Computers & Graphics 126 (2025) 104156

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

A multi-view projection-based object-aware graph network for dense captioning of point clouds☆

Zijing Ma [a,1], Zhi Yang [a,1], Aihua Mao [a,*], Shuyi Wen [a], Ran Yi [b], Yongjin Liu [c]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510000, Asia, China
[b] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200000, Asia, China
[c] BNRist, Department of Computer Science and Technology, MOE-Key Laboratory of Pervasive Computing, Tsinghua University, Beijing, 100000, Asia, China

ARTICLE INFO

Keywords:
Point clouds
3D dense captioning
Multimodel
Graph network

ABSTRACT

3D dense captioning has received increasing attention in the multimodal field of 3D vision and language. This task aims to generate a specific descriptive sentence for each object in the 3D scene, which helps build a semantic understanding of the scene. However, due to inevitable holes in point clouds, there are often incorrect objects in the generated descriptions. Moreover, most existing models use KNN to construct relation graphs, which are not robust and have poor adaptability to different scenes. They cannot represent the relationship between the surrounding objects well. To address these challenges, in this paper, we propose a novel multi-level mixed encoding model for accurate 3D dense captioning of objects in point clouds. To handle holes in point clouds, we extract multi-view projection image features of objects based on our key observation that a hole in an object seldom exists in all projection images from different view angles. Then, the image features are fused with object detection features as the input of subsequent modules. Moreover, we combine KNN and DBSCAN clustering algorithms to construct a graph G and fuse their output features subsequently, which ensures the robustness of the graph structure for accurately describing the relationships between objects. Specifically, DBSCAN clusters are formed based on density, which alleviates the problem of using a fixed K value in KNN. Extensive experiments conducted on ScanRefer and Nr3D datasets demonstrate the effectiveness of our proposed model.

1. Introduction

Visual semantic comprehension (VSC) has drawn increasing interest in the cross-field of computer vision and natural language processing. So far, VSC in the context of 2D images is a widely studied concern to the majority of academics. In these years, many researchers have dynamically conducted multi-modal processing, which benefits several multi-modal research fields, e.g., image captioning [1,2], video captioning [3,4], and even visual question answering [5]. Recently, captioning for 3D scenes has been a new direction along with the easy acquisition of 3D point clouds. The relevant research is just at the beginning, but it is of crucial importance for robotics, augmented reality, and autonomous vehicles.

Similarly to the 2D VSC, captioning is an important research topic in 3D VSC and can be classified into different tasks, such as 3D dense captioning [6], 3D visual question answering [7], etc. 3D scenes are usually inputted in the form of point clouds, which differs from 2D scenes in that they typically take the shape of point clouds. In contrast to the regular structure in 2D images, point clouds have several points that are not in any particular order. Accordingly, 3D dense captioning differs significantly from traditional 2D image captioning in that it generates descriptions for each object in a 3D scene, focusing on the characteristics of each object (what it is, how it looks, etc.) and how it interacts with other objects (what is next to it, what are the objects next to it, what is the spatial position relationship between them, etc.). This undoubtedly raises the bar for the model's comprehension of the scene.

Given the construction of high-quality 3D point cloud datasets, some pioneering works have been developed for 3D VSC. E.g., Chen et al. [6] proposed a solution to a 3D dense captioning task based on the ScanRefer [8] dataset. Note that in this 3D dense captioning task, the object detection backbone network, which is analogous to the detection network of the region features in the field of 2D images, is required in

☆ This article was recommended for publication by R. Hu.

* Corresponding author.
E-mail address: ahmao@scut.edu.cn (A. Mao).

[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.cag.2024.104156
Received 23 June 2024; Received in revised form 2 December 2024; Accepted 19 December 2024
Available online 31 December 2024

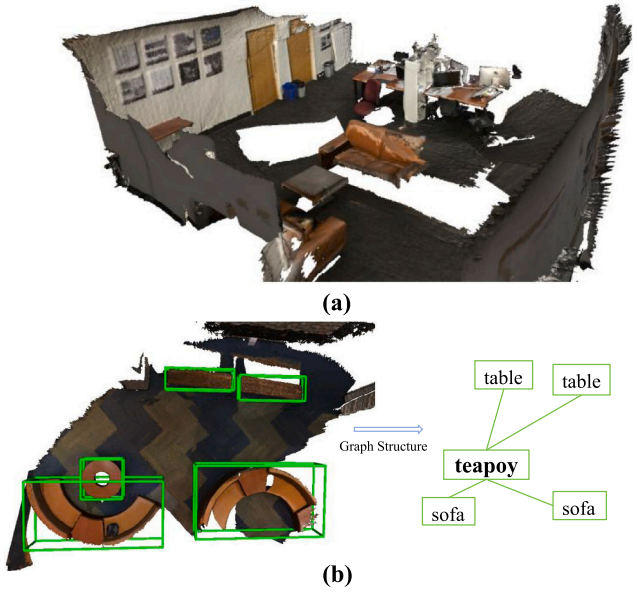


Fig. 1. Problems in the point cloud data and existing methods. (a) shows holes in the point cloud in the ScanNet dataset. (b) demonstrates adjacent objects to the middle teapoy, showing the problem of using KNN to build relation graphs in some scenarios, i.e., in a sparse scene, for each object, all objects will be modeled as adjacent neighbors to it.

the preprocessing of the point cloud to describe each object in a 3D scene.

