an upsampling pre-task followed by the completion task, it demonstrates superior performance compared to solely using the completion task. The results presented in Table XI demonstrate that pre-training the backbone in an easy-to-hard manner yields the best performance on downstream tasks. This highlights the importance of a cascaded pre-training approach in enhancing the representative learning ability of the Curriculumformer.

C. Versatility of Our Framework

1) Promoted Existing SOTA Methods via Our Pre-Training Framework: Since the recently proposed MaskPoint [51], Point-MAE [11], and Point-M2AE [9] own excellent performance, we systematically compare them with our method. Performance comparisons on ModelNet40, ModelNet40-FS, and ScanObjectNN are shown in Tables I–III, where the best methods are bold. Our method can obtain the second-best results under "five-way 20-shot" and "ten-way 20-shot" settings on the ModelNet40-FS dataset and OBJ-ONLY setting on ScanObjectNN compared with Point-MAE and Point-M2AE. The main reason for these results lies in that Point-MAE and Point-M2AE directly apply MAE on the tokenized point cloud, while our Curriculumformer follows Point-BERT to tokenize the point cloud by dVAE, leading to the less comparable performance. However, to validate the effectiveness of our curriculum pre-training pipeline, we integrate our pre-training framework into MaskPoint, Point-MAE, and Point-M2AE and test the results on the ModelNet40, ScanObjectNN, and ModelNet40-FS datasets. As expected, our M4CL can further promote the performance of existing self-supervised methods (see Tables I–III), demonstrating our curriculum pre-training pipeline can enhance representation learning of existing reconstruction-based methods.

TABLE XV
Ablation Study on Loss Term $\mathcal{L}_w$

	$\mathcal{L}_w$	Acc. (MN40)	Acc. (SONN)	CD-ℓ_1 (PCN)
Model VII		92.88	87.7	7.698
Curriculumformer	✔	**93.59**	**89.5**	**7.297**

TABLE XVI
Computational and Time Costs of Pre-Training

	Stage 1	Stage 2	Stage 3
Param.	21.1M	21.1M	23.2M
Pre-training Time	6.5 h/20 epoch	6.5 h/20 epoch	6 h/10 epoch

2) Incorporating Existing Pre-Training Methods Into Our Framework: In addition to promoting existing methods through our pre-training framework, we can also integrate other pre-training tactics such as OcCo [35], masked autoencoder (MAE) [81], and M4CL-Depth, which is derived from CLIP2Point [82] and DepthContrast [37]. In our framework, we perform pre-training on these tactics and subsequently fine-tune the pre-trained models using ModelNet40 and ScanObjectNN (OBJ-ONLY) datasets. As demonstrated in Table XII, our method (Model I) achieves superior performance compared to Model II, which is pre-trained using OcCo, MAE, and our M4CL in a cascaded manner. This improvement can be attributed to the fact that while OcCo and MAE primarily emphasize local representations, our upsampling pre-text task focuses on capturing global information. Additionally, we incorporated contrastive learning between point clouds and depth images, which we refer to as M4CL-Depth. However, due to the limited information content of depth images compared to RGB images, specifically in terms of edge information, the performance of downstream tasks (Model I, II versus Model III, IV) experienced a decline.

D. Incorporating Existing CLIP-Based SSL Methods Into Our Framework

Given the growing popularity of large foundation models, they offer a valuable means of transferring extensive knowledge to various domains. Consequently, this section aims to provide a comprehensive examination of the distinctions between our approach and other CLIP-based methods. As a pioneer work, PointCLIP [83] projected point clouds onto 2-D images and designed an inter-view adapter to better extract the

global feature and adaptively fuse the 3-D few-shot knowledge into CLIP pre-trained in 2-D. Building upon this progress, PointCLIP V2 [84] took a further step by integrating CLIP and GPT-3. This integration facilitated the transfer of pre-trained vision-language knowledge into 3-D domains. The primary advantage of our Curriculumformer is its ability to enhance the representation learning of our 3-D backbone by adopting their knowledge transfer methods. In other words, it has the capacity to continuously acquire knowledge similar to a human. As a means of verification, we ensemble PointCLIP into our framework following [83]. The findings are presented in Table XIII, indicating that the insights from PointCLIP can effectively enhance the performance of our Curriculumformer. This outcome highlights the versatility and synergy between our method and PointCLIP.

Additionally, ULIP [85] is another CLIP-based approach that aims to learn a 3-D representation space aligned with the shared image-text space. It achieves this by leveraging CLIP and a limited set of automatically generated triplets. One notable advantage of ULIP is its compatibility with various 3-D backbone networks, allowing seamless integration into any 3-D architecture. Thus, we integrate ULIP into our framework as the final task. As illustrated in Table XIII, the integration of ULIP greatly improves the performance of our Curriculumformer, thereby confirming the versatility of our approach.

