

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/169617/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Chen, Wu, Jiang, Qiuping, Zhou, Wei, Xu, Long and Lin, Weisi 2024. Dynamic hypergraph convolutional network for no-reference point cloud quality assessment. *IEEE Transactions on Circuits and Systems for Video Technology* 34 (10) , pp. 10479-10493. 10.1109/TCSVT.2024.3410052

Publishers page: <http://dx.doi.org/10.1109/TCSVT.2024.3410052>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See <http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



Dynamic Hypergraph Convolutional Network for No-Reference Point Cloud Quality Assessment

Wu Chen, Qiuping Jiang, *Senior Member, IEEE*, Wei Zhou, *Senior Member, IEEE*,
Long Xu, *Senior Member, IEEE*, and Weisi Lin, *Fellow, IEEE*

Abstract—With the rapid advancement of three-dimensional (3D) sensing technology, point cloud has emerged as one of the most important approaches for representing 3D data. However, quality degradation inevitably occurs during the acquisition, transmission, and process of point clouds. Therefore, point cloud quality assessment (PCQA) with automatic visual quality perception is particularly critical. In the literature, the graph convolutional networks (GCNs) have achieved certain performance in point cloud-related tasks. However, they cannot fully characterize the nonlinear high-order relationship of such complex data. In this paper, we propose a novel no-reference (NR) PCQA method with hypergraph learning. Specifically, a dynamic hypergraph convolutional network (DHCN) composing of a projected image encoder, a point group encoder, a dynamic hypergraph generator, and a perceptual quality predictor, is devised. First, a projected image encoder and a point group encoder are used to extract feature representations from projected images and point groups, respectively. Then, using the feature representations obtained by the two encoders, dynamic hypergraphs are generated during each iteration, aiming to constantly update the interactive information between the vertices of hypergraphs. Finally, we design the perceptual quality predictor to conduct quality reasoning on the generated hypergraphs. By leveraging the interactive information among hypergraph vertices, feature representations are well aggregated, resulting in a notable improvement in the accuracy of quality prediction. Experimental results on several point cloud quality assessment databases demonstrate that our proposed DHCN can achieve state-of-the-art performance. The code will be available at: <https://github.com/chenwuwq/DHCN>.

Index Terms—Point clouds, dynamic hypergraph, no-reference, quality assessment.

I. INTRODUCTION

THE rapid development of three-dimensional (3D) sensing technology promotes the wider applications of 3D vision in many fields, such as autonomous driving [1]–[3], virtual reality [4], augmented reality [5], point cloud registration [6]–[8], etc. Such 3D visual data is typically represented in the forms of point clouds, voxels [9], or meshes [10]. Since the point cloud can describe 3D objects completely and accurately, it is regarded as an effective method for 3D data representation.

This work was supported in part by the Natural Science Foundation of Zhejiang (LR22F020002), in part by the Natural Science Foundation of China (62271277), and in part by the Natural Science Foundation of Ningbo (2022J081).

W. Chen, Q. Jiang, and L. Xu are with the School of Information Science and Engineering, Ningbo University, Ningbo 315211, China (e-mail: jiangqiuping@nbu.edu.cn).

W. Zhou is with the School of Computer Science and Informatics, Cardiff University, Cardiff, United Kingdom (e-mail: zhouw26@cardiff.ac.uk)

W. Lin is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (wslin@ntu.edu.sg)

Corresponding author: Qiuping Jiang

A point cloud contains a set of 3D points, each of which involves geometry information and several attributes, e.g., color, normal vector, and so on [11]. Although point clouds play a significant role in many fields due to their rich and realistic representation, the large number of point sets makes it difficult to achieve efficient storage and transmission without point cloud compression [12]–[14]. Therefore, it is necessary to encode, downsample, and simplify the point clouds before transmission, which inevitably leads to distortions. To better quantify the distortion level of point clouds, designing efficient point cloud quality assessment (PCQA) methods are in urgent need.

In general, PCQA methods are classified into subjective and objective aspects, depending on whether subjects are involved. The human visual system (HVS) can make accurate subjective evaluation of point clouds with many kinds of distortions such as downsampling, Gaussian noise, Octree-based compression, etc. However, conducting subjective experiments are expensive and labor-intensive. Thus, objective models are developed in the literature, which usually adopt subjective quality scores as ground truth labels to imitate the quality perception process of the HVS [15]. Similar to conventional image quality assessment (IQA), objective PCQA models include full-reference (FR), reduced-reference (RR), and no-reference (NR) ones based on whether reference point clouds are employed. Moreover, since the reference point cloud is not always available, NR-PCQA is more suitable for practical applications.

Typically, NR-PCQA can be divided into traditional methods [16]–[20] and deep learning-based methods [21]–[32]. Among them, traditional methods directly extract hand-crafted features (e.g., geometry and color) of distorted point clouds, and then predict the visual quality score with conventional regression algorithms such as support vector regression (SVR), random forest (RF), etc. Most deep learning-based methods build convolutional neural networks (CNNs) for automatic feature learning in a data-driven manner. Despite that CNNs are very powerful in solving various computer vision tasks [33]–[35], they cannot handle non-Euclidean data well [36].

To overcome the above-mentioned problem, graph convolutional networks (GCNs) have been applied as an alternative to CNNs [37] for processing the non-Euclidean data. GCNs allow graphs to be used as input to the network, extending CNNs applications to non-Euclidean data with surprising results. As non-Euclidean data, point cloud typically contains rich geometric and topological information, with many-to-many point-to-point relationships. However, the graph only considers

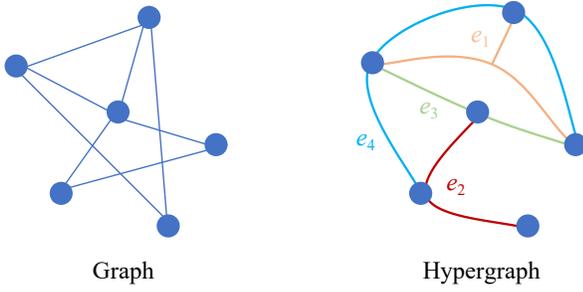


Fig. 1. Comparison of graphs and hypergraphs in which e_1, e_2, e_3, e_4 with different colors represent four different hyperedges.

the low-order correlation between data, and an edge of the graph can only connect two vertices, which cannot fully describe the nonlinear high-order relationship among data. To this end, researchers have further proposed hypergraphs, as shown in Fig. 1. Unlike the conventional graph, each hyperedge of a hypergraph can connect any number of vertices, and the edge degree of the hypergraph can be much higher than the edge degree of the graph. Consequently, the hypergraph extends the data to a higher-dimensional and more complex nonlinear space, which can more flexibly model the high-order correlation among the point cloud data [38]. In the literature, hypergraph has been widely used in the field of computer vision, such as video segmentation [39], 3D object classification [40], [41], image retrieval [42], [43], and visual tracking [44], etc.

Inspired by the high-order modeling capability of hypergraphs for non-Euclidean data, we propose a novel Dynamic Hypergraph Convolutional Network (DHCN) for NR-PCQA. Our proposed method introduces hypergraph learning into the PCQA task for the first time. In our proposed DHCN, there are two key issues to be addressed: 1) how to construct effective hypergraphs to model the complex interaction among vertices; 2) how to perform hypergraph learning to extract and aggregate features from the hypergraphs for quality prediction. The main contributions of this work are as follows:

- 1) We devise a novel NR-PCQA method based on DHCN which can fully utilize of the high-order modeling capability of hypergraphs for non-Euclidean data, effectively learning intricate data patterns and mapping feature information to target scores. To the best of our knowledge, we are the first to introduce hypergraph learning for the challenging PCQA task. In addition, we simultaneously combine the advantages of 2D projection information and 3D original information to achieve more precise quality prediction of point clouds through multi-modal learning.
- 2) We propose a dynamic hypergraph generator that takes into account multiple aspects of data correlation, such as spatial angular distance, radial basis kernel distance, feature reconstruction correlation, and cosine similarity, to generate higher quality hypergraph structures. Our devised hypergraph generator is flexible, allowing future update of more efficient methods. The hypergraph structure is continuously updated in each iteration, en-

suring that the interaction information remains up-to-date throughout the process.

- 3) Experimental results substantiate the superiority of our proposed DHCN method over state-of-the-art NR-PCQA methods, as it achieves remarkable performance gains on three widely used PCQA benchmark databases. The ablation study results prove that hypergraphs play an important role in our method. In addition, cross-database validation and computational efficiency experiments show that our method has better generalization and computational efficiency.

The rest of this paper is organized as follows. In Section II, we provide an overview of the related work. Section III presents the technical details of our proposed DHCN method. In Section IV, we perform extensive experiments and analyze the results. Finally, we draw conclusions in Section V.

II. RELATED WORK

According to the type of data used by PCQA methods, existing PCQA methods are mainly divided into PC-based and projection-based methods. The PC-based methods directly operate on the raw point cloud by extracting various features such as geometric and color attributes of the point cloud to assess its quality score. On the other hand, the projection-based methods perform quality assessment of the projected images generated from the corresponding point cloud.

A. PC-based Methods

At the beginning of the development of PCQA methods, Mekuria and Tian et al. proposed some point-based quality assessment methods, such as $PSNR_{MSE, p2po}$ and $PSNR_{HF, p2po}$ [45], $PSNR_{MSE, p2pl}$ and $PSNR_{HF, p2pl}$ [46]. After that, $PSNR_{\gamma}$ [47] is further proposed to evaluate the texture distortion of color point clouds. Alexiou et al. [48] employed the angular similarity between corresponding points to capture the degradation of the distorted point cloud. Meynet et al. [49] proposed the PC-MSDM based on local curvature statistics for quality assessment of point clouds. Javaheri et al. [50] used the generalized Hausdorff distance as an important metric for PCQA. Furthermore, color information has also garnered significant attention. Based on PC-MSDM, Meynet et al. [51] added color attributes to facilitate prediction. Viola et al. [52] use global color statistics to assess the degree of distortion of point clouds. In their work, Alexiou et al. [53] employed a structural similarity index based on geometric and color features. In addition, Diniz et al. [54]–[56] proposed a method for extracting statistical information from point clouds using local binary pattern (LBP) descriptors and local luminance pattern (LLP) descriptors. Then, the quality evaluation is carried out based on the statistical information. Yang et al. [57], [58] measure the quality score using the local graph representation and the multiscale potential energy difference (MPED) of the point cloud. When a complete reference point cloud is unavailable, the RR method plays an important role. Viola et al. [59] proposed a new RR measure to find the best combination of features such as geometry, brightness, and normal through a linear optimization

algorithm. Liu et al. [60] similarly proposed an RR method to predict the quality of V-PCC compressed point clouds with geometric and color features. Su et al. [61] proposed an RR point cloud quality assessment method based on SVR. While the above methods have achieved a certain success, it is important to note that the reference point cloud is not always available in real scenarios. Thus, Zhang et al. [17] propose a NR method that utilizes 3D natural scene statistics (3D-NSS) to extract geometric and color features and uses SVR for quality prediction. Zhou et al. [18] proposed a blind quality assessment method for point clouds based on structure-guided resampling. Similar to other tasks, the introduction of deep learning has also prompted further development of PCQA. Chetouani et al. [21] create a mapping from feature representations to quality scores using a deep neural network (DNN). Liu et al. [22] proposed a NR-PCQA method that utilizes sparse convolutional neural network (SCNN). Shan et al. [23] proposed a multi-task graph convolutional network (GPA-Net) to realize no-reference PCQA. Wang et al. [24] proposed a non-local geometric and color gradient aggregation graph model, also employing a multi-task learning module. Tliba et al. [25] proposed a PCQA metric based on a dynamic graph convolutional neural network (DGCNN) that operates directly on the entire point cloud. More recently, Su et al. [19] developed bitstream-based NR model for perceptual quality assessment of point clouds. Liu et al. [20] proposed a NR bitstream-layer model for the quality assessment of V-PCC encoded point clouds.