3D dense captioning involves generating a descriptive sentence for each object in a provided 3D point cloud. Due to the disordered nature and generally low quality of the point cloud data, it is very challenging to apply 2D image captioning techniques. Even with the high-quality point cloud dataset, there are still many holes in the point clouds (see an example in Fig. 1(a), yet these issues rarely appear in 2D images. Below, we summarize the key challenges of the 3D dense caption task: 1) The majority of existing techniques use additional input features to compensate for the low quality of the point cloud, such as taking into account the color and normal vectors of the point cloud. These features are shallow and can be directly obtained from the original point cloud. They did not consider how to extract deeper and unique features from the initial point cloud, such as features from various viewpoints or the relational characteristics of objects within the scene; (2) Existing 3D dense captioning methods use the K nearest neighbor algorithm (KNN) [9] to determine the closest K objects for describing the relationships with surrounding objects. But KNN is significantly influenced by the value of K, and the graph structure constructed in many scenes has obvious errors, *e.g.,* when the scene only consists of a few objects, all objects (even those far away) may be considered adjacent, resulting in inaccurate descriptions (see an example in Fig. 1(b).

To tackle these challenges, in this paper, we propose a Multi-level Mix Encoding Model (MMEM), which is the integration of a multi-view feature extraction module with a graph-based feature interaction mechanism. First, we propose a Multi-view Feature Extraction Module (MFEM) for Point Clouds, significantly improving the representation quality despite missing data points. Specifically, the multi-view CLIP features [10] of an object are integrated into the feature representation without the need for additional features. This module aims to (1) project the object's point cloud from a number of different viewpoints, (2) use the pre-trained CLIP model to extract the CLIP features from different viewpoints as auxiliary features, and (3) fuse them with the object features extracted from a 3D object detection backbone (*e.g.,* VoteNet [11]). Concerning CLIP is capable to provide rich semantic information, we apply CLIP into extracting multi-view features from point cloud projections, making it particularly effective for feature representation for point clouds with holes. The CLIP model used for multi-view feature extraction is naturally suitable for the captioning task, since it learns a shared embedding space for images and texts. It should be noted that the contribution of this novelty comes from that we employ the CLIP module by a multi-view feature extraction mechanism instead of simply integrating it. In this way, our model ensures a more comprehensive understanding of the object's characteristics across different viewpoints, leading to more accurate captioning results. Second, we propose a Feature Interaction Module based on Mixed Graph Convolution (mixG) to correctly describe the relationships between objects. Leveraging the spatial relationships within the point cloud, this module facilitates efficient message passing and fusion mechanisms. It uses KNN and DBSCAN clustering algorithm [12] to construct two different graph structures, where DBSCAN helps to address the challenge caused by a fixed K value in KNN. Then we use two independent graph neural networks to model the object relationships and finally fuse the output features. Last, we use a Transformer-based semantic generation module to generate the description for each object. Thanks to these enhancements, our model is capable to generate more accurate captions for 3D point clouds, particular for the complex 3D scenes with holes, which is not addressed in the SOTA works.

In summary, our main contributions are threefold:

- A Multi-level Mix Encoding Model (MMEM) for a 3D dense captioning task is proposed. This model greatly enhances feature expressiveness by designing a novel multi-view projection-based feature representation strategy, and has concise structure and economic computational cost. Extensive experiments on ScanRefer [8] and Nr3D [13] datasets demonstrate the excellent performance of this model.
- Based on VoteNet features, we suggest projecting the object from the original point cloud from various viewpoints and extracting the multi-view features by the CLIP model, which makes the feature representations of objects more comprehensive and accurate, particularly effective to handle point clouds containing holes. We are the pioneer to apply the CLIP model into the task of 3D dense caption.
- We utilize KNN and DBSCAN to strengthen the representation of spatial relationships between objects in the scene in a complementary manner, which effectively extracts and describes the relationships between each object and its surrounding objects. Rather than adopting complex models, we delicately integrate these mature and relatively simple models and make the most advantage of them to improve the feature quality.

2. Related work

2.1. Captioning in 2D images

Image captioning is a popular research topic at the intersection of computer vision and natural language processing, which generates textual descriptions for images [14]. The methods in this field typically utilize a visual encoder (for extracting image features) and a text generation model such as the Transformer [15].

To achieve good performance, Anderson et al. [16] introduced a two-stage attention mechanism to extract salient image features and generate captions conditioned on the extracted features. By further expanding on the idea of utilizing multimodal contexts, Cornia et al. proposed a meshed transformer with memory [17]. Deng et al. devised LaBERT [18] for generating length-controllable captions.

Wu et al. [1] proposed a global-local discriminative objective to facilitate the generation of fine-grained descriptive captions. Zhang et al. [19] explored pairwise relationships adaptively from linguistic context in image captioning. Wang et al. [2] designed a text-guided

relation encoder to learn the visual representation consistent with human visual cognition. With the popularity of pre-training models (such as BERT [20]) in natural language processing tasks, some researchers have also utilized pre-training models for visual language tasks, such as ViLBERT [21], VL-BERT [22].

Researchers also extended image captioning for video captioning, which aims to automatically generate natural-language descriptions of the events in videos. Song et al. [23] modeled a semantic-rich context learning for recognizing facts and emotions in the video. Aafaq et al. [3] proposed a visual-semantic embedding framework for dense video captioning that exploits visual information as well as the associated linguistic content in the event. Zhang et al. [4] designed a graph-based partition-and-summarization framework to address the problem of scene evolution within an event for dense video captioning.

In the field of visual semantic comprehension (VSC), there is already abundant research on 2D image and video captioning. In this paper, we pay attention to 3D VSC and focus on captioning for 3D point clouds.

2.2. Dense captioning in 3D point clouds

Given the construction of high-quality 3D point cloud datasets such as ScanRefer [8] and Nr3D [13], as summarized below, some classical works have been developed in the field of point cloud and language modeling.

Noting that the 3D dense caption task requires a detailed description of each object. It is necessary to use a backbone network for object detection in point cloud preprocessing, analogous to the region features employed in 2D image captioning.