E. Effects of Simultaneously Applying Pre-Training Strategies

To gain insights into the effects of concurrently employing pre-training strategies, additional ablation studies were conducted. However, it is important to note that the application of both upsampling and completion methods simultaneously is not feasible due to their reliance on the masking strategy. Hence, as depicted in Table XIV, we devised Model V, which was pre-trained using completion and M4CL, and Model VI, which was pre-trained using upsampling and M4CL. However, when fine-tuning these models on ModelNet40 and ScanObjectNN (OBJ-BG), they were unable to achieve performance comparable to our Curriculumformer. The primary reason for this phenomenon can be attributed to the need for careful exploration of loss combination when pre-training a model using multi-task learning [86].

F. Ablation Study on Loss Term $\mathcal{L}_w$

To conduct a thorough ablation study, we conducted further investigation into the effectiveness of the MoCo term $\mathcal{L}_w$. The results are presented in Table XV. When MoCo $\mathcal{L}_w$ is not utilized, Model VII exhibits a decline in performance when transferred to the ModelNet40, ScanObjectNN, and PCN datasets. This decline can be attributed to the fact that MoCo $\mathcal{L}_w$ aids the Transformers in acquiring a more proficient understanding of high-level semantics.

G. Analysis on Computational Costs

Since our Curriculumformer involves three stages of pre-training, the computational cost might be a main concern. Therefore, we detailedly list trainable parameters and pre-training time of three stages in Table XVI. The model initialization from the previous stage is excellent, thus requiring only a few iterations to train the model further in the current pre-training stage. Specifically, the parameters of pre-training models in the first two stages are both 21.1 M, while the pre-training time is also 6.5 h for 20 epochs. Further, the last pre-training stage only takes 6 h for 10 epochs and slightly improves the pre-training parameters due to the auxiliary 2-D and 3-D features extractor. In summary, we only need 50 epochs for pre-training in total, greatly less than existing methods [9], [11].

VI. Conclusion

In this article, we introduced Curriculumformer, a curriculum pre-training framework designed to enhance point cloud understanding by addressing the challenges of previous single pre-tasks that were incapable of learning both local and global geometries as well as 3-D–2-D correspondence. We conducted comprehensive experiments on various datasets and downstream tasks to validate the performance of Curriculumformer. The results demonstrate that Curriculumformer outperforms previous work across all downstream applications and becomes a new state-of-the-art method for point cloud completion. Our ablation studies also confirm that our cascaded pre-training strategy is a strength in transferring knowledge learned from unlabeled data to downstream tasks, highlighting the significant potential of Curriculumformer in point cloud understanding.

References

[1] B. Fei et al., "Comprehensive review of deep learning-based 3D point cloud completion processing and analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 22862–22883, Dec. 2022.

[2] B. Fei, J. Xu, R. Zhang, Q. Zhou, W. Yang, and Y. He, "3D Gaussian splatting as new era: A survey," *IEEE Trans. Vis. Comput. Graphics*, early access, 2024.

[3] 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.

[4] J. Zhou et al., "3D-OAE: Occlusion auto-encoders for self-supervised learning on point clouds," 2022, *arXiv:2203.14084*.

[5] 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. 19313–19322.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[7] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[8] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, no. 2, pp. 187–199, Jun. 2021.

[9] R. Zhang et al., "Point-M2AE: Multi-scale masked autoencoders for hierarchical point cloud pre-training," 2022, *arXiv:2205.14401*.

[10] K. He, R. Girshick, and P. Dollár, "Rethinking ImageNet pre-training," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4918–4927.

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

[12] 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.*, Sep. 2022, pp. 9902–9912.

[13] D. W. Shu and J. Kwon, "Hierarchical bidirected graph convolutions for large-scale 3-D point cloud place recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, 2023.

[14] C.-Q. Huang, F. Jiang, Q.-H. Huang, X.-Z. Wang, Z.-M. Han, and W.-Y. Huang, "Dual-graph attention convolution network for 3-D point cloud classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4813–4825, Apr. 2022.

[15] A.-D. Nguyen et al., "Single-image 3-D reconstruction: Rethinking point cloud deformation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 5, pp. 6613–6627, Nov. 2022.

[16] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7077–7087.

[17] Y. Bengio, J. Louradour, and R. Collobert, "Curriculum learning," in *Proc. Int. Conf. Mach. Learn.*, Aug. 2009, pp. 41–48.

[18] S. Guo et al., "Curriculumnet: Weakly supervised learning from large-scale web images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 135–150.

[19] C. Liu et al., "Curriculum learning for natural answer generation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 4223–4229.

[20] A. Graves, M. G. Bellemare, J. Menick, R. Munos, and K. Kavukcuoglu, "Automated curriculum learning for neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1311–1320.