B. Projection-based Methods

Projection-based methods first project the 3D point cloud onto a 2D image plane, and then directly evaluate the quality of the projected images. As a result, existing IQA methods can be used, including SSIM [62], MS-SSIM [63], IW-SSIM [64], VIFP [65], and more. Additionally, Hua et al. [16] proposed a blind quality evaluator for colored point clouds based on visual perception, which extracts geometric, color, and joint features by projecting the point cloud onto a plane. Freitas et al. [66] proposed a novel FR-PCQA metric to accurately estimate the quality of the point cloud by fusing geometric and texture information in projected texture maps. According to the content based saliency projection, Zhou et al. [67] proposed an RR point cloud quality metric named RR-CAP. Deep learning techniques and NR methods have also received extensive attention in projection-based methods. Tao et al. [26] proposed a multi-scale feature fusion network that weights the quality scores of local patches in a graph. Inspired by the multi-task model, Liu et al. [27] used DNN to extract multi-view features and then predict the probability vector and quality score vector. Tu et al. [28] proposed a two-stream CNN that is specifically designed to extract features from both texture projection maps and geometric projection maps. Yang et al. [29] introduced unsupervised adversarial domain adaptation into PCQA to solve the quality assessment problem by adapting knowledge from the source domain to the target domain. More recently, Xie et al. [30] proposed a projection-based multi-modal learning blind quality metric,

TABLE I
COMPARISON OF THE MAIN CHARACTERISTICS (CHS) OF THE EXISTING PCQA METRICS AND OUR PROPOSED METHOD. WE MAINLY COMPARE THEM FROM THREE ASPECTS: METHOD TYPE, INPUT DATA TYPE, AND QUALITY PREDICTOR TYPE.

PCQA methods	Method type	Input data type	Quality predictor type
<i>PSNR_{RMSSE,p2po}</i> [45]	FR	Point cloud	Distance metric
<i>PSNR_{RHF,p2po}</i> [45]			Distance metric
<i>PSNR_{RMSSE,p2pl}</i> [46]			Distance metric
<i>PSNR_{RHF,p2pl}</i> [46]			Distance metric
<i>PSNR_Y</i> [47]			Distance metric
Alexiou et al. [48]			Similarity metric
PC-MSDM [49]			Distance metric
Javaheri et al. [50]			Distance metric
PCQM [51]			Distance metric
Viola et al. [52]			Distance metric
Alexiou et al. [53]		Similarity metric	
Diniz et al. [54]-[56]		Distance metric	
GraphSIM [57]		Similarity metric	
MPED [58]		Distance metric	
SSIM [62]		Distance metric	
MS-SSIM [63]		Distance metric	
IW-SSIM [64]		Distance metric	
VIFP [65]		Distance metric	
Freitas et al. [66]		Similarity metric	
PCMR [59]		RR	Point cloud
R-PCQA [60]	Projected images		Linear model
PCQAML [61]	RR	Point cloud	Support Vector Regression (SVR)
RR-CAP [67]		Projected images	Distance metric
3D-NSS [17]	NR	Point cloud	SVR
SGR [18]			SVR/Random Forest (RF)
Chetouani et al. [21]			Convolutional Neural Network (CNN)
ResSCNN [22]			Sparse CNN (SCNN)
GPA-Net [23]			Graph convolutional network
Wang et al. [24]		Graph convolutional network	
PCQA-GraphPoint [25]		Graph convolutional network	
streamPCQ [19]		Linear model	
bitstreamPCQ [20]		Linear model	
BQE-CVP [16]		Projected images	RF
PM-BVQA [26]			CNN
PQANet [27]			CNN
Tu et al. [28]			CNN
IT-PCQA [29]			CNN
pmBQA [30]		NR	Both
GMS-3DQA [31]	Transformer		
MM-PCQA [32]	Ours	Both	CNN
Ours			Hypergraph convolutional network

achieving multi-modal feature fusion through graphs. Zhang et al. [31] proposed a 3D model quality assessment based on multi-projection grid mini-patch sampling to reduce resource consumption. In addition to single-modal methods, Zhang et al. [32] extracted multi-modal information from 3D point clouds for better perceptual quality.

We show the main features of the existing PCQA metrics and our proposed method in Table I. PC-based methods allows a more comprehensive understanding of the geometric and spatial features, while projection-based methods can well take the advantage of the existing well-established IQA metrics. However, these two methods fail to characterize the potential higher-order correlations that exist between the groups of points within the point cloud as well as between different projected images, and there are few methods that utilize both simultaneously. On the other hand, compared with the traditional methods based on distance metrics and similarity metrics, learning-based method automatically learns the feature representation from the data in a data-driven way and performs better in dealing with complex data. Therefore, in this paper, we propose a novel NR-PCQA method, which uses a hybrid-based method to extract richer and more accurate visual information from both point clouds and projected images. Additionally, the implicit higher-order correlations of the above data are effectively modelled by DHCN.

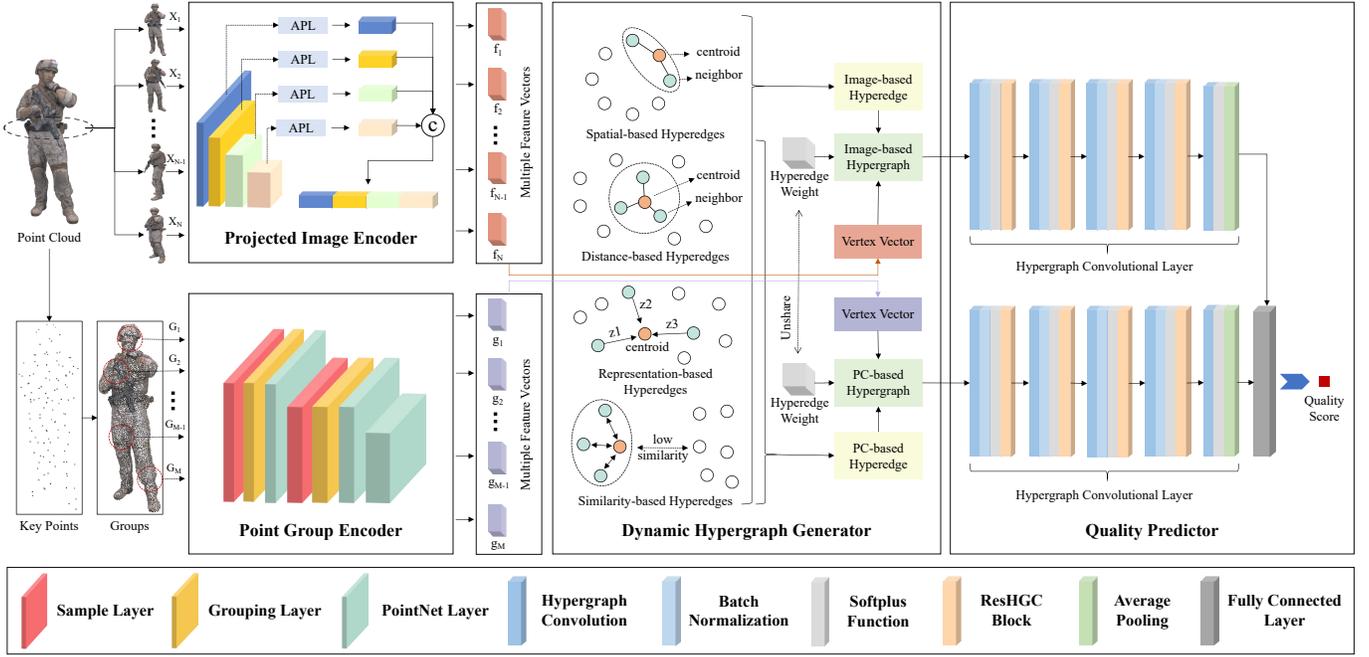


Fig. 2. Framework of the proposed DHCN method. The whole framework primarily consists of a projected image encoder, a point group encoder, a dynamic hypergraph generator, and a quality predictor. In the front end of the framework, the point cloud is pre-processed to obtain projection images and point groups, which are then input into the corresponding encoder, respectively. The feature representations generated by the encoder will be used as vertex vectors for dynamic hypergraph learning.

III. PROPOSED METHOD

In this section, we first provide an overview of the proposed DHCN method. Then, its key components and functions are described in detail. Finally, the details of the training process will be presented.

A. Overview

The framework of our proposed DHCN method is depicted in Fig. 2. It can be seen that DHCN is mainly composed of a projected image encoder, a point group encoder, a dynamic hypergraph generator, and a quality predictor. Suppose we have a point cloud $\mathbf{P} = \{p_x, p_y, p_z\} \in \mathbb{R}^T \times 3$, where T represents the total number of points in the point cloud, p_x, p_y, p_z represents the geometric coordinates of each point in 3D space. Here, we first resample and project \mathbf{P} to generate N projected images and M point groups as the input of our network. Then, a projected image encoder and a point group encoder are used to perform feature extraction on the projected images and the point groups, respectively. After that, given the feature representations obtained by the dual encoders, the hypergraphs are dynamically generated during each iteration. Finally, the quality predictor aggregates the feature representation based on the interaction information between the vertices of the hypergraph, and combines the two branches to jointly perceive the quality.

B. Multi-modal Data Feature Extraction

Point clouds serve as data carriers to describe 3D objects realistically through a vast collection of point sets. This inevitably leads to a strong demand for point cloud storage

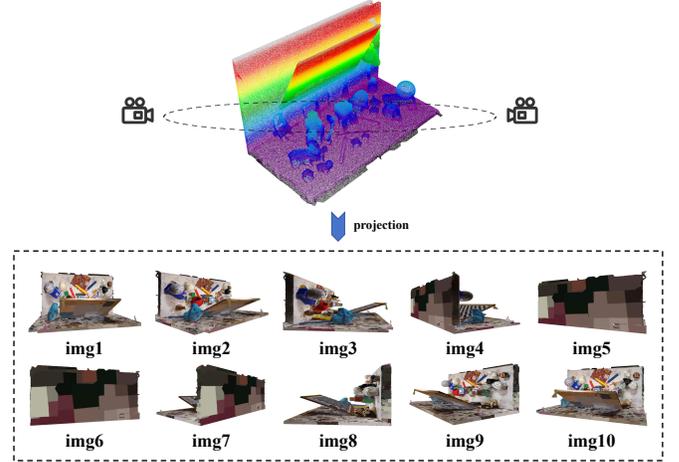


Fig. 3. Illustration of the point cloud projection process to obtain projection images with different viewpoints.

space. To solve this problem, various resampling [68]–[71] and projection [26]–[29] strategies have been proposed to reduce the point cloud cost. As discussed in the preceding section, our method chooses resampling and projection to simplify the point cloud.

1) *Projected Image Encoder*: Inspired by the HVS perception process of 3D objects, we utilize multi-view rotational projection to generate projected images. Similar to the panoramic perception [72] of the human eye, we apply fixed strides to rotate the point cloud, obtaining different views of the object, as shown in Fig. 3. Ultimately, we obtain N

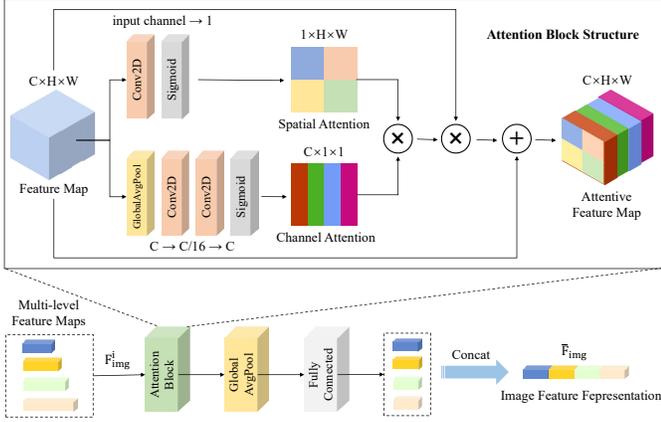


Fig. 4. Architecture of the MAFF module in the projected image encoder.

projected images, defined as:

$$\mathbf{I} = \{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_{N-1}, \mathbf{I}_N\} \in \mathbb{R}^{N \times 3 \times H \times W}, \quad (1)$$

where \mathbf{I}_i denotes the i^{th} projected image, H and W denotes the height and width of the each projected image. For projected images \mathbf{I} , we construct a projected image encoder to extract the deep feature representation. Inspired by the success of deep learning in PCQA, we adopt the pre-trained ResNet [73] as the projected image encoder to better model the image feature representation using multi-level feature. The specific process of multi-level attention feature fusion (MAFF) is shown in Fig. 4 in detail. Among them, MAFF module contains attention block, global average pooling and fully connected (FC) layer. Inspired by the human visual, the attention mechanism [74]–[76] in DNNs directs the network’s focus towards significant regions of the target. Assume that the multi-level feature map output by the projected image encoder is defined as $\mathbf{F}_{img}^i \in \mathbb{R}^{C^i \times H^i \times W^i}$, where i represents the feature map output from layer i^{th} of the projected image encoder, and at the same time reflects that the feature maps of different levels have different shapes. The attentive feature map obtained after attention block can be expressed as:

$$\hat{\mathbf{F}}_{img}^i = (\delta(\phi(\mathbf{F}_{img}^i)) \times \delta(\phi(\text{avg}(\mathbf{F}_{img}^i)))) \odot \mathbf{F}_{img}^i + \mathbf{F}_{img}^i, \quad (2)$$

where avg denotes global average pooling, ϕ denotes the 2D convolution, $\delta(\cdot)$ denotes the sigmoid function, and \odot denotes pixel-by-pixel multiplication. Finally, each corresponding attentive feature map $\hat{\mathbf{F}}_{img}^i$ is generated. Subsequently, the image feature representation $\bar{\mathbf{F}}_{img}$ is obtained by concatenating the attentive feature maps, which can be formulated as follows:

$$\bar{\mathbf{F}}_{img} = \iota(\text{avg}(\hat{\mathbf{F}}_{img}^1)) \cup \iota(\text{avg}(\hat{\mathbf{F}}_{img}^2)) \cup \iota(\text{avg}(\hat{\mathbf{F}}_{img}^3)) \cup \iota(\text{avg}(\hat{\mathbf{F}}_{img}^4)), \quad (3)$$

where ι represents the FC layer for dimensionality reduction, and \cup is the concatenation operator.