Scan2Cap [6] is a pioneering model proposed for 3D dense captioning, which primarily consists of a point cloud object detection module, a graph learning module, and a context-aware attention mechanism generator. The point cloud object detection module is based on the classic VoteNet [11] model. This module utilizes PointNet++[24] to extract the point cloud features, which are then used to vote for seed points through Hough voting and subsequently generate object detection bounding boxes. In the graph learning module, for each object, KNN [9] is used to find the nearest K objects to generate the graph structure, and then the standard graph neural network, based on message passing [25] is used to update the node features. Finally, the generation module employs an attention mechanism and RNN with autoregressive decoding to generate text. X-Trans2Cap [26] is based on the M2-Transformer architecture in the 2D domain. By incorporating a novel cross-modal fusion module into transformers, a feature alignment method, and an improved knowledge extraction method, it effectively eliminates excess computational burden during training and enhances knowledge transfer. In contrast, MORE [27] encodes the relationship between objects gradually. It initially encodes one-dimensional spatial relationships using the volume product of the spatial layout and subsequently extracts and encodes the 3D spatial relationship using a 3D attention graph. REMAN [28] defines the specific positional relationship according to the bounding box and proposes a feature transfer module to narrow the feature gap between vision and language. However, these existing models typically rely on additional input features to compensate for the low quality of the point cloud. Our work deals with low-quality point clouds with holes by multi-view projected image features, which results in abundant and accurate feature representations.

2.3. 2D-3D feature fusion

Given that scanned point cloud data are usually of low quality, it is common practice to leverage corresponding images to assist learning. In Frustum PointNets [29], the 3D frustum of a point cloud region of the object is extracted through 2D object detection. This procedure is followed by the application of two modified versions of PointNet to achieve segmentation and detection. PointFusion [30] designs two fusion modules: global fusion and dense fusion, to fuse image and point cloud features. The point features utilized in Point-MVSNet [31] are composed of 2D features extracted from images under multi-scale conditions and normalized 3D coordinates. MVX-Net [32] proposes Voxel fusion, which supplements the feature vector of each voxel in the point cloud with the ROI feature vector. However, voxel fusion is unable to fuse all point features. Color information of images is attached to the voxel through projection in [33], and the concept is further developed by employing the 3D discrete convolutional neural network [34]. In [35], the Query Fusion Mechanism (QFM) is proposed by introducing an operation based on self-attention, which adaptively combines point cloud and image features. In this mechanism, each point cloud voxel will query all image voxels to achieve homogeneous feature fusion and combine them with the original point cloud voxel features to form joint camera LiDAR features.

2.4. Graph-based method

Graph structure is very common in social networks and is also used to model the spatial relationship between objects in images. The core idea of graph-based methods is that each node in the graph is updated by aggregating the features of adjacent nodes. Graph-based methods have been widely used to enhance image understanding, thereby facilitating image captioning and visual question answering. Li et al. [36] proposed Re-GAT, in which an image is transformed into a spatial graph and a semantic graph. Then, the node information is updated through a graph convolution network and a graph attention network. Finally, deep relationships between various nodes were learned. Huang et al. [37] proposed DC-GCN, in which some irrelevant object relations are pruned in the graph structure. Two objects with minimal overlapping area are considered to have a weak relationship. Chen et al. [38] and Duc Minh Vo et al. [39] utilized ConceptNet [40] to construct a commonsense knowledge graph for images. Yang et al. proposed TSG [41] which contains a multi-head attention GNN [42] for embedding scene graphs. Scan2Cap [6] uses KNN to construct a graph, where each node is connected only to the K nearest objects. MORE [27] defines a clear spatial relation on graph, and uses Object-Central Triplet Attention Graph to mine the second-order spatial relation. All the aforementioned models utilize KNN to construct graphs, ignoring the influence of using a fixed K value in different scenarios. In this paper, to address this challenge, we employ DBSCAN [12] to construct the relational graph.

3. Methods

We build up the pipeline of our model by following Scan2Cap [6] as the baseline. However, different from Scan2cap, at the stage of feature extraction, we introduce a novel strategy to extract multi-view features from point clouds, in order to enhance feature expressiveness, especially for point clouds containing holes that are not well addressed in the previous works. Furthermore, at the stage of feature interaction, we enhance the expression of spatial relationships between objects by delicately combining KNN and DBSCAN, which work in a complementary manner. These new efforts greatly improve the overall performance and particularly effectively work for the point clouds with holes.

Our model consists of three modules: a multi-view feature extraction module (MFEM) for a point cloud, a feature interaction module based on mixed graph convolution (mixG), and a semantic generation module based on Transformer. 1) MFEM module: The input point cloud is initially processed by MFEM to extract multi-view features. This involves extracting object detection features and multi-view projection features based on CLIP, which are combined by the bridge module. The MFEM module helps to handle the challenge related to holes in the point cloud. (2) mixG module: The obtained features are then sent to the mixG, which simultaneously utilizes the graph convolution network,