[21] J. Guo, X. Tan, L. Xu, T. Qin, E. Chen, and T.-Y. Liu, "Fine-tuning by curriculum learning for non-autoregressive neural machine translation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 5, 2020, pp. 7839–7846.

[22] C. Gong, D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang, "Multi-modal curriculum learning for semi-supervised image classification," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3249–3260, Jul. 2016.

[23] S. Zakharov, W. Kehl, A. Bhargava, and A. Gaidon, "Autolabeling 3D objects with differentiable rendering of SDF shape priors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12221–12230.

[24] B. Fei, W. Yang, L. Ma, and W.-M. Chen, "DcTr: Noise-robust point cloud completion by dual-channel transformer with cross-attention," *Pattern Recognit.*, vol. 133, Jan. 2023, Art. no. 109051.

[25] B. Fei, W. Yang, W.-M. Chen, and L. Ma, "VQ-DcTr: Vector-quantized autoencoder with dual-channel transformer points splitting for 3D point cloud completion," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 4769–4778.

[26] C. R. Qi, H. Su, K. Mo, 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. 652–660.

[27] 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. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–10.

[28] 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.*, vol. 31, 2018, pp. 1–11.

[29] C. Sharma and M. Kaul, "Self-supervised few-shot learning on point clouds," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 7212–7221.

[30] B. Fei, Y. Li, W. Yang, L. Ma, and Y. He, "Towards unified representation of multi-modal pre-training for 3D understanding via differentiable rendering," 2024, *arXiv:2404.13619*.

[31] B. Fei et al., "Self-supervised learning for pre-training 3D point clouds: A survey," 2023, *arXiv:2305.04691*.

[32] O. Poursaeed, T. Jiang, H. Qiao, N. Xu, and V. G. Kim, "Self-supervised learning of point clouds via orientation estimation," in *Proc. Int. Conf. 3D Vis. (3DV)*, Nov. 2020, pp. 1018–1028.

[33] I. Achituve, H. Maron, and G. Chechik, "Self-supervised learning for domain adaptation on point clouds," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 123–133.

[34] J. Sauder and B. Sievers, "Self-supervised deep learning on point clouds by reconstructing space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[35] 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.

[36] 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.* Cham, Switzerland: Springer, 2020, pp. 574–591.

[37] Z. Zhang, R. Girdhar, A. Joulin, and I. Misra, "Self-supervised pretraining of 3D features on any point-cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10232–10243.

[38] Z. Zhang, B. Yang, B. Wang, and B. Li, "Growsp: Unsupervised semantic segmentation of 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Sep. 2023, pp. 17619–17629.

[39] Z. Chen et al., "PointDC: Unsupervised semantic segmentation of 3D point clouds via cross-modal distillation and super-voxel clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2023, pp. 14290–14299.

[40] C. Xu et al., "Image2Point: 3D point-cloud understanding with 2D image pretrained models," 2021, *arXiv:2106.04180*.

[41] 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.

[42] 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.

[43] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "PointWeb: Enhancing local neighborhood features for point cloud processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5565–5573.

[44] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "SpiderCNN: Deep learning on point sets with parameterized convolutional filters," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 87–102.

[45] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6411–6420.

[46] Y. Rao, J. Lu, and J. Zhou, "Global–local bidirectional reasoning for unsupervised representation learning of 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5376–5385.

[47] Y. Liu, B. Fan, G. Meng, J. Lu, S. Xiang, and C. Pan, "DensePoint: Learning densely contextual representation for efficient point cloud processing," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 5239–5248.

[48] C. Zhang, H. Wan, X. Shen, and Z. Wu, "PVT: Point-voxel transformer for point cloud learning," 2021, *arXiv:2108.06076*.

[49] 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. 16259–16268.

[50] 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. 6535–6545.

[51] H. Liu, M. Cai, and Y. J. Lee, "Masked discrimination for self-supervised learning on point clouds," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2022, pp. 657–675.

[52] 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.

[53] M. A. Uy, Q. Pham, B. Hua, T. Nguyen, and S. 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.

[54] Y. Xia, Y. Xia, W. Li, R. Song, K. Cao, and U. Stilla, "ASFM-Net: Asymmetrical Siamese feature matching network for point completion," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 1938–1947.

[55] L. P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, and S. Savarese, "TopNet: Structural point cloud decoder," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 383–392.

[56] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, "GRNet: Gridding residual network for dense point cloud completion," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 365–381.

[57] 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.

[58] X. Wang, M. H. Ang, and G. H. Lee, "Cascaded refinement network for point cloud completion with self-supervision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8139–8150, Nov. 2022.

[59] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 728–737.

[60] L. Pan, "ECG: Edge-aware point cloud completion with graph convolution," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4392–4398, Jul. 2020.

[61] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "PoinTr: Diverse point cloud completion with geometry-aware transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12498–12507.