2) *Point Group Encoder*: We adopt the farthest point sampling (FPS) [68] strategy to capture M key points from the point cloud. Next, we employ the K-nearest neighbor (KNN) [77] algorithm to identify K neighboring points in close proximity to each key point, thus forming the point group denoted

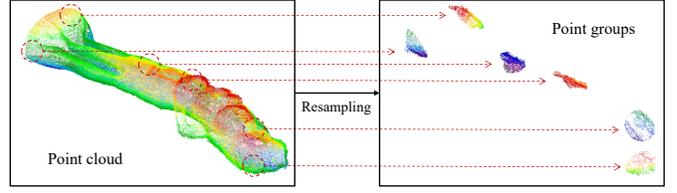


Fig. 5. Illustration of the resampling process to obtain point groups with diverse shapes from the distorted point cloud.

as $g_i (i \in \{1, 2, \dots, M\})$, as shown in Fig. 5. As can be seen from the figure, we observe that these point groups come from different positions of the point cloud, exhibiting diverse shapes, which can describe more comprehensive information with fewer points. Eventually, we can obtain M point groups, defined as:

$$\mathbf{G} = \{g_1, g_2, \dots, g_{M-1}, g_M\} \in \mathbb{R}^{M \times K}, \quad (4)$$

We adopt PointNet++ [78] as the point group encoder, which is specifically designed to capture 3D point set features. The encoder consists of three layers with distinct functions, namely sampling layer, grouping layer, and PointNet layer. The sampling layer utilizes FPS to downsample the point group, effectively reducing the size of the input point group. Based on the key points sampled in the previous step, the grouping layer employs the ball query algorithm to find neighbors of fixed size. Finally, the PointNet layer performs the characterization of the local neighborhood, producing the final representation of the point group.

C. Dynamic Hypergraph Generator

To capture the correlation between different data effectively, we construct a hypergraph. A hypergraph denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$, consisting of three parts: vertices \mathcal{V} , hyperedges \mathcal{E} , and hyperedge weights \mathcal{W} . We treat different projected images or point groups as vertices, with the feature representations generated by the encoder as vertex vectors, thereby capturing relevant points according to different strategies to generate hyperedges. A hyperedge can connect multiple vertices, and there is a many-to-many relationship between the vertices. In addition, each hyperedge is assigned a weight to represent its importance in the whole hypergraph. To generate higher quality hypergraph structures, we employ three strategies for two types of data, including distance-based methods [79], [80], representation-based methods [81], [82], and similarity-based methods. For the projected images, we additionally adopt spatial-based methods to assess the correlation based on the angular distance between different viewports. Instances of these methods are visible in the dynamic hypergraph generator of Fig. 2. Below, we describe these four methods in detail by using N projected images as an example.

1) *Spatial-Based Method*: Spatial-based methods generate spatial-based hyperedges \mathcal{E}_{spa} based on the position of the viewport in space. When we take the i^{th} vertex as the centroid,

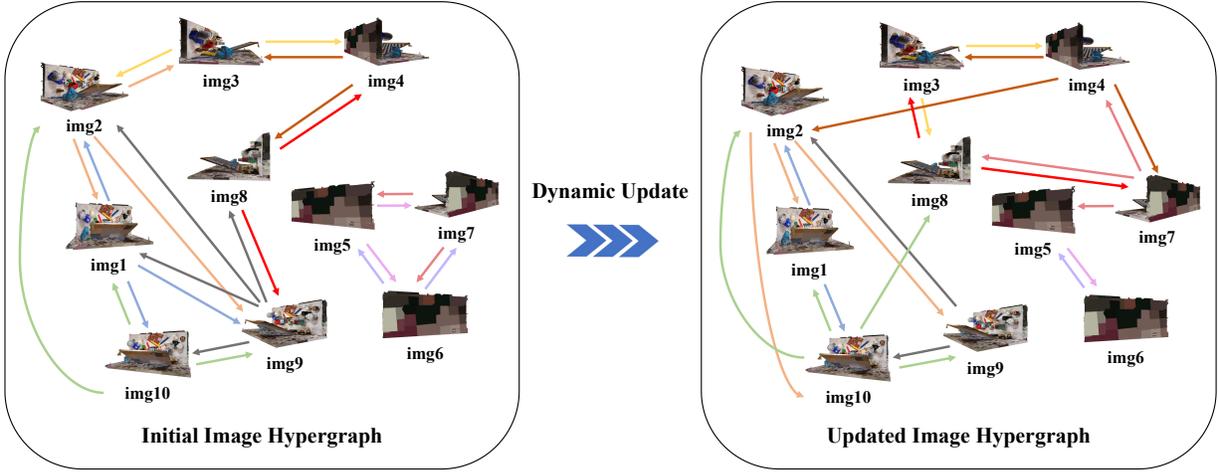


Fig. 6. Illustration of the dynamic update of image hypergraph in our proposed DHCN. The left is the initial image hypergraph, and the right is the final image hypergraph. Each image corresponds to a vertex in the hypergraph, and vertices with high-order correlation to the centroid are connected with lines of the same color. These related vertices ultimately form hyperedges.

the hyperedge $e_{spa}^i \in \mathbb{R}^{1 \times N}$ of the i^{th} vertex can be generated by repeating the following formula:

$$e_{spa}^{ij} = \begin{cases} 1, & \text{if } \mathcal{D}_A(v_i, v_j) \leq \theta \\ 0, & \text{otherwise} \end{cases}, \quad (5)$$

where $\mathcal{D}_A(\cdot)$ computes the angular distance between two vertices, v_i and v_j represents the feature representation corresponding to the i^{th} and j^{th} vertices, θ represents a predefined threshold, which is set to 36° in the experiments. Thus, the spatial-based hyperedges \mathcal{E}_{spa} can be expressed as:

$$\mathcal{E}_{spa} = \{e_{spa}^1, e_{spa}^2, \dots, e_{spa}^N\} \in \mathbb{R}^{N \times N}, \quad (6)$$

2) *Distance-Based Method*: The distance-based method constructs a hyperedge by finding the nearest k neighbor vertices in the feature space according to the feature representation of each vertex. This method mainly uses KNN to calculate the distance d_{ij} ($i \in \{1, 2, \dots, N\}, j \in \{1, 2, \dots, N\}$) between two vertices. In order to measure the degree of correlation between vertices more effectively, we normalize the distance as:

$$\bar{d}_{ij} = \exp\left(-\frac{d(v_i, v_j)^2}{\sigma^2}\right), \quad (7)$$

where σ represents the median distance of all pairs of vertices, the distance will be normalized to $[0, 1]$. Therefore, the hyperedge $e_{dis}^i \in \mathbb{R}^{1 \times N}$ of the i^{th} vertex can be generated by the following equation:

$$e_{dis}^{ij} = \begin{cases} \bar{d}_{ij}, & \text{if } \mathcal{D}_R(d_{ij}) \leq k \\ 0, & \text{otherwise} \end{cases}, \quad (8)$$

where $\mathcal{D}_R(\cdot)$ represents the rank of the distance among all distances. If the distance ranking between the point and the centroid is in the top k , then the point is selected as a neighbor, otherwise the distance is set to 0. Finally, the distance-based hyperedges \mathcal{E}_{dis} can be expressed as:

$$\mathcal{E}_{dis} = \{e_{dis}^1, e_{dis}^2, \dots, e_{dis}^N\} \in \mathbb{R}^{N \times N}, \quad (9)$$

3) *Representation-Based Method*: Compared with the more direct distance-based methods, representation-based methods capture the correlation between vertices by reconstructing their features. Specifically, we choose a vertex as a centroid and use its neighboring vertices for sparse representation, which effectively captures the relationship between vertices. The sparse representation is computed as follows:

$$\underset{c}{\operatorname{argmin}} \|NV(k, c) - v_i\|_2^2 + \gamma \|c\|_1, \quad (10)$$

where NV represents all neighbor vertices, k represents the available number of neighbors in the sparse representation, and $c = \{c_1, c_2, \dots, c_k\}$ ($s.t. \forall i, c_i \geq 0$) represents the learned reconstruction coefficients of the available neighbors, γ represents a fixed adjustment parameter. The first term in the equation is the reconstruction term, which is used to provide a good reconstruction of the centroid by the neighbor vertices. The second term is a penalty term, which is used to constrain the representation of the first term to be sparse. After obtaining the sparse representation, the hyperedge $e_{rep}^i \in \mathbb{R}^{1 \times N}$ of the i^{th} vertex can be generated by the following equation:

$$e_{rep}^{ij} = \begin{cases} c_j, & \text{if } v_j \in NV(k, c) \\ 0, & \text{otherwise} \end{cases}, \quad (11)$$

where c_j denotes the corresponding reconstruction coefficient when the current vertex is used for sparse representation, at which point they are considered correlated. Similarly, the representation-based hyperedges \mathcal{E}_{rep} can be expressed as:

$$\mathcal{E}_{rep} = \{e_{rep}^1, e_{rep}^2, \dots, e_{rep}^N\} \in \mathbb{R}^{N \times N}, \quad (12)$$

4) *Similarity-Based Method*: Similarity is a crucial metric for assessing the relationship between two vectors. Here, we adopt cosine similarity to calculate the correlation between feature representations of vertices as follows:

$$s_{ij} = \frac{v_i \cdot v_j}{\|v_i\|_2 \cdot \|v_j\|_2}, \quad (13)$$

Based on this, the hyperedge $e_{sim}^i \in \mathbb{R}^{1 \times N}$ of the i^{th} vertex can be generated by the following equation:

$$e_{sim}^{ij} = \begin{cases} 1, & \text{if } \mathcal{S}_R(s_{ij}) \leq k \\ 0, & \text{otherwise} \end{cases}, \quad (14)$$

where $\mathcal{S}_R(\cdot)$ is similar to the previous and represents the overall ranking of similarity. Therefore, the similarity-based hyperedges \mathcal{E}_{sim} can be expressed as:

$$\mathcal{E}_{sim} = \{e_{sim}^1, e_{sim}^2, \dots, e_{sim}^N\} \in \mathbb{R}^{N \times N}, \quad (15)$$

5) *Dynamic Hypergraph Generation*: Due to the extensibility of hypergraphs, we can easily generate hypergraphs using multiple hyperedges. The hyperedge of the projected image \mathcal{E}_{img} and the point group \mathcal{E}_{gro} can be expressed as:

$$\mathcal{E}_{img} = \{\mathcal{E}_{spa}, \mathcal{E}_{dis}, \mathcal{E}_{rep}, \mathcal{E}_{sim}\} \in \mathbb{R}^{N \times 4N}, \quad (16)$$

$$\mathcal{E}_{gro} = \{\mathcal{E}_{dis}, \mathcal{E}_{rep}, \mathcal{E}_{sim}\} \in \mathbb{R}^{M \times 3M}, \quad (17)$$

Therefore, the final generated projected image hypergraph \mathcal{G}_{img} and point group hypergraph \mathcal{G}_{gro} can be expressed as:

$$\mathcal{G}_{img} = \{\mathcal{V}_{img}, \mathcal{E}_{img}, \mathcal{W}_{img}\}, \quad (18)$$

$$\mathcal{G}_{gro} = \{\mathcal{V}_{gro}, \mathcal{E}_{gro}, \mathcal{W}_{gro}\}, \quad (19)$$

Static hypergraphs are monotone, and dynamic changes in feature representations during iterations may invalidate static hypergraphs. To address this issue, we continuously update the interaction information between hypergraph vertices using a dynamic hypergraph generator, dynamically learning the optimal structure of the hypergraph during the iteration process. In this process, the hypergraph structure is gradually perfected. The structure of dynamic hypergraph update includes feature representation of vertices \mathcal{V} , hyperedges \mathcal{E} and hyperedge weights \mathcal{W} , which makes data-dependent modeling more accurate. Taking the projection image as an example, the result of dynamic hypergraph update is shown in Fig. 6. In the figure, we only plot the hyperedges with higher weights. As can be seen from the figure, we can observe that the initial image hypergraph tends to assign larger weights to hyperedges with small angular distances between vertices. After the optimization is completed, more weights will be given to the hyperedges of two vertices that are closer, more similar and more correlated in the high-dimensional space. Therefore, the above method can generate hypergraphs with higher quality, more accurately describe the high-order correlation between vertices, and play a more critical role in the process of network optimization.