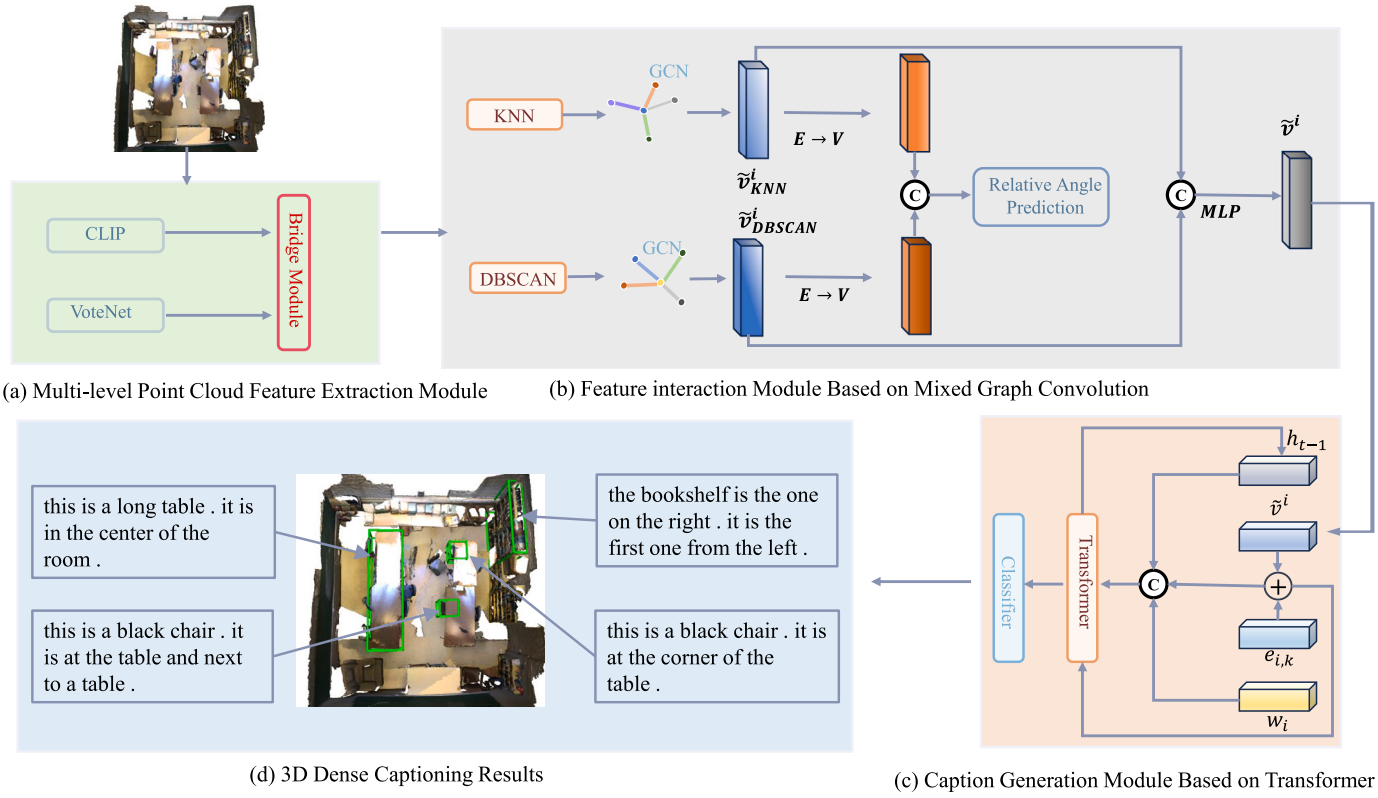


Fig. 2. An overview of our model. The Multi-view Feature Extraction Module for Point Cloud (MFEM, (a)) first detects all objects in the point cloud and extracts multi-scale features including VoteNet features and multi-view features based on CLIP. After that, these features are updated through the interaction between graph nodes through a Mixed Graph Convolution module based on KNN and DBSCAN (mixG, (b)). Finally, a concrete description of each object is generated by a semantic generation module based on Transformer (c).

KNN, and DBSCAN clustering to construct the graph structure. (3) Semantic generation module: In this module, the standard GRU [43] recurrent neural network is replaced with Transformer to generate the description. An overview of our model is illustrated in Fig. 2.

3.1. Multi-view feature extraction module for a point cloud

In this module, we aim to extract multi-view features for each object in the input point cloud. We first detect objects from the input point cloud using a 3D object detection backbone and extract object-level features. We then extract CLIP features from each object's projections at different angles, where the multi-view projection is designed to help alleviate the difficulties caused by holes in the point cloud. The object detection features and multi-view CLIP features are then fused by a bridge module.

3.1.1. Object detection features based on VoteNet

Since the 3D dense captioning task requires generating a description for each object, a 3D object detection backbone network and object-level features are required. VoteNet [11] excels in 3D object detection by leveraging a novel voting mechanism, providing robustness and accuracy surpassing existing algorithms. In view of its outstanding performance, we adopt a pre-trained VoteNet as the object detection backbone to extract object features.

We input a point cloud into VoteNet, and the outputs include the bounding box of the M detected objects $B_{\text{Vote}} = \left\{ b^i_{\text{Vote}} \right\}_{i=1}^{M} \in \mathbb{R}^{6\times M}$ (where the number 6 represents the 3D center coordinates and the size of each bounding box), and the accompanying features $V_{\text{Vote}} = \left\{ v^i_{Vote} \right\}_{i=1}^{M} \in \mathbb{R}^{d\times M}$ (where d indicates the dimension of the candidate object features).

3.1.2. Multi-view features of a single object based on CLIP

To cope with the gaps and holes in point cloud data (Fig. 1a), we propose to project the point cloud of each object along a number of different viewpoints and extract features from the projections using CLIP [10]. The reason behind this design is that the holes in the point cloud will be occluded and become invisible along some viewpoints. The corresponding extracted features can provide meaningful and supplementary information for understanding the 3D scene.

Motivated by the success of [44], we attempt to utilize CLIP features [10] into the task of 3D dense captioning. CLIP is a multimodal model trained on 400 million image-text pairs, excelling in the 2D vision-language domain. It employs an image encoder and a text encoder to extract features from each modality, aligning their feature spaces for effective representation. This pre-trained model can be easily adapted for downstream tasks, enhancing our approach's performance in 3D dense captioning.

To obtain multi-view features of a single object, given a point cloud P of a 3D scene consisting of M objects, we first need to obtain the point cloud of a single object. We consider two approaches to get the point cloud of a target object. The first approach is to use the bounding box information that is available from 3D object detection in Section 3.1.1. We use the bounding box to crop parts of the point cloud from the whole point cloud. The second approach uses an additional semantic segmentation model to obtain an object's segmentation results. In this paper, we utilize PointNet++[24] as the semantic segmentation model. The comparison of the two methods for cropping the scene to obtain single-object features is analyzed in detail in Section 4.4.4. We then denote the point cloud of a single object as P^{sub}.