[62] P. Xiang et al., "SnowflakeNet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Sep. 2021, pp. 5499–5509.
[63] C. R. Qi, O. Litany, K. He, and L. Guibas, "Deep Hough voting for 3D object detection in point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9277–9286.
[64] S. Yan et al., "Implicit autoencoder for point-cloud self-supervised representation learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2023, pp. 14530–14542.
[65] Y. Rao, B. Liu, Y. Wei, J. Lu, C.-J. Hsieh, and J. Zhou, "RandomRooms: Unsupervised pre-training from synthetic shapes and randomized layouts for 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3263–3272.
[66] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2906–2917.
[67] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6023–6032.
[68] J. Zhang et al., "PointCutMix: Regularization strategy for point cloud classification," *Neurocomputing*, vol. 505, pp. 58–67, Sep. 2022.
[69] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.
[70] A. X. Chang et al., "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.
[71] S. Marcel and Y. Rodriguez, "Torchvision the machine-vision package of torch," in *Proc. 18th ACM Int. Conf. Multimedia*, Oct. 2010, pp. 1485–1488.
[72] 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.
[73] L. Yi et al., "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph. (ToG)*, vol. 35, no. 6, pp. 1–12, 2016.
[74] L. Pan et al., "Variational relational point completion network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8524–8533.
[75] I. Armeni et al., "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1534–1543.
[76] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5828–5839.
[77] X. Zhao, B. Zhang, J. Wu, R. Hu, and T. Komura, "Relationship-based point cloud completion," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 12, pp. 4940–4950, Dec. 2022.
[78] Z. Zhu et al., "CSDN: Cross-modal shape-transfer dual-refinement network for point cloud completion," *IEEE Trans. Vis. Comput. Graphics*, 2023.
[79] Z. Yan, Z. Yi, R. Hu, N. J. Mitra, D. Cohen-Or, and H. Huang, "Consistent two-flow network for tele-registration of point clouds," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 12, pp. 4304–4318, Dec. 2022.
[80] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.
[81] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16000–16009.
[82] T. Huang, "CLIP2Point: Transfer clip to point cloud classification with image-depth pre-training," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2023, pp. 22157–22167.
[83] R. Zhang et al., "PointCLIP: Point cloud understanding by CLIP," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8552–8562.
[84] X. Zhu et al., "PointCLIP V2: Prompting CLIP and GPT for powerful 3D open-world learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2023, pp. 2639–2650.
[85] L. Xue et al., "ULIP: Learning a unified representation of language, images, and point clouds for 3D understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 1179–1189.
[86] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7482–7491.

Ben Fei (Graduate Student Member, IEEE) received the M.S. degree from the Department of Material Science, Fudan University, Shanghai, China, in 2021, where he is currently pursuing the Ph.D. degree with the School of Computer Science.

His research interests include generative models, point cloud processes, transformers, and 3-D vision.

Mr. Fei is an IEEE Young Professional.

Tianyue Luo received the B.S. degree from the Software School, Fudan University, Shanghai, China, in 2021, where she is currently pursuing the M.S. degree with the School of Computer Science.

Her research interests include point cloud understanding and 3-D reconstruction.

Weidong Yang (Member, IEEE) received the Ph.D. degree in software engineering from Xidian University, Xi'an, China, in 1999.

He was a Post-Doctoral Researcher with the School of Computer Science, Fudan University, Shanghai, China, from 1999 to 2001, where he is currently a Professor. His research interests include big data, knowledge engineering, database and data mining, and software engineering.

Liwen Liu received the B.S. degree in software engineering from Northeast Forestry University (NEFU), Harbin, China, in 2021. She is currently pursuing the master's degree in computer science with Fudan University, Shanghai, China.

Her research interests include point cloud completion.

Rui Zhang (Graduate Student Member, IEEE) received the bachelor's degree from the School of Management, Jilin University, Changchun, China, in 2020. He is currently pursuing the M.S. degree with the School of Computer Science, Fudan University, Shanghai, China.

His research interests include point cloud completion, 3-D reconstruction, and 3-D generation.

Ying He (Member, IEEE) is currently an Associate Professor with the College of Computing and Data Science and serves as the Director of the Centre for Augmented and Virtual Reality, Nanyang Technological University, Singapore. His research primarily focuses on geometric computing and analysis.

Dr. He is an active member of the technical program committees of major conferences in geometric modeling and has held positions on the editorial boards of leading journals, including IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, *Computer Graphics Forum*, and *Computational Visual Media*. His leadership roles have included the General or Program Co-Chair for several prominent conferences, such as the Shape Modeling International Conference in 2022, the Symposium on Solid and Physical Modeling in 2022 and 2023, the Geometric Modeling and Processing Conference in 2014 and 2021, and the Conference on Computational Visual Media in 2020.