D. Quality Predictor

After the dynamic hypergraph generation, we utilize hypergraph convolution [83]–[85] to model the complex interactions between the projected image and the point groups. The quality predictor has two branches, which are composed of five hypergraph convolutional layers, and the number of output channels of each layer is [512, 256, 128, 32, 1]. The components of hypergraph convolutional layer include hypergraph convolution, batch normalization, softplus activation function,

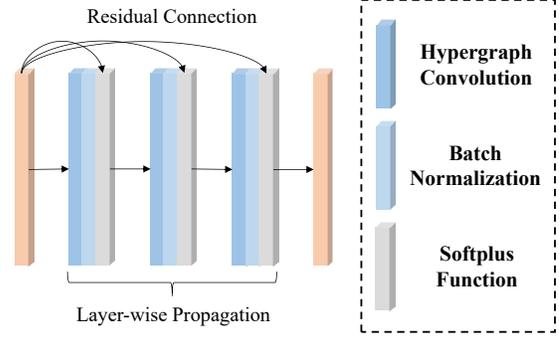


Fig. 7. Structure of the ResHGC block.

Residual Hypergraph Convolution (ResHGC) block and average pooling, as shown in Fig. 2. Taking the hyperedge \mathcal{E}_{img} of the projected image as an example, we first normalize it as follows:

$$\bar{\mathcal{E}}_{img} = \mathbf{D}_v^{-\frac{1}{2}} \mathcal{E}_{img} \mathbf{W} \mathbf{D}_e^{-1} \mathcal{E}_{img}^T \mathbf{D}_v^{-\frac{1}{2}}, \quad (20)$$

$$\mathbf{D}_v = \text{diag} \left\{ \sum_{i=0}^N \mathcal{E}_{1,i}, \dots, \sum_{i=0}^N \mathcal{E}_{N,i} \right\}, \quad (21)$$

$$\mathbf{D}_e = \text{diag} \left\{ \sum_{i=0}^{4N} \mathcal{E}_{1,i}^T, \dots, \sum_{i=0}^{4N} \mathcal{E}_{4N,i}^T \right\}, \quad (22)$$

where $\mathbf{D}_v \in \mathbb{R}^{N \times N}$ is the diagonal matrix of vertex degrees, $\mathbf{D}_e \in \mathbb{R}^{4N \times 4N}$ is the diagonal matrix of hyperedge degrees. Then, hypergraph convolutional layer continuously aggregates neighborhood information based on $\bar{\mathcal{E}}_{img} \in \mathbb{R}^{N \times N}$ to generate new feature representations, and its layer-wise propagation rules are as follows:

$$\mathbf{M}^{(l+1)} = \sigma(\text{BN}(\bar{\mathcal{E}}_{img} \mathbf{M}^{(l)} \mathbf{w}^{(l)})), \quad (23)$$

where σ is the softplus activation function, BN represents batch normalization, $\mathbf{M}^{(l)}$ is the feature matrix of the l^{th} layer, $\mathbf{w}^{(l)}$ is a trainable parameter of the l^{th} layer of the network. Inspired by [86], we also adopt the ResHGC block to deepen the network structure through the residual connection to achieve the performance improvement of the network. The specific structure of the ResHGC block is shown in Fig. 7. We use multiple residual connections to improve the stability of layer-wise propagation and the effectiveness of the network. Finally, we obtain the feature vectors of the projected image and point groups through the average pooling operation. The quality score is obtained by weighted summation of the two feature vectors.

E. Training Setup

The entire network serves as an end-to-end framework for jointly training two branches: the projection image branch and the point group branch. The trainable parameters of the network are optimized by Adam optimizer [87]. The loss function used to optimize the network l_1 is defined as:

$$l_1 = \frac{1}{n} \sum_{i=0}^n |y_i - \bar{y}_i|, \quad (24)$$

TABLE II

PERFORMANCE COMPARISON RESULTS ON SJTU-PCQA, WPC AND SIAT-PCQD DATABASES. ALL INDICATORS ADOPT ABSOLUTE VALUES FOR PERFORMANCE COMPARISON FOR BETTER VISIBILITY. THE FIRST, SECOND, AND THIRD OF THE FOUR INDICATORS ARE MARKED IN RED, BLUE AND GREEN, RESPECTIVELY.

Ref	Type	Metric	SJTU-PCQA				WPC				SIAT-PCQD			
			PLCC↑	SRCC↑	KRCC↑	RMSE↓	PLCC↑	SRCC↑	KRCC↑	RMSE↓	PLCC↑	SRCC↑	KRCC↑	RMSE↓
FR	PC-Based	$PSNR_{MSE,p2po}$ [45]	0.7622	0.6002	0.4917	1.4382	0.2673	0.1607	0.1147	20.6947	0.3372	0.1555	0.1236	0.1217
		$PSNR_{MSE,p2pl}$ [46]	0.7381	0.5505	0.4375	1.5357	0.2879	0.1182	0.0851	21.1898	0.3369	0.1541	0.1232	0.1214
		$PSNR_{HF,p2po}$ [45]	0.7737	0.6744	0.5217	1.4481	0.3555	0.0557	0.0384	20.8197	0.3510	0.2801	0.1995	0.1215
		$PSNR_{HF,p2pl}$ [46]	0.7286	0.6208	0.4701	1.6000	0.3263	0.0989	0.0681	21.1100	0.3179	0.2633	0.1837	0.1229
		AS_{Mean} [48]	0.5297	0.5317	0.3723	2.7129	0.3397	0.2484	0.1801	21.5013	0.1213	0.1303	0.0963	0.1288
		AS_{RMS} [48]	0.7156	0.5653	0.4144	1.6550	0.3347	0.2479	0.1802	21.5325	0.1455	0.1559	0.1175	0.1284
		AS_{MSE} [48]	0.5115	0.5472	0.3865	2.6431	0.3397	0.2484	0.1801	21.5013	0.1512	0.1559	0.1175	0.1283
		$PSNR_Y$ [47]	0.8124	0.7871	0.6116	1.3222	0.6166	0.5823	0.4164	17.9001	0.5054	0.4071	0.2967	0.1114
		PCQM [51]	0.8301	0.7748	0.6152	1.2978	0.6162	0.5504	0.4409	17.9027	0.8176	0.7005	0.5405	0.0721
	PointSSIM [53]	0.7422	0.7051	0.5321	1.5601	0.5225	0.4639	0.3394	19.3863	0.5514	0.4405	0.3326	0.1029	
	GraphSIM [56]	0.9158	0.8853	0.7063	0.9462	0.6833	0.6217	0.4562	16.5107	0.7857	0.6724	0.5134	0.0717	
	Projection-Based	SSIM [62]	0.8868	0.8667	0.6988	1.0454	0.6690	0.6483	0.4685	16.8841	0.4399	0.3904	0.2764	0.1161
		MS-SSIM [63]	0.8930	0.8738	0.7069	1.0091	0.7349	0.7179	0.5385	15.3341	0.4357	0.4491	0.3217	0.1150
		IW-SSIM [64]	0.8932	0.8638	0.6934	1.0268	0.7688	0.7608	0.5707	14.5453	0.5343	0.4853	0.3454	0.1088
VIFP [65]		0.8977	0.8624	0.6934	1.0173	0.7508	0.7426	0.5575	15.0328	0.4692	0.4327	0.3082	0.1140	
RR	PC-Based	PCMRR [59]	0.6699	0.5622	0.4091	1.7589	0.3926	0.3605	0.2543	20.9203	0.7172	0.6945	0.5294	0.0874
	Projection-Based	RR-CAP [67]	0.7691	0.7577	0.5508	1.5512	0.7307	0.7162	0.5260	15.6485	\	\	\	\
NR	PC-Based	3D-NSS [17]	0.7813	0.7819	0.6023	1.7740	0.6284	0.6309	0.4573	18.1706	0.4022	0.3568	0.2482	0.1186
		Wang et al. [24]	0.9400	0.9300	0.7900	\	0.8100	0.8000	0.6200	\	\	\	\	\
		ResSCNN [22]	0.8865	0.8328	0.6514	1.0728	0.4531	0.4362	0.2987	20.2591	0.3826	0.3013	0.2092	0.1198
	Projection-Based	PQANet [27]	0.7998	0.7593	0.5796	1.3773	0.6671	0.6368	0.4684	16.6758	0.4779	0.4426	0.3048	0.1140
		IT-PCQA [29]	0.8605	0.8286	0.6453	1.1686	0.4870	0.4329	0.3006	19.8960	0.5062	0.2845	0.2166	0.1088
		GMS-3DQA [31]	0.9177	0.9108	0.7735	0.7872	0.8338	0.8308	0.6457	12.2292	0.7382	0.6518	0.4881	0.0843
	Hybrid-Based	MM-PCQA [32]	0.9226	0.9103	0.7838	0.7716	0.8556	0.8414	0.6513	12.3506	0.6392	0.5561	0.3807	0.0988
Ours	0.9574	0.9421	0.8124	0.6805	0.8659	0.8616	0.6794	11.3910	0.8364	0.7847	0.6049	0.0679		

where n is the batch size during training, y_i represents the ground truth score, \hat{y}_i represents the predicted score. During training, the training epoch of the network is set to 100, the batch size is set to 16, and the initial learning rate is $1e-3$, which is scaled to 0.5 times of the original every 10 epochs. To avoid overfitting, the training is early stopped when the network parameters are not updated for 20 consecutive epochs.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we first describe the databases used in our experiments, the performance metrics, and the implementation details. Next, we conduct performance comparison and ablation studies to evaluate the effectiveness of the proposed DHCN method. Finally, we analyze the experimental results and draw the conclusions.

A. Databases

To evaluate the effectiveness of the proposed method, we conduct experiments on three publicly available databases, including SJTU-PCQA [88], WPC [89], and SIAT-PCQD [90].

The SJTU-PCQA database contains 9 reference point clouds. Based on this, a total of 42 distorted point clouds are generated for each reference point cloud by applying 7 distortion types (i.e., OT, CN, DS, D+C, D+G, GGN, C+G) with 6 distortion levels. In the end, 378 distorted point clouds are generated.

The WPC database contains 20 reference point clouds. Each reference point cloud in the database generates 37 distorted

point clouds using 5 distortion types (i.e., downsample, gaussian noise, G-PCC(T), V-PCC, G-PCC(O)), resulting in a total of 740 distorted point clouds.

The unique aspect of the SIAT Point Cloud Quality Database (SIAT-PCQD), unlike other databases, lies in its emphasis on exploring the impact of geometric and texture properties on compression distortion. This database encompasses 340 distorted point clouds, generated from 20 distinct reference point clouds by utilizing 17 varying geometric and texture quantization parameters (QP).

B. Performance Criteria

We use four criteria for performance evaluation of different quality metrics, including Spearman Rank Correlation Coefficient (SRCC), Pearson Linear Correlation Coefficient (PLCC), Kendall Rank Correlation Coefficient (KRCC), and Root Mean Squared Error (RMSE). Higher values of SRCC, PLCC, KRCC and lower value of RMSE indicate better performance of PCQA metrics. To address the scale discrepancy between the predicted quality scores and the subjective scores, we perform a non-linear mapping on the predicted scores by using a five-parameter logistic function [91] before calculating the PLCC and RMSE:

$$\hat{y} = \beta_1 \left(\frac{1}{2} - \frac{1}{1 + \exp(\beta_2(\bar{y} - \beta_3))} \right) + \beta_4 \bar{y} + \beta_5, \quad (25)$$

where \bar{y} is the predicted score, \hat{y} indicates the predicted score after mapping, β_i , $i \in \{1, 2, \dots, 5\}$ are the parameters to be fitted.

TABLE III

SRCC PERFORMANCE EVALUATION OF EXISTING PCQA METRICS BASED ON POINT CLOUD CONTENT AND DISTORTION TYPE IS PERFORMED ON THE SJTU-PCQA DATABASE. ABSOLUTE SRCC IS USED FOR COMPARISON TO OBTAIN BETTER VISIBILITY. THE LETTERS A-R IN THE TABLE STAND FOR $PSNR_{RMSE,p2po}$, $PSNR_{RMSE,p2pl}$, $PSNR_{HF,p2po}$, $PSNR_{HF,p2pl}$, AS_{RMS} , $PSNR_Y$, PCQM, POINTSSIM, GRAPHSIM, SSIM, MS-SSIM, IW-SSIM, VIFP, PCMR, 3D-NSS, ResSNCNN, PQANet, IT-PCQA AND OUR PROPOSED METHOD IN TURN. THE FIRST, SECOND, AND THIRD PLACES IN THE SPCC INDICATOR ARE MARKED IN RED, BLUE AND GREEN, RESPECTIVELY.