To effectively capture object features, we project the point cloud of a single object onto 2D images using multi-view projections. This strategy is significant to improve the limitation of a single viewpoint which may render the object unrecognizably. For instance, viewing a chair from

the top would project it as a rectangle. Furthermore, due to possible holes in point clouds, it may not be possible to obtain good features for objects. However, after projection from certain perspectives, the missing parts will overlap with the non-missing parts to avoid the negative impact of holes, which can effectively assist in the expression of missing parts, and enhance the feature expressiveness, thus improve the performance of 3D dense captioning.

In our model, we generate multiple viewpoints by projecting the point clouds onto coordinate planes without a viewpoint selection module. Specially, we conduct multi-view projection by first rotating the object using F different angles, and then conducting Z-axis projection (projecting (x, y, z, r, g, b) to (x, y, r, g, b)). To capture comprehensive features of the objects and achieve better feature alignment, we empirically set F as 10 in our experiment. Fig. 3 illustrates one example of multi-view projection. After projecting from F different viewpoints, each object has F projection images, and the features of F images can be extracted by using CLIP image encoder. Due to the possible missing points (*e.g.,* holes and gaps) in a point cloud, it is difficult to obtain good 3D features for certain objects. This issue can result in multi-view projections capturing imperfections in the point cloud, leading to inaccuracies in features derived from a single image. However, this problem can be mitigated by using multiple 2D images from different perspectives as input, which collectively enhance the accuracy of the features. In our model, after projection from multiple viewpoints, the missing points are occluded in certain viewpoints, thus attenuating the problem caused by holes and improving the understanding of the 3D objects. The feature space encoded by the CLIP model is also quite similar to the feature space of texts, allowing for a smoother feature space transformation when generating specific words and producing more accurate descriptions.

3.1.3. Bridge module based on VoteNet and CLIP

Using the techniques presented in Sections 3.1.1 and 3.1.2, the VoteNet feature V_{Vote} and CLIP feature f of an object are readily obtained. To further fuse these two types of features, we design a bridge module[2] as shown in Fig. 4. The module first concatenates F multi-view CLIP features of object i into one tensor and then uses a MLP to convert them into a global feature. The specific procedure can be formulated as:

$$f^i_{\text{global}} = \sigma\left(\text{concat}\left(f^i_{1\sim F}\right) W_1^T\right) W_2^T, \tag{1}$$

where f^i_j is the CLIP feature of the projection of object i under the jth viewpoint ($j = 1, \ldots, F$), σ is the activation function (ReLU), and W_1, W_2 are learnable parameters.

The objective of the above step is to aggregate features from all F viewpoints. Then, a MLP is applied to increase the dimension of f^i_{global}, which turns it back into F features. The dimension is now consistent with the initial input dimension of the multi-view CLIP features, and a residual connection is used to update the feature:

$$f^{i,a}_j = f^i_j + \sigma\left(f^i_{global} W^T_{3j}\right), \tag{2}$$

where σ is the activation function, and W_{3j} is a learnable parameter.

At this end, the multi-view CLIP features have been updated, but have not yet been integrated with the VoteNet feature. We use MLP to transform the dimension of the updated multi-view CLIP feature to that of the VoteNet feature, concatenate the two types of features, and then fuse them with another MLP:

$$V^i = \sigma\left(\text{concat}\left(V^i_{\text{Vote}}; \sigma\left(f^{i,a} W_4^T\right)\right) W_5^T\right), \tag{3}$$

where σ is the activation function, and W_4, W_5 are learnable parameters. This process relies solely on point cloud data, making it a highly versatile feature enhancement strategy, i.e., it can be seamlessly integrated to different types of models.

[2] When training this bridge module, we froze the parameters of CLIP image encoder, and only train the bridge module.

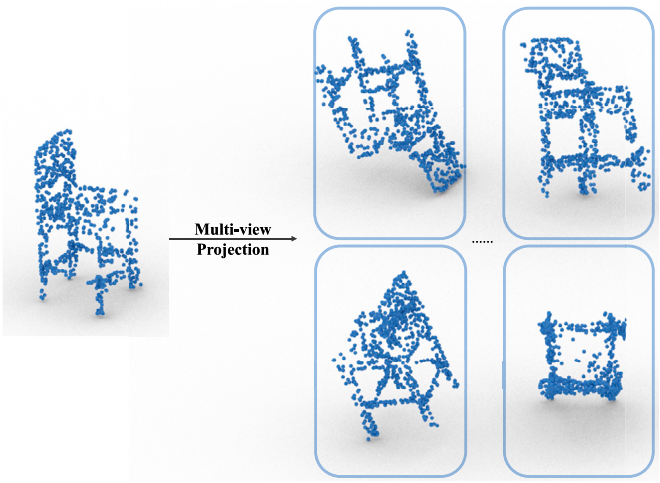


Fig. 3. Each object is rotated by aligning to multiple viewpoints. From each viewpoint, a 2D image is obtained by projection.

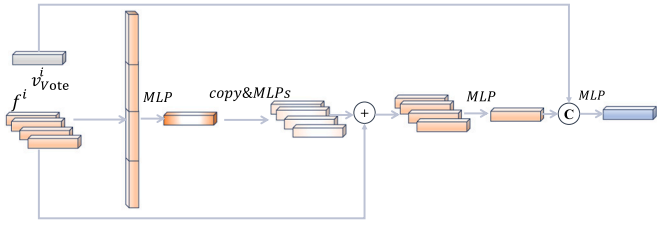


Fig. 4. The bridge module is designed to fuse CLIP features from multiple viewpoints with VoteNet features to get a better representation for each object.

3.2. Feature interaction module based on mixed graph convolution

3D dense captioning focuses on describing the relationship between an object and its surrounding objects, which can be modeled by graphs. Existing methods [6,28] construct a relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with the detected objects as nodes and the relationships between objects as edges. However, these methods use KNN [9] to construct graphs, treating the nearest K objects of each object as adjacent nodes. Since the value of K is fixed, in many scenes where objects are sparse, KNN will result in all objects in the scene (including those far away) being considered as adjacent nodes and thus provide inaccurate descriptions.

To address this challenge, we propose using a mixed graph convolution (mixG) with both KNN and DBSCAN [12] to build the graph structure, and construct a feature interaction module based on the relational graph module in Scan2Cap [6]. As shown in Fig. 2(b), we construct two different graph structures using the KNN algorithm and the DBSCAN clustering algorithm, respectively. The graph nodes are initialized with the features of the candidate frames, and the edge relationships are generated by KNN and DBSCAN, respectively.

3.2.1. Graph structure of KNN and DBSCAN

Given two objects X and Y, the distance from object X to object Y can be expressed as the minimal Euclidean distance from all vertices of X to the center of Y. Then, for each object in the scene, we utilize the KNN algorithm to identify the K nearest objects and construct a KNN graph $G_{KNN}(V, E)$, where V represents node features and E represents edge relationships. As mentioned above, when there are very few objects in the scene, KNN will consider all objects as adjacent nodes, causing the graph structure to become undifferentiated. Therefore, we further use the DBSCAN clustering algorithm to construct another graph structure, $G_{DBSCAN}(V, E)$. DBSCAN is a density-based clustering algorithm that can adaptively divide the scene into different clusters according to the

density of scene distribution. Compared with K-Means clustering [45], its major advantage is that it does not require specifying the number of clusters, which is especially suitable for scenarios where the number of clusters is uncertain. Although the graph constructed by DBSCAN could perform better than the one constructed by KNN, we found it is worthwhile to utilize the KNN graph together. The KNN graph focuses on capturing comprehensive global spatial relationships, while DBSCAN concentrates on the local adjacency relationships around objects. These two approaches are complementary. KNN can complement the limitations of DBSCAN, thereby enhancing the robustness and adaptability of our model when we use them in parallel. The ablation study of KNN and DBSCAN in Section 4.4.2 has shown that using both KNN and DBSCAN together yields better results compared to using either module alone.

3.2.2. Message passing and fusion mechanism

After constructing the two graph structures mentioned above, the standard mechanism [25] is utilized to enhance node features and extract object relation features. The edge convolution operation is defined as:

$$V \rightarrow E : e_{i,j}^{\tau+1} = f^{\tau}\left(\left[v_i^{\tau}, v_j^{\tau} - v_i^{\tau}\right]\right), \tag{4}$$

where v_i^{τ} and v_j^{τ} are the features of objects i and j at step τ, and $e_{i,j}^{\tau+1}$ is the directed edge relationship between objects i and j after the τth message passing. $[:]$ denotes the concatenation of two features. f is a nonlinear transformation which can be achieved by a MLP. After updating the edge relationship, the current node is updated by aggregating the node's edge relationships as follows:

$$E \rightarrow V : v_i^{\tau+1} = \sum_{j=1}^{K} e_{i,j}^{\tau+1}, \tag{5}$$

where $v_i^{\tau+1}$ denotes the feature of the ith object after the τth aggregation.

After several steps of message passing, a MLP is used to fuse the node features of the two different graph structures. We denote the fused node features as $\tilde{V} = \{\tilde{v}_i\}_{i=1}^{M}$.

3.2.3. Relative position judgment module

An auxiliary subtask to predict the relative orientation angle between two objects is defined to help learn the object relationships in mixG. As proposed in Scan2Cap [6], to accomplish this auxiliary task, an additional edge convolution operation is added at the last layer of each graph to extract object relation features, and a MLP layer is then added to predict the relative orientation angle between two objects. This orientation angle will be treated as a 6-classification problem rather than predicting a particular angle. Specifically, 0° to 180° is uniformly divided into 6 intervals. The reason behind this classification is that only the approximate orientation between two objects is needed, and the precise angle is not required.

3.3. Caption generation module based on transformer

The 3D dense captioning task needs to generate descriptive sentences, and LSTM [46] or GRU [43] is used in existing methods. In this paper, we replace them with a more powerful Transformer-based caption generation module. Unlike the LSTM or GRU, Transformer allows for parallelization of sequential data processing and offers more efficient handling of long-term dependencies within the sequence owing to its attention mechanism, thereby enhancing the performance of tasks that demand complex sentence generation. We therefore use a Transformer decoder to generate descriptions.

During the training phase of 3D dense captioning, each batch only selects one object for description, rather than generating descriptions of all objects in the scene in one batch. To enhance the relationship between each object in the scene and the target object k for description, the edge relationship output by the mixed graph module can be incorporated into the object features as follows, where the updated object features are denoted as $V_r = \{v_r^i\}_{i=1}^{M}$:

$$v_r^i = \tilde{v}_i + e_{i,k}. \tag{6}$$

The inputs to the Transformer decoder (TD) are Query, Key, and Value. Different from the traditional image description task, the 3D dense captioning task needs to specify the described object. Thus Query, Key, and Value cannot be directly set as the features of all objects. Query is the feature to be queried. In the traditional autoregressive task, Query is usually the word feature w_{t-1} predicted in the previous time step. However, in our task, because the description object of the current batch is specified, the feature of the current description object $v_r^{i,t-1}$ also needs to be included in the Query, otherwise the model does not know which description of object to generate. Finally, the hidden layer feature h_{t-1} output in the previous time step needs to be added to Query:

$$Query = \text{concat}\left(v_r^{i,t-1}, w_{t-1}, h_{t-1}\right). \tag{7}$$

For Key and Value, we extend the method in Scan2Cap [6] as follows. KNN is used to find the K nearest objects of the sampled object, followed by DBSCAN clustering to find all the objects within the corresponding cluster. The union of the features of these two groups of objects is used as the Key and Value of the decoder. The above approach continues the design in our mixed graph neural network, making the model only consider interactions with surrounding objects, and avoiding being influenced by distant objects. At each time step, the calculation process of the caption generation module based on Transformer can be briefly described as follows:

$$y_t^{\text{pred}} = MLP(TD(Query, Key, Value)), \tag{8}$$

where the descriptive sentences are generated in an autoregressive manner.