Subset		FR													RR	NR				
		A [45]	B [46]	C [45]	D [46]	E [48]	F [47]	G [51]	H [53]	I [56]	J [62]	K [63]	L [64]	M [65]	N [59]	O [17]	P [22]	Q [27]	R [29]	Ours
Content	Redandblack	0.6196	0.5943	0.7421	0.6819	0.5799	0.7478	0.8024	0.6670	0.8702	0.8603	0.8718	0.8911	0.8885	0.6506	0.8647	0.8003	0.8603	0.8557	0.9527
	Romanoillamp	0.4247	0.3617	0.7457	0.6032	0.6022	0.4278	0.5145	0.5150	0.8525	0.7509	0.7869	0.7939	0.7882	0.6044	0.6885	0.6193	0.7509	0.7248	0.9686
	Loot	0.6738	0.6405	0.7447	0.6391	0.4817	0.7875	0.8426	0.7299	0.8868	0.8693	0.8809	0.8846	0.8619	0.6770	0.8890	0.8780	0.8693	0.8778	0.9626
	Soldier	0.6781	0.6478	0.7493	0.6329	0.5404	0.8336	0.8684	0.7718	0.9118	0.8917	0.8843	0.8843	0.8744	0.5809	0.8731	0.9123	0.8917	0.8050	0.9673
	ULB Unicorn	0.7085	0.6082	0.8500	0.8081	0.4773	0.8687	0.7496	0.5715	0.8597	0.9084	0.8981	0.8548	0.8514	0.5148	0.4101	0.8364	0.9084	0.9129	0.8832
	Longdress	0.6640	0.6437	0.7885	0.7096	0.5704	0.9326	0.8896	0.8608	0.9499	0.9245	0.9191	0.8710	0.8976	0.6474	0.9005	0.8650	0.9245	0.8243	0.9689
	Statue	0.5678	0.5362	0.5883	0.5652	0.6291	0.8241	0.7483	0.7391	0.8744	0.8578	0.8663	0.8428	0.8637	0.4181	0.8520	0.9002	0.8578	0.8757	0.9272
	Shiva	0.4129	0.4074	0.1168	0.2689	0.7057	0.8375	0.8060	0.7896	0.8595	0.8968	0.8914	0.8744	0.8903	0.4884	0.8198	0.8599	0.8968	0.8243	0.9182
	Hhi	0.6526	0.5150	0.7443	0.6785	0.5012	0.8242	0.7524	0.7010	0.9028	0.8409	0.8658	0.8773	0.8462	0.4785	0.7394	0.8240	0.8409	0.7577	0.9306
	Distortion	OT	0.4407	0.4407	0.3788	0.3524	0.5210	0.3068	0.6495	0.7108	0.7049	0.2198	0.2712	0.3382	0.3743	0.1800	0.4068	0.1683	0.0883	0.0189
CN		NaN	NaN	NaN	NaN	NaN	0.5588	0.6070	0.7660	0.7779	0.6283	0.6453	0.7531	0.7429	0.7157	0.1480	0.2265	0.5507	0.0655	0.9075
DS		0.4495	0.4489	0.6847	0.3286	0.3653	0.4697	0.6990	0.8500	0.8654	0.3246	0.4718	0.4535	0.4546	0.1489	0.5051	0.4292	0.2958	0.0556	0.9395
D+C		0.5735	0.5979	0.7619	0.7499	0.4025	0.7397	0.8014	0.7449	0.8846	0.5062	0.6281	0.6661	0.6932	0.6120	0.5895	0.5158	0.4899	0.0468	0.9648
D+G		0.6779	0.7058	0.7423	0.7196	0.8915	0.5413	0.7476	0.9288	0.8833	0.6920	0.7589	0.8222	0.7989	0.7439	0.7442	0.5263	0.5033	0.0411	0.9723
GGN		0.7008	0.7144	0.7453	0.7328	0.9376	0.5727	0.7143	0.9027	0.9064	0.7436	0.7783	0.8324	0.8436	0.7813	0.8435	0.4497	0.3771	0.0798	0.9624
C+G		0.7577	0.7758	0.8205	0.8025	0.9241	0.6692	0.7078	0.7991	0.9334	0.7307	0.7948	0.8406	0.8463	0.8329	0.8645	0.5523	0.6137	0.1044	0.9582

TABLE IV

SRCC PERFORMANCE EVALUATION OF EXISTING PCQA METRICS BASED ON POINT CLOUD CONTENT AND DISTORTION TYPE IS PERFORMED ON THE WPC DATABASE. ABSOLUTE SRCC IS USED FOR COMPARISON TO OBTAIN BETTER VISIBILITY. THE LETTERS REPRESENT THE SAME PCQA METRICS AS THE TABLE ABOVE. THE FIRST, SECOND, AND THIRD PLACES IN THE SPCC INDICATOR ARE MARKED IN RED, BLUE AND GREEN, RESPECTIVELY.

Subset		FR													RR	NR				
		A [45]	B [46]	C [45]	D [46]	E [48]	F [47]	G [51]	H [53]	I [56]	J [62]	K [63]	L [64]	M [65]	N [59]	O [17]	P [22]	Q [27]	R [29]	Ours
Content	Bag	0.6669	0.5751	0.4363	0.4365	0.4325	0.8051	0.5955	0.4829	0.7164	0.7300	0.7584	0.7309	0.7093	0.6069	0.7731	0.1603	0.3504	0.6174	0.8326
	Banana	0.6471	0.5691	0.1933	0.2033	0.3147	0.6211	0.4649	0.2202	0.5045	0.8011	0.7677	0.7790	0.7771	0.5287	0.6524	0.2475	0.6949	0.2485	0.8224
	Biscuits	0.5252	0.4160	0.3085	0.3368	0.3505	0.7764	0.6245	0.5816	0.7198	0.9173	0.9500	0.7992	0.7416	0.4310	0.6645	0.4765	0.6147	0.3570	0.8025
	Cake	0.3074	0.1798	0.1724	0.1796	0.0609	0.5180	0.4566	0.3177	0.4251	0.7390	0.7691	0.6534	0.6477	0.3070	0.4547	0.4467	0.5835	0.7300	0.9004
	Cauliflower	0.3501	0.2058	0.0918	0.1653	0.1781	0.5927	0.4903	0.4237	0.5529	0.8004	0.8608	0.8182	0.7008	0.4187	0.5517	0.5095	0.6238	0.0593	0.8848
	Flowerpot	0.6509	0.5298	0.4348	0.4515	0.3629	0.6385	0.5875	0.3784	0.6609	0.8303	0.9066	0.9047	0.8954	0.0477	0.6958	0.4900	0.2357	0.8127	0.8006
	GlassesCase	0.5845	0.4390	0.2020	0.3238	0.4288	0.7826	0.5861	0.5258	0.6546	0.7617	0.7577	0.7304	0.7459	0.3883	0.4790	0.2003	0.7674	0.7750	0.8954
	HoneydewMelon	0.4890	0.3299	0.2768	0.2300	0.3228	0.6740	0.4500	0.5609	0.7248	0.8549	0.8917	0.9180	0.8279	0.5742	0.7229	0.4026	0.7418	0.7352	0.8687
	House	0.5866	0.4483	0.3429	0.3434	0.4522	0.7798	0.5880	0.5590	0.7373	0.7788	0.7793	0.7357	0.7200	0.4905	0.7646	0.4780	0.8668	0.4201	0.9201
	Litchi	0.5109	0.4291	0.3478	0.3204	0.3554	0.7027	0.5965	0.6422	0.6958	0.7748	0.8623	0.7496	0.7018	0.4839	0.8113	0.1994	0.7207	0.0868	0.8803
	Mushroom	0.6396	0.5156	0.3486	0.3105	0.2911	0.6550	0.5725	0.5443	0.6802	0.7821	0.8781	0.8160	0.7897	0.2556	0.8153	0.0754	0.5835	0.3570	0.9094
	PenContainer	0.7720	0.6688	0.2159	0.3635	0.5465	0.7328	0.6394	0.5948	0.8250	0.8954	0.8758	0.8485	0.8397	0.6830	0.7809	0.5676	0.6470	0.7859	0.8819
	Pineapple	0.3777	0.2785	0.1376	0.1831	0.2155	0.7217	0.6427	0.5386	0.6401	0.7307	0.7805	0.5856	0.6441	0.4011	0.6074	0.5275	0.6318	0.5913	0.9064
	PingpongBat	0.5924	0.4984	0.4958	0.4357	0.4521	0.5428	0.5783	0.6051	0.7697	0.8054	0.8812	0.7570	0.7539	0.5092	0.6935	0.3518	0.6358	0.4737	0.8452
	PuerTea	0.6069	0.4746	0.1173	0.0384	0.4734	0.7639	0.5685	0.4139	0.7999	0.8917	0.8668	0.8359	0.7866	0.4308	0.4763	0.1456	0.7359	0.5467	0.7366
	Pumpkin	0.4947	0.3423	0.3092	0.3068	0.3220	0.6901	0.5934	0.5699	0.6517	0.9111	0.8156	0.9042	0.8976	0.3241	0.5768	0.4052	0.7857	0.5536	0.8919
	Ship	0.7464	0.6267	0.3404	0.5158	0.4943	0.7786	0.5434	0.4488	0.7558	0.8973	0.8578	0.8340	0.8013	0.4400	0.6935	0.6612	0.5349	0.3777	0.8497
	Statue	0.8040	0.6707	0.2450	0.4487	0.4900	0.7001	0.5714	0.5085	0.7390	0.8985	0.9372	0.9099	0.8950	0.1811	0.6368	0.5782	0.3762	0.4976	0.9018
	Stone	0.6219	0.5129	0.3551	0.3424	0.3649	0.7115	0.6475	0.6126	0.1920	0.8426	0.8881	0.8587	0.8196	0.3632	0.6968	0.2122	0.8234	0.1790	0.8867
	ToolBox	0.3937	0.2969	0.1972	0.1884	0.2984	0.8706	0.6304	0.4927	0.7935	0.7821	0.8255	0.8056	0.7411	0.5239	0.5806	0.5026	0.8653	0.4694	0.9163
Distortion	Downsampling	0.4815	0.3251	0.5356	0.4879	0.2465	0.5542	0.4537	0.8319	0.7903	0.8234	0.8834	0.8822	0.8828	0.7407	0.7508	0.2899	0.7234	0.3327	0.9014
	Gaussian noise	0.6155	0.6194	0.6149	0.6150	0.6844	0.7644	0.8775	0.5844	0.7469	0.6264	0.7118	0.8560	0.8847	0.7762	0.7460	0.5459	0.7938	0.1718	0.8770
	G-PCC (T)	0.3451	0.3568	0.2811	0.3085	0.1342	0.5916	0.7775	0.6745	0.7457	0.4669	0.6042	0.6742	0.6304	0.2702	0.5947	0.2531	0.4710	0.1987	0.8298
	V-PCC	0.1602	0.1992	0.2051	0.2370	0.3877	0.3203	0.5534	0.3546	0.5989	0.5141	0.5812	0.7063	0.7410	0.2966	0.3927	0.1028	0.0045	0.0090	0.7717
G-PCC (O)	NaN	NaN	NaN	NaN	0.0350	0.8072	0.8944	0.7917	0.8258	0.5290	0.7214	0.7128	0.7116	0.6468	0.2891	0.0247	0.4204	0.1180	0.9196	

C. Implementation Details

In the experiments, we use Pytorch1.13.1 deep learning framework. Specifically, we set the number of projected images $M=10$ and the number of point groups $N=6$. To ensure the reliability of the experimental results, we follow the k -fold cross-validation strategy during the training and testing process. The k -fold cross-validation is a dynamic validation method that can reduce the impact of data splitting. Firstly, all the samples are divided into k equal size subsets. Then, we

iterate over each subset in turn, each time using the current subset as the test set and all other subsets as the train set. The final performance result is the average results over all test sets. For the SJTU-PCQA, WPC, and SIAT-PCQD databases, the number of subsets is set to 9, 5, 5, respectively.

D. Performance Comparison

The performance comparison results on the SJTU-PCQA, WPC and SIAT-PCQD databases are shown in Table II. In

TABLE V
 SRCC PERFORMANCE EVALUATION OF EXISTING PCQA METRICS BASED ON POINT CLOUD CONTENT IS PERFORMED ON THE SIAT-PCQD DATABASE. ABSOLUTE SRCC IS USED FOR COMPARISON TO OBTAIN BETTER VISIBILITY. THE LETTERS REPRESENT THE SAME PCQA METRICS AS THE TABLE ABOVE. THE FIRST, SECOND, AND THIRD PLACES IN THE SPCC INDICATOR ARE MARKED IN RED, BLUE AND GREEN, RESPECTIVELY.

Subset	FR														RR	NR				
	A [45]	B [46]	C [45]	D [46]	E [48]	F [47]	G [51]	H [53]	I [56]	J [62]	K [63]	L [64]	M [65]	N [59]	O [17]	P [22]	Q [27]	R [29]	Ours	
Content	Andrew	0.3354	0.3354	0.4922	0.4922	0.3791	0.6151	0.5979	0.5979	0.5882	0.6421	0.6519	0.6274	0.5390	0.6039	0.3990	0.4923	0.4926	0.7256	0.7525
	AngelSeated	0.1490	0.2066	0.2566	0.2923	0.2747	0.8529	0.7010	0.7794	0.0282	0.5172	0.5539	0.5123	0.4069	0.5672	0.4926	0.2966	0.2010	0.8382	0.9461
	Banana	0.2241	0.2241	0.5770	0.5770	0.7774	0.8284	0.8358	0.8186	0.7157	0.9044	0.9461	0.8799	0.8505	0.8294	0.0858	0.3480	0.2721	0.7525	0.5392
	Biplane	0.0939	0.0939	0.4770	0.3593	0.5195	0.9828	0.9779	0.9951	0.9877	0.6176	0.7157	0.5833	0.5907	0.6667	0.7819	0.0980	0.3946	0.8725	0.8946
	Bush	0.1740	0.1740	0.3017	0.2641	0.3017	0.8578	0.8824	0.8971	0.8995	0.6544	0.6789	0.5172	0.6250	0.8652	0.4632	0.02157	0.1324	0.8676	0.9069
	Facade	0.0926	0.0926	0.0302	0.0900	0.2203	0.9779	0.9583	0.9779	0.9657	0.9265	0.8971	0.5980	0.5613	0.9187	0.3137	0.0049	0.6373	0.5980	0.9559
	Grass	0.1627	0.1627	0.4457	0.1627	0.5884	0.9240	0.9363	0.9510	0.9289	0.8676	0.8775	0.7966	0.7525	0.9632	0.8309	0.0980	0.4583	0.8652	0.7941
	House	0.0476	0.0476	0.0832	0.0976	0.0776	0.8407	0.8725	0.9093	0.9289	0.5294	0.5417	0.4069	0.4314	0.3985	0.3554	0.0784	0.4069	0.8162	0.9534
	Longdress	0.2228	0.2228	0.4331	0.3956	0.4782	0.8554	0.9093	0.8725	0.8382	0.9093	0.8701	0.7377	0.7647	0.9416	0.6397	0.1324	0.3407	0.8676	0.9044
	Loot	0.2103	0.2103	0.2854	0.2103	0.2729	0.9657	0.9191	0.9651	0.9412	0.8652	0.8456	0.7843	0.6299	0.9576	0.4314	0.0294	0.5074	0.6887	0.9093
	Nike	0.3593	0.3593	0.4196	0.3393	0.3167	0.9069	0.8725	0.8995	0.9020	0.6495	0.5956	0.5735	0.4877	0.8375	0.5025	0.2770	0.5074	0.1275	0.8995
	Phil	0.1645	0.1645	0.4215	0.3753	0.4704	0.7317	0.7833	0.6998	0.7108	0.7980	0.7980	0.7735	0.8201	0.8219	0.5427	0.3339	0.2353	0.7023	0.7377
	RedAndBlack	0.1365	0.1365	0.4644	0.3718	0.4770	0.8260	0.8848	0.8235	0.8015	0.8946	0.8848	0.7990	0.7794	0.9142	0.5221	0.3799	0.4657	0.4804	0.8799
	Ricardo	0.1645	0.1645	0.3830	0.3701	0.3830	0.5783	0.5684	0.6176	0.5466	0.7084	0.6839	0.7452	0.7354	0.7551	0.5242	0.2149	0.2525	0.7256	0.6275
	RomanOillamp	0.2053	0.2053	0.3705	0.3831	0.3330	0.9657	0.9608	0.9730	0.9485	0.7426	0.7819	0.7721	0.6250	0.9761	0.8235	0.2059	0.4828	0.3529	0.9706
	Sarah	0.3521	0.3521	0.6529	0.6529	0.6143	0.2922	0.3438	0.1645	0.1838	0.3438	0.4273	0.05672	0.5549	0.4017	0.4813	0.4936	0.4706	0.5770	0.3113
	Soldier	0.2566	0.2566	0.3468	0.3167	0.2566	0.9608	0.8848	0.9608	0.9608	0.7696	0.7647	0.7206	0.5907	0.8817	0.5735	0.1838	0.1912	0.7010	0.9657
	The20sMaria	0.1502	0.1502	0.3055	0.3188	0.3630	0.9559	0.8946	0.9485	0.9044	0.5809	0.6005	0.5441	0.5025	0.8687	0.5956	0.0931	0.3824	0.1005	0.8897
	ULBUicorn	0.0463	0.0463	0.0613	0.0613	0.1314	0.9412	0.9436	0.9069	0.9191	0.5490	0.5662	0.3627	0.6740	0.8302	0.2402	0.0294	0.3554	0.4779	0.8039
	UlliWegner	0.2216	0.2216	0.3668	0.4018	0.3918	0.9093	0.8603	0.8676	0.8603	0.8382	0.8701	0.7377	0.7034	0.9118	0.3946	0.0907	0.5637	0.8456	0.8873

the experiments, the performance of our proposed DHCN is compared with 24 state-of-the-art PCQA methods, including $PSNR_{MSE,p2po}$ [45], $PSNR_{MSE,p2pl}$ [46], $PSNR_{HF,p2po}$ [45], $PSNR_{HF,p2pl}$ [46], AS_{Mean} [48], AS_{RMS} [48], AS_{MSE} [48], $PSNR_Y$ [47], PCQM [51], PointSSIM [53], GraphSIM [56], SSIM [62], MS-SSIM [63], IW-SSIM [64], VIFP [65], PCMRR [59], RR-CAP [67], 3D-NSS [17], Wang et al. [24], ResSCNN [22], PQANet [27], IT-PCQA [29], GMS-3DQA [31], and MM-PCQA [32]. The methods are classified based on data type and the utilization of reference data. The experimental results presented in Table II lead us to draw the following conclusions:

- Compared with the existing methods, our proposed DHCN achieves the best performance on all three benchmark databases. Notably, the SRCC of DHCN is 0.0121 higher than that of the second place Wang et al. on SJTU-PCQA database, 0.0202 higher than that of the second place MM-PCQA on WPC database, and 0.0842 higher than that of the second place PCQM on SIAT-PCQD database.
- Unlike the simple SJTU-PCQA database, the WPC database involves more complex distortions, and the SIAT-PCQD database has more quantization parameters for the compression distortion. Hence, the exceptional performance of the existing PCQA methods on the SJTU-PCQA database does not necessarily carry over on the other two databases. However, the performance of our proposed DHCN shows excellent performance on all the three databases and does not show serious performance degradation, which proves its effectiveness and robustness.
- Both PC-based and projection-based methods can achieve good performance, while our proposed DHCN with hy-

brid inputs combines the advantages of both methods to achieve superior performance. The experimental results show that multi-modal learning is able to learn more effective feature representations by using feature information of data from different modalities.

In addition, to further verify the effectiveness of the PCQA method, we classify the point clouds in each database by content and distortion type, and then conduct performance tests. The detailed experimental results are shown in Table III, Table IV and Table V. From these tables, we can find that: (1) On SJTU-PCQA database and WPC database, our proposed DHCN achieves more top-3 performances, and the number of best performances is the most among all methods. (2) On the SIAT-PCQD database, due to the small number of point clouds in a single subset and the interference of compression distortion of different parameters, the reference-based method achieves better results than the no-reference method. However, our proposed DHCN also achieves good results as a no-reference method and ranks the second place in terms of the number of best performances (only less than PCMRR). (3) On the subset experiments of the first two databases, the projection-based method performs better, while on the last database, the PC-based method performs better. These results further underscore the advantages of both PC-based and projection-based methods and can be used in suitable scenarios. Therefore, our proposed hybrid-based DHCN integrates the advantages of both to improve the stability and robustness.

E. Cross-database Evaluation

In order to further validate the generalization capability of the proposed DHCN, we conducted cross-database evaluation experiments. The experimental results are shown in Table VI.

TABLE VI

EXPERIMENTAL RESULTS FOR CROSS-DATABASE EVALUATION. WPC \rightarrow SJTU-PCQA REPRESENTS TRAINING ON THE WPC DATABASE AND THEN TESTING ON THE SJTU-PCQA DATABASE. CONVERSELY, SJTU-PCQA \rightarrow WPC DENOTES TRAINING ON THE SJTU-PCQA DATABASE AND THEN TESTING ON THE WPC DATABASE. THE BEST PERFORMANCE IS INDICATED IN **BOLD**.

Method	WPC \rightarrow SJTU-PCQA		SJTU-PCQA \rightarrow WPC	
	PLCC \uparrow	SRCC \uparrow	PLCC \uparrow	SRCC \uparrow
3D-NSS	0.1248	0.1824	0.0731	0.1352
ResSCNN	0.4089	0.4031	0.2771	0.2329
PQANet	0.3245	0.3141	0.3007	0.1177
IT-PCQA	0.3185	0.3105	0.1966	0.1949
Ours	0.7955	0.7295	0.4586	0.4027

TABLE VII

PERFORMANCE RESULTS OF POINT GROUPS, PROJECTED IMAGES, DHCN ON THE WPC DATABASE. THE BEST PERFORMANCE IS INDICATED IN **BOLD**.

Model	WPC			
	PLCC \uparrow	SRCC \uparrow	KRCC \uparrow	RMSE \downarrow
FC with groups	0.5524	0.5006	0.3484	19.0168
FC with images	0.7819	0.7756	0.5917	14.1605
FC with groups and images	0.8271	0.8219	0.6367	12.7408
DHCN with groups and images	0.8659	0.8616	0.6794	11.3910

TABLE VIII

PERFORMANCE RESULTS OF DIFFERENT HYPERGRAPH GENERATION METHODS ON THE WPC DATABASE. THE BEST PERFORMANCE IS INDICATED IN **BOLD**.

Method	WPC			
	PLCC \uparrow	SRCC \uparrow	KRCC \uparrow	RMSE \downarrow
\mathcal{E}_{dis}	0.8433	0.8364	0.6509	12.2256
$\mathcal{E}_{dis} + \mathcal{E}_{rep}$	0.8511	0.8383	0.6509	11.9579
$\mathcal{E}_{dis} + \mathcal{E}_{rep} + \mathcal{E}_{sim}$	0.8659	0.8616	0.6794	11.3910

Here, our proposed method is contrasted with other NR-PCQA metrics. From the table, it can be observed that our proposed DHCN demonstrates stronger model generalization compared to other metrics.

F. Ablation Study

To prove the effectiveness of both hypergraph, hypergraph generation and the number of best neighbor vertices, we conduct ablation studies as follows:

1) *Contributions of point groups, projected images, and hypergraph*: To demonstrate the contribution of point groups, projected images, and hypergraph on the network performance, we conduct corresponding ablation studies. The final experimental results are shown in Table VII. From the experimental results, we can find that the combination of point groups and projection images as inputs can achieve better results than single input type, and the introduction of hypergraph also brings performance improvement. This demonstrates that point clouds, projection images, and DHCN can make a significant contribution to the performance improvement of the network.