It is worth noting that during the testing phase, each batch of input contains all objects, and random sampling is not used, while descriptions are generated for each object sequentially through loop processing.

3.4. Loss functions

Relative angle loss. In Section 3.2.2, we defined an auxiliary subtask to learn the relative orientation angle between two objects. To supervise this subtask, we formulate the relative angle loss as follows:

$$L_{\text{angle}} = L_{\text{cross-entropy}}\left(y_{\text{angle}}^{i,j}, \hat{y}_{\text{angle}}^{i,j}\right), \tag{9}$$

where $y_{\text{angle}}^{i,j}$ is the ground-truth class label of the relative orientation angle between the ith and the jth objects (0° to 180° divided into six intervals, corresponding to six classes), and $\hat{y}_{\text{angle}}^{i,j}$ is the predicted class label. This loss greatly helps the model to better consider the spatial relationships between objects, and thus improves the accuracy of the final results. Particularly, our model strengthens the spatial relationships between objects captured by KNN and DBSCAN. The strengthened spatial features contributes more remarkable impact on the final results through this loss function.

Caption loss. Scene description needs to select the prediction vocabulary of the current time step from the vocabulary, so it can be regarded as a classification problem:

$$L_{\text{cap}} = L_{\text{cross-entropy}}\left(y_t, y_{gt}\right), \tag{10}$$

where y_t is the probability distribution of each word predicted by the model at the current time step, and y_{gt} is the one-hot encoding of the ground-truth word.

Finally, the overall loss function can be expressed as:

$$L_{total} = L_{\text{cap}} + L_{\text{angle}}. \tag{11}$$

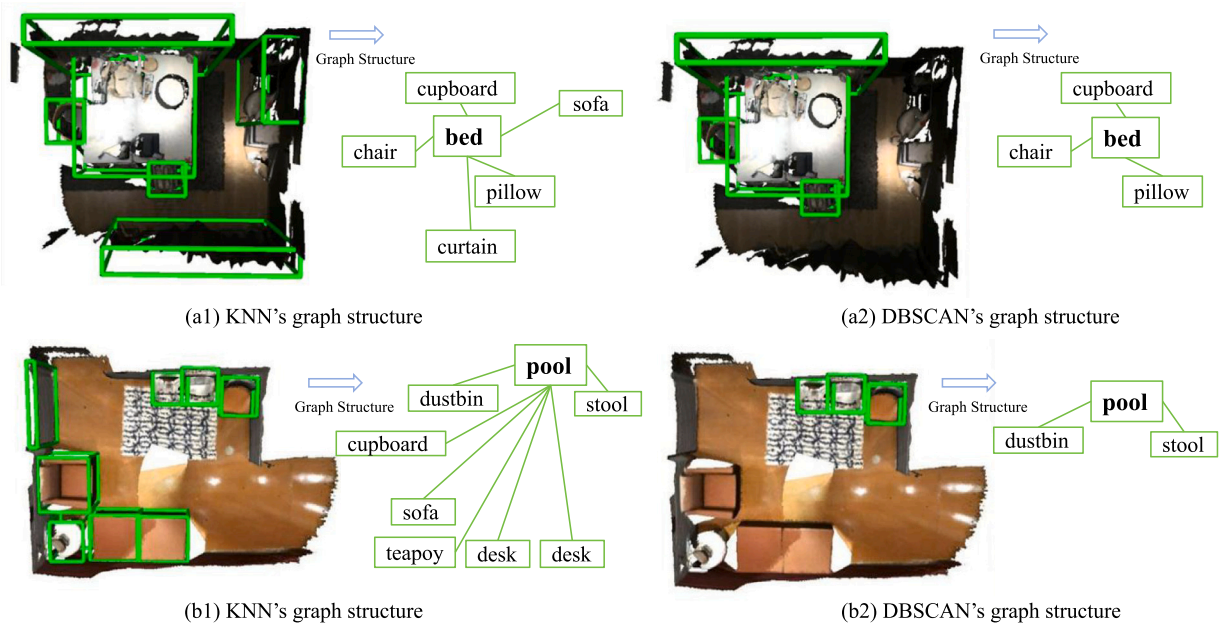


Fig. 5. Visual comparison of the object graph generated by KNN (on the left) and DBSCAN (on the right). The first and second rows depict the graph relational structures of the bed and the pool in relation to their surrounding objects.

4. Experiment

In this section, we first introduce the datasets and metrics, and then discuss the experimental settings and hardware environment. Based on these settings, extensive experiments have been carried out on the two mainstream datasets, ScanRefer and Nr3D, including the comparison with state-of-the-art models and the ablation studies of our model. We also provide plenty of visualization results for visual demonstration of our model.

4.1. Datasets and metrics

ScanRefer dataset: ScanRefer [8] is a 3D scene-text dataset based on ScanNet scene data. A total of 51,583 descriptions of 800 ScanNet [47] scenes were collected. On average, each scene has 13.81 objects and 64.48 descriptions, and each object has an average of 4.67 descriptions. Among them, 561 scenes were used for training, and 141 scenes were used for local testing. These descriptions are complex and diverse, covering more than 250 common indoor items. Due to the complexity of descriptions, one of the key challenges is to determine which parts of the descriptions describe the target object and which parts describe the adjacent objects. There are a total of 41,034 mentioned object attributes, such as color, shape, size, etc., in all descriptions. This dataset is mainly used for 3D text-object location, *i.e.,* given a sentence, the model needs to find out the position of each object mentioned in this sentence in the scene. Chen et al. [6] first used this dataset to describe 3D dense semantics, *i.e.,* to generate a sentence for each object to describe its characteristics in the scene.