2) *Comparison of different hypergraph generation methods*: The extensibility of hypergraphs opens up the possibility of combining diverse hyperedge methods. In order to verify the

TABLE IX

PERFORMANCE RESULTS OF THE NUMBER k OF NEIGHBOR VERTICES ON THE WPC DATABASE. THE BEST PERFORMANCE IS INDICATED IN **BOLD**.

k in img	k in gro	WPC			
		PLCC \uparrow	SRCC \uparrow	KRCC \uparrow	RMSE \downarrow
2	1	0.8537	0.8531	0.6678	11.8423
	2	0.8465	0.8464	0.6622	12.1149
	4	0.8491	0.8473	0.6660	12.0051
4	1	0.8659	0.8616	0.6794	11.3910
	2	0.8332	0.8313	0.6429	12.6408
	4	0.8362	0.8334	0.6464	12.4219
6	1	0.8480	0.8488	0.6642	12.0913
	2	0.8419	0.8369	0.6536	12.2407
	4	0.8356	0.8331	0.6489	12.4245
8	1	0.8405	0.8378	0.6517	12.3164
	2	0.8401	0.8380	0.6511	12.3196
	4	0.8355	0.8310	0.6435	12.5069

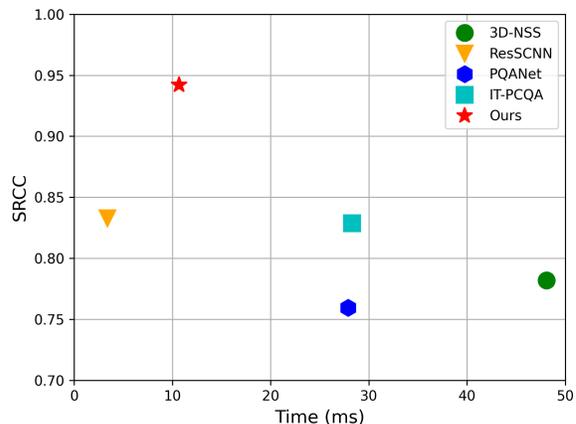


Fig. 8. Results of average computational time versus SRCC on SJTU-PCQA.

effectiveness of the proposed hypergraph generation method, we conduct experiments to compare the performance across different methods for hypergraph generation. Since the space-based method is only used for projected images, we default that it has been used. The final experimental results are shown in Table VIII. From this, we can realize that the introduction of different hyperedge generation methods can bring certain performance improvement. When three hyperedge generation methods are introduced at the same time, the performance improvement is the largest, which is 0.0252 and 0.0233 higher than that of using one or two methods.

3) *Selection of the best number of neighbor vertices*: For a hypergraph, if the number of nodes and edges is not fixed, it is impossible to directly input the hypergraph into a neural network. Therefore, it is well known that the number of neighbor vertices k is an important parameter of hypergraph. In order to select a more reasonable number of neighbor vertices k , we conduct performance comparison experiments on the WPC database. Table IX shows the performance comparison results under different configurations. In the table, the first and second columns represent the setting of the number of neighbor vertices in the hypergraph of the projected image and the hypergraph of the point group, respectively. The experimental

TABLE X
COMPARISON RESULTS OF POINT CLOUD DENOISING. CD AND P2M
MULTIPLIED BY 10^4 . THE BEST RESULTS ARE MARKED IN **BOLD**.

Method	10K points				50K points			
	1% noise		2% noise		1% noise		2% noise	
	CD↓	P2M↓	CD↓	P2M↓	CD↓	P2M↓	CD↓	P2M↓
DGCNN	3.21	1.11	5.97	2.54	1.27	0.79	3.49	2.55
DGCNN+PCQA	3.20	1.11	5.77	2.41	1.14	0.71	2.96	2.11

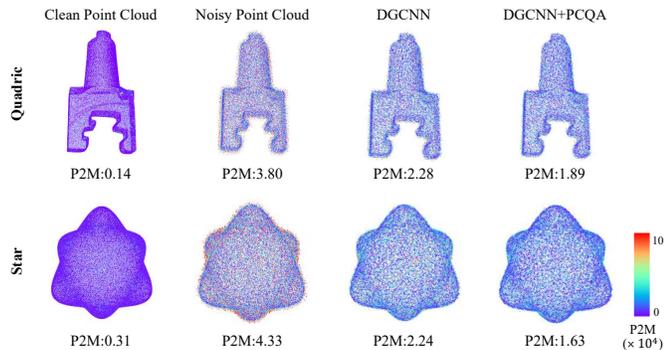


Fig. 9. Visual results on 50K resolution point clouds. The color of each point in the point cloud represents its P2M, ranging from 0 to 100.

results indicate that the introduction of the hypergraph ensures a consistently high performance. Among them, the optimal settings for the number of neighbor vertices on the projected image and point group are 4 and 1, respectively.

G. Computational Efficiency

For large-scale data such as point cloud, the computational efficiency of PCQA metric is very important. We mainly select NR-PCQA metrics for comparison, including 3D-NSS, ResSCNN, PQANet and IT-PCQA. The computational time and SRCC performance are shown in Fig. 8 where the average computational time for each point cloud is reported. Projection-based methods tend to require more computational time due to additional operations on the point cloud. ResSCNN consumes the least computational time, while 3D-NSS consumes the most. Our proposed method achieves optimal SRCC performance while ensuring suboptimal computation time, which further proves the superiority of our proposed method in balancing computational efficiency and metric performance. In addition, our method only requires 10 ms in average, which is still able to achieve real-time inference.

H. PCQA for Benchmarking PC Denoising Algorithms

PCQA metrics can be used to extract quality-aware features of point clouds and guide the optimization of PC denoising algorithms. Here, we use DGCNN [92] as the baseline model for PC denoising. Then, we additionally introduce intermediate quality-aware features to improve the DGCNN model towards a better PC denoising performance. We conduct verification experiments on 20 different PCs. The experimental results are shown in the Table X where lower values of Chamfer distance (CD) and Point2Mesh distance (P2M) indicate fewer noises and better quality. We can see that the additional quality-aware features extracted by our proposed PCQA metric are

effective to promote the performance of PC denoising. The visual results on two specific 50K-resolution point clouds are also shown in Fig. 9. The results also demonstrate the effectiveness of our newly introduced intermediate quality-aware features learned by the guidance of our PCQA metric.

V. CONCLUSION

In this paper, we propose a novel DHCN for NR-PCQA. The proposed method leverages the modeling capacity of hypergraphs to capture high-order correlations between data and constructs dynamic hypergraphs for point groups and projected images. It effectively harnesses the learning capability of multimodal data and the non-linear correlations between data, and ultimately improves the network performance. The experimental results on several public point cloud subjective quality databases demonstrate that the performance results of our proposed DHCN are significantly better than state-of-the-art PCQA methods, providing compelling evidence of its effectiveness. Furthermore, the ablation study verifies the rationality and effectiveness of the proposed network components and architecture. Additionally, while using hypergraphs can obtain performance gains, it also increases computational complexity. In future works, we will further explore more efficient hypergraph learning methods to facilitate PCQA.

REFERENCES

- [1] A. Javaheri, C. Brites, F. Pereira, and J. Ascenso, "Point cloud rendering after coding: Impacts on subjective and objective quality," *IEEE Transactions on Multimedia*, vol. 23, pp. 4049–4064, 2020.
- [2] Y. Cui, R. Chen, W. Chu, L. Chen, D. Tian, Y. Li, and D. Cao, "Deep learning for image and point cloud fusion in autonomous driving: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 722–739, 2022.
- [3] C. Wu, Y. Lin, Y. Guo, C. Wen, Y. Shi, and C. Wang, "Vehicle completion in traffic scene using 3d lidar point cloud data," in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022, pp. 7495–7498.
- [4] E. Alexiou, N. Yang, and T. Ebrahimi, "PointXR: A toolbox for visualization and subjective evaluation of point clouds in virtual reality," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2020, pp. 1–6.
- [5] A. P. Placitelli and L. Gallo, "Low-cost augmented reality systems via 3D point cloud sensors," in *2011 Seventh International Conference on Signal Image Technology & Internet-Based Systems*, 2011, pp. 188–192.
- [6] Y. Wu, X. Hu, Y. Zhang, M. Gong, W. Ma, and Q. Miao, "SACF-Net: Skip-attention based correspondence filtering network for point cloud registration," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 3585–3595, 2023.
- [7] Y. Wu, Y. Zhang, W. Ma, M. Gong, X. Fan, M. Zhang, A. K. Qin, and Q. Miao, "RORNet: Partial-to-partial registration network with reliable overlapping representations," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2023.
- [8] Y. Yuan, Y. Wu, X. Fan, M. Gong, W. Ma, and Q. Miao, "EGST: Enhanced geometric structure transformer for point cloud registration," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–13, 2023.
- [9] X. Li and P. Kuang, "3D-VRVT: 3D voxel reconstruction from a single image with vision transformer," in *2021 International Conference on Culture-oriented Science & Technology (ICCSST)*, 2021, pp. 343–348.
- [10] V. Heimann, A. Spruck, and A. Kaup, "Frequency-selective mesh-to-mesh resampling for color upsampling of point clouds," in *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*, 2021, pp. 1–6.
- [11] P. de Oliveira Rente, C. Brites, J. Ascenso, and F. Pereira, "Graph-based static 3D point clouds geometry coding," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 284–299, 2019.

- [12] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li *et al.*, “Emerging mpeg standards for point cloud compression,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, 2019.
- [13] S. Gu, J. Hou, H. Zeng, and H. Yuan, “3D point cloud attribute compression via graph prediction,” *IEEE Signal Processing Letters*, vol. 27, pp. 176–180, 2020.
- [14] M. Bui, L.-C. Chang, H. Liu, Q. Zhao, and G. Chen, “Comparative study of 3D point cloud compression methods,” in *2021 IEEE International Conference on Big Data (Big Data)*, 2021, pp. 5859–5861.
- [15] S. Perry, H. P. Cong, L. A. da Silva Cruz, J. Prazeres, M. Pereira, A. Pinheiro, E. Dumić, E. Alexiou, and T. Ebrahimi, “Quality evaluation of static point clouds encoded using mpeg codecs,” in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3428–3432.
- [16] L. Hua, G. Jiang, M. Yu, and Z. He, “BQE-CVP: Blind quality evaluator for colored point cloud based on visual perception,” in *2021 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. IEEE, 2021, pp. 1–6.
- [17] Z. Zhang, W. Sun, X. Min, T. Wang, W. Lu, and G. Zhai, “No-reference quality assessment for 3D colored point cloud and mesh models,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7618–7631, 2022.
- [18] W. Zhou, Q. Yang, Q. Jiang, G. Zhai, and W. Lin, “Blind quality assessment of 3D dense point clouds with structure guided resampling,” *arXiv preprint arXiv:2208.14603*, 2022.
- [19] H. Su, Q. Liu, Y. Liu, H. Yuan, H. Yang, Z. Pan, and Z. Wang, “Bitstream-based perceptual quality assessment of compressed 3d point clouds,” *IEEE Transactions on Image Processing*, vol. 32, pp. 1815–1828, 2023.
- [20] Q. Liu, H. Su, T. Chen, H. Yuan, and R. Hamzaoui, “No-reference bitstream-layer model for perceptual quality assessment of v-pcc encoded point clouds,” *IEEE Transactions on Multimedia*, vol. 25, pp. 4533–4546, 2023.
- [21] A. Chetouani, M. Quach, G. Valenzise, and F. Dufaux, “Deep learning-based quality assessment of 3D point clouds without reference,” in *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2021, pp. 1–6.
- [22] Y. Liu, Q. Yang, Y. Xu, and L. Yang, “Point cloud quality assessment: Dataset construction and learning-based no-reference metric,” *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 19, no. 2s, pp. 1–26, 2023.
- [23] Z. Shan, Q. Yang, R. Ye, Y. Zhang, Y. Xu, X. Xu, and S. Liu, “GPA-Net: No-reference point cloud quality assessment with multi-task graph convolutional network,” *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–13, 2023.
- [24] S. Wang, X. Wang, H. Gao, and J. Xiong, “Non-local geometry and color gradient aggregation graph model for no-reference point cloud quality assessment,” in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 6803–6810.
- [25] M. Tliba, A. Chetouani, G. Valenzise, and F. Dufaux, “Efficient deep-based graph metric for point cloud quality assessment,” in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2023)*, 2023.
- [26] W. Tao, G. Jiang, Z. Jiang, and M. Yu, “Point cloud projection and multi-scale feature fusion network based blind quality assessment for colored point clouds,” in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 5266–5272.
- [27] Q. Liu, H. Yuan, H. Su, H. Liu, Y. Wang, H. Yang, and J. Hou, “PQA-Net: Deep no reference point cloud quality assessment via multi-view projection,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4645–4660, 2021.
- [28] R. Tu, G. Jiang, M. Yu, T. Luo, Z. Peng, and F. Chen, “V-PCC projection based blind point cloud quality assessment for compression distortion,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 2, pp. 462–473, 2023.
- [29] Q. Yang, Y. Liu, S. Chen, Y. Xu, and J. Sun, “No-reference point cloud quality assessment via domain adaptation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2022, pp. 21 179–21 188.
- [30] W. Xie, K. Wang, Y. Ju, and M. Wang, “pmbqa: Projection-based blind point cloud quality assessment via multimodal learning,” in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 3250–3258.
- [31] Z. Zhang, W. Sun, H. Wu, Y. Zhou, C. Li, Z. Chen, X. Min, G. Zhai, and W. Lin, “GMS-3DQA projection-based grid mini-patch sampling for 3d model quality assessment,” *ACM Transactions on Multimedia Computing, Communications and Applications*, 2023.
- [32] Z. Zhang, W. Sun, X. Min, Q. Zhou, J. He, Q. Wang, and G. Zhai, “MM-PCQA: Multi-modal learning for no-reference point cloud quality assessment,” *Proceedings of the International Joint Conference on Artificial Intelligence*, 2023.
- [33] N. Jmour, S. Zayen, and A. Abdelkrim, “Convolutional neural networks for image classification,” in *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, 2018, pp. 397–402.
- [34] Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei, “Fully convolutional adaptation networks for semantic segmentation,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6810–6818.
- [35] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, “Deep Convolutional Neural Networks for Computer-Aided Detection: CNN architectures, dataset characteristics and transfer learning,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.
- [36] G. Li, M. Müller, G. Qian, I. C. Delgado, A. Abualshour, A. Thabet, and B. Ghanem, “DeepGCNs: Making GCNs Go as Deep as CNNs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 6923–6939, 2023.
- [37] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, “A comprehensive survey on graph neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.
- [38] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, and C. Zou, “Hypergraph Learning: Methods and practices,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2548–2566, 2022.
- [39] X. Lv, L. Wang, Q. Zhang, N. Zheng, and G. Hua, “Video object co-segmentation from noisy videos by a multi-level hypergraph model,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 2207–2211.
- [40] Z. Zhang, H. Lin, X. Zhao, R. Ji, and Y. Gao, “Inductive multi-hypergraph learning and its application on view-based 3d object classification,” *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 5957–5968, 2018.
- [41] L. Nong, J. Peng, W. Zhang, J. Lin, H. Qiu, and J. Wang, “Adaptive multi-hypergraph convolutional networks for 3d object classification,” *IEEE Transactions on Multimedia*, pp. 1–14, 2022.
- [42] S. Yu, D. Niu, X. Zhao, and M. Liu, “Color image retrieval based on the hypergraph and the fusion of two descriptors,” in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2017, pp. 1–6.
- [43] H. Yu, C. Deng, L. Zhao, L. Hao, X. Liu, W. Lu, and H. You, “A light-weighted hypergraph neural network for multimodal remote sensing image retrieval,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 2690–2702, 2023.
- [44] D. Du, H. Qi, L. Wen, Q. Tian, Q. Huang, and S. Lyu, “Geometric hypergraph learning for visual tracking,” *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4182–4195, 2017.
- [45] R. Mekuria, Z. Li, C. Tulvan, and P. Chou, “Evaluation criteria for point cloud compression,” *ISO/IEC MPEG*, no. 16332, 2016.
- [46] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, “Geometric distortion metrics for point cloud compression,” in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3460–3464.
- [47] R. Mekuria, S. Lasserre, and C. Tulvan, “Performance assessment of point cloud compression,” in *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2017, pp. 1–4.
- [48] E. Alexiou and T. Ebrahimi, “Point cloud quality assessment metric based on angular similarity,” in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1–6.
- [49] G. Meynet, J. Digne, and G. Lavoué, “PC-MSDM: A quality metric for 3D point clouds,” in *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2019, pp. 1–3.
- [50] A. Javaheri, C. Brites, F. Pereira, and J. Ascenso, “A generalized hausdorff distance based quality metric for point cloud geometry,” in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2020, pp. 1–6.
- [51] G. Meynet, Y. Nehmé, J. Digne, and G. Lavoué, “PCQM: A full-reference quality metric for colored 3D point clouds,” in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2020, pp. 1–6.
- [52] I. Viola, S. Subramanyam, and P. Cesar, “A color-based objective quality metric for point cloud contents,” in *2020 Twelfth International*

- Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2020, pp. 1–6.
- [53] E. Alexiou and T. Ebrahimi, “Towards a point cloud structural similarity metric,” in *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2020, pp. 1–6.
- [54] R. Diniz, P. G. Freitas, and M. C. Farias, “Towards a point cloud quality assessment model using local binary patterns,” in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2020, pp. 1–6.
- [55] R. Diniz, P. G. Freitas, and et al., “Multi-distance point cloud quality assessment,” in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3443–3447.
- [56] R. Diniz, P. G. Freitas, and M. C. Farias, “Local luminance patterns for point cloud quality assessment,” in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSp)*. IEEE, 2020, pp. 1–6.
- [57] Q. Yang, Z. Ma, Y. Xu, Z. Li, and J. Sun, “Inferring point cloud quality via graph similarity,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3015–3029, 2022.
- [58] Q. Yang, Y. Zhang, S. Chen, Y. Xu, J. Sun, and Z. Ma, “MPED: Quantifying point cloud distortion based on multiscale potential energy discrepancy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 6037–6054, 2023.
- [59] I. Viola and P. Cesar, “A reduced reference metric for visual quality evaluation of point cloud contents,” *IEEE Signal Processing Letters*, vol. 27, pp. 1660–1664, 2020.
- [60] Q. Liu, H. Yuan, R. Hamzaoui, H. Su, J. Hou, and H. Yang, “Reduced reference perceptual quality model with application to rate control for video-based point cloud compression,” *IEEE Transactions on Image Processing*, vol. 30, pp. 6623–6636, 2021.
- [61] H. Su, Q. Liu, H. Yuan, Q. Cheng, and R. Hamzaoui, “Support vector regression-based reduced-reference perceptual quality model for compressed point clouds,” *IEEE Transactions on Multimedia*, pp. 1–12, 2023.
- [62] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [63] Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Multiscale structural similarity for image quality assessment,” in *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. Ieee, 2003, pp. 1398–1402.
- [64] Z. Wang and Q. Li, “Information content weighting for perceptual image quality assessment,” *IEEE Transactions on Image Processing*, vol. 20, no. 5, pp. 1185–1198, 2011.
- [65] H. R. Sheikh and A. C. Bovik, “Image information and visual quality,” *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 430–444, 2006.
- [66] X. G. Freitas, R. Diniz, and M. C. Farias, “Point cloud quality assessment: unifying projection, geometry, and texture similarity,” *The Visual Computer*, pp. 1–8, 2022.
- [67] W. Zhou, G. Yue, R. Zhang, Y. Qin, and H. Liu, “Reduced-reference quality assessment of point clouds via content-oriented saliency projection,” *IEEE Signal Processing Letters*, vol. 30, pp. 354–358, 2023.
- [68] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Zeevi, “The farthest point strategy for progressive image sampling,” in *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 2 - Conference B: Computer Vision & Image Processing. (Cat. No.94CH3440-5)*, 1994, pp. 93–97 vol.3.
- [69] S. Chen, D. Tian, C. Feng, A. Vetro, and J. Kovačević, “Contour-enhanced resampling of 3D point clouds via graphs,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2941–2945.
- [70] S. Chen, D. Tian, C. Feng, A. Vetro, and et al., “Fast resampling of three-dimensional point clouds via graphs,” *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 666–681, 2018.
- [71] V. Heimann, A. Spruck, and A. Kaup, “Frequency-selective mesh-to-mesh resampling for color upsampling of point clouds,” in *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSp)*, 2021, pp. 1–6.
- [72] J. Xu, W. Zhou, and Z. Chen, “Blind omnidirectional image quality assessment with viewport oriented graph convolutional networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1724–1737, 2020.
- [73] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 770–778.
- [74] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 7132–7141.
- [75] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018, pp. 3–19.
- [76] A. G. Roy, N. Navab, and C. Wachinger, “Concurrent spatial and channel ‘squeeze & excitation’ in fully convolutional networks,” in *Medical Image Computing and Computer Assisted Intervention—MICCAI 2018: 21st International Conference, Granada, Spain, September 16–20, 2018, Proceedings, Part I*. Springer, 2018, pp. 421–429.
- [77] Y. Yang, “An evaluation of statistical approaches to text categorization,” *Inf. Retr.*, vol. 1, no. 1–2, p. 69–90, may 1999.
- [78] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [79] Y. Huang, Q. Liu, and D. Metaxas, “Video object segmentation by hypergraph cut,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1738–1745.
- [80] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai, “3-D object retrieval and recognition with hypergraph analysis,” *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4290–4303, 2012.
- [81] M. Wang, X. Liu, and X. Wu, “Visual classification by ℓ_1 -hypergraph modeling,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2564–2574, 2015.
- [82] Q. Liu, Y. Sun, C. Wang, T. Liu, and D. Tao, “Elastic net hypergraph learning for image clustering and semi-supervised classification,” *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 452–463, 1 2017.
- [83] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, “Hypergraph neural networks,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3558–3565.
- [84] J. Huang, X. Huang, and J. Yang, “Residual enhanced multi-hypergraph neural network,” in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 3657–3661.
- [85] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, and C. Zou, “Hypergraph Learning: Methods and practices,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2548–2566, 2022.
- [86] G. Li, M. Müller, G. Qian, I. C. Delgado, A. Abualshour, A. Thabet, and B. Ghanem, “DeepGCNs: Making gcns go as deep as cnns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 6923–6939, 2023.
- [87] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [88] Q. Yang, H. Chen, Z. Ma, Y. Xu, R. Tang, and J. Sun, “Predicting the perceptual quality of point cloud: A 3D-to-2D projection-based exploration,” *IEEE Transactions on Multimedia*, vol. 23, pp. 3877–3891, 2021.
- [89] Q. Liu, H. Su, Z. Duanmu, W. Liu, and Z. Wang, “Perceptual quality assessment of colored 3d point clouds,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 8, pp. 3642–3655, 2023.
- [90] X. Wu, Y. Zhang, C. Fan, J. Hou, and S. Kwong, “Subjective quality database and objective study of compressed point clouds with 6dof head-mounted display,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4630–4644, 2021.
- [91] V. Q. E. Group et al., “Final report from the video quality experts group on the validation of objective models of video quality assessment, phase ii,” 2003 *VQEG*, 2003.
- [92] 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 Transactions on Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.



Wu Chen received the B.E. degree in software engineering from the Hangzhou Normal University in 2022. He is currently pursuing the M.E. degree with the School of Information Science and Engineering, Ningbo University, Ningbo, China. His research interests include the application of 3D point clouds in computer vision, such as quality assessment and denoising.



Qiuping Jiang is currently a Professor with the School of Information Science and Engineering, Ningbo University, Ningbo, China. His research interests include image quality assessment, visual perception modeling, and water-related vision. He is an Associate Editor or on the Editorial Board of several SCI-indexed journals such as *Displays*, *Journal of Visual Communication and Image Representation*, *IET Image Processing*, and *Journal of Electronic Imaging*.



Wei Zhou is an Assistant Professor at Cardiff University, United Kingdom. Dr. Zhou was a Postdoctoral Fellow at University of Waterloo, Canada. Wei received the Ph.D. degree from the University of Science and Technology of China in 2021, joint with the University of Waterloo from 2019 to 2021. Dr. Zhou was a visiting scholar at National Institute of Informatics, Japan, a research assistant with Intel, and a research intern at Microsoft Research and Alibaba Cloud. Wei is now an Associate Editor of IEEE Transactions on Neural Networks and Learning Systems. Wei's research interests span multimedia computing, perceptual image processing, and computational vision.



Long Xu (Senior Member, IEEE) received the M.S. degree in applied mathematics from Xidian University, Xi'an, China, in 2002, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing, China. He was a Postdoctoral Researcher with the Department of Computer Science, City University of Hong Kong, from July 2009 to December 2012, and the Department of Electronic Engineering, The Chinese University of Hong Kong, from August 2009 to December 2012. From January 2013 to March 2014, he was a Postdoctoral Researcher with the School of Computer Engineering, Nanyang Technological University, Singapore. Currently, he is with the Faculty of Electrical Engineering and Computer Science, Ningbo University, as a full professor. His current research interests include image/video processing, solar radio astronomy, wavelet, machine learning, and computer vision. He was selected into the 100-Talents Plan, CAS, in 2014.



Weisi Lin (Fellow, IEEE) received the bachelor's degree in electronics and the master's degree in digital signal processing from Sun Yat-Sen University, Guangzhou, China, and the Ph.D. degree in computer vision from King's College London, U.K.

He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include image processing, perceptual modeling, video compression, multimedia communication, and computer vision. He is a fellow of the IET, an Honorary Fellow of the Singapore Institute of Engineering Technologists, and a Chartered Engineer in U.K. He was awarded as a Distinguished Lecturer of the IEEE Circuits and Systems Society from 2016 to 2017. He served as a Lead Guest Editor for a Special Issue on Perceptual Signal Processing of the IEEE JSTSP in 2012. He has also served or serves as an Associate Editor for IEEE TIP, IEEE TCSVT, IEEE TMM, IEEE TNNLS, IEEE SPL, and Journal of Visual Communication and Image Representation. He was the Chair of the IEEE MMTC Special Interest Group on Quality of Experience.