Nr3D dataset: Nr3D [13] is similar to ScanRefer [8], and is also a 3D text-object positioning data set based on ScanNet scenes. It contains a total of 41,053 descriptions, covering about 100 indoor object categories. The major difference between Nr3D and ScanRefer is that the template level of the text descriptions in Nr3D is relatively low, *i.e.,* there are fewer sentences with a fixed pattern of "this is a ..", which makes Nr3D more challenging.

Input point cloud form: The input point cloud to our model consists of the geometry (3D coordinates) and additional point features including color and normal vectors. The ScanNet dataset also provides height data, which is utilized to indicate how far each point is from the ground. Additionally, ScanNet provides video data of indoor scenes. We can obtain multiple perspective views of several sub-scenes by sampling the video data every other frame. We then feed these views to the ENET [48] to extract the features from 2D scanned images and map them to the corresponding points. The final point cloud contains 135-dimensional point features, where the first 3 dimensions provide location information and the remaining 132 dimensions are additional point features.

Metrics: The quantitative metrics for experimental comparison are BLEU-4 [49], CIDEr [50], ROUGE [51], and METEOR [52], which are denoted as B-4, C, R, and M respectively. The higher values of these metrics mean better performance.

4.2. Implementation details

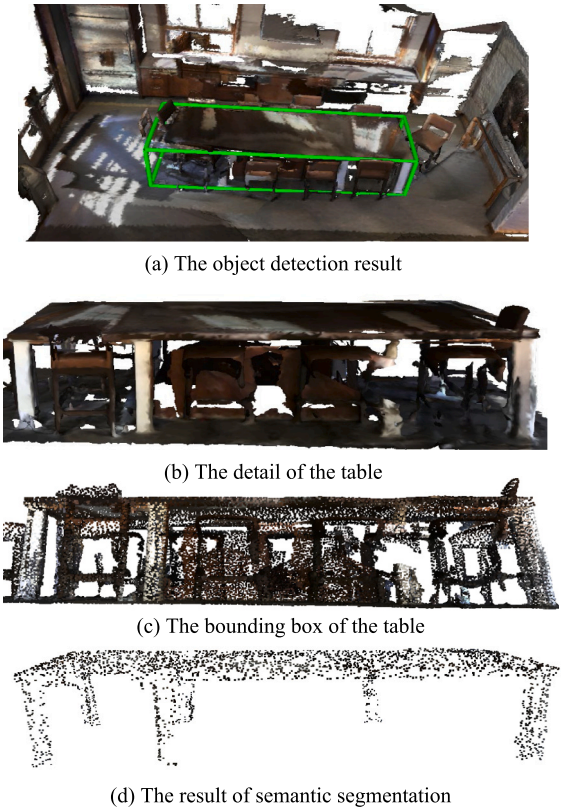
We implement the proposed model based on Scan2Cap, and most of hyper-parameters are consistent with Scan2Cap. Our model is trained for 60,000 iterations until it converges. The batch size is set to 12, the number of items identified by the target is $M = 64$, the hidden layer feature dimension is $d = 128$, and the verification is performed every 2000 iterations. DBSCAN's eps and minPts settings are configured to 0.7 and 2, respectively. The KNN algorithm's parameter K is set to 10. In the feature extraction module, we project each object from 10 different views, namely the number of viewpoints is $F = 10$. The hidden layer feature dimension $d = 512$ and VIT/32 [53] are used in the CLIP model's backbone network. The weight attenuation factor is adjusted to $1e-5$ to prevent overfitting. To increase the randomness of point cloud scenes and conduct data augmentation, the point cloud is randomly rotated between $-5°$ and $+5°$ and translated randomly in all directions within a range of 0–5 m. Since the point cloud in ScanNet is not exactly aligned with the ground, the rotation occurs around the coordinate axis. The overall decoding length in the experiment is 32, given that descriptions longer than 30 are trimmed, and [SOS] and [EOS] tags are put before and after the phrase to denote the start and end of the description. All the experiments are conducted on a single RTX 3090 with 24 GB of VRAM. It is noted that the utilized models in our pipeline such as pre-trained CLIP and KNN and DBSCAN which have low computational cost. Furthermore, our model is trained in an end-to-end manner leading to easy application. Overall, our model is lightweight and has good potential to achieve real-time performance.

Table 1
Comparison with state-of-the-art methods on the ScanRefer dataset. (*) denotes reproduced results under the same experimental conditions and annotated datasets with our experiment. The best results are indicated in bold.

Method	B-4	C	R	M	Running Time
Scan2Cap(*) [6]	39.79	63.63	62.31	28.84	30h
TransCap(*) [26]	40.62	68.27	62.79	28.65	**15h**
MORE(*) [27]	42.01	69.68	63.78	29.49	33h
REMAN(*) [28]	43.48	**73.32**	64.99	**30.05**	32h
Ours	**44.00**	70.92	**65.28**	29.81	28h

Table 2
Comparison with state-of-the-art methods on the Nr3D dataset. (*) denotes reproduced results under the same experimental conditions and annotated datasets with our experiment. The best results are indicated in bold.

Method	B-4	C	R	M
Scan2Cap(*) [6]	32.68	60.79	64.89	28.58
TransCap(*) [26]	32.98	63.1	65.49	29.18
MORE(*) [27]	33.78	64.89	65.99	29.48
REMAN(*) [28]	32.94	65.52	65.20	29.45
Ours	**34.01**	**66.03**	**66.06**	**29.56**



(a) The object detection result

(b) The detail of the table

(c) The bounding box of the table

(d) The result of semantic segmentation

Fig. 6. The visual results of point cloud of a table, which are cropped by bounding box (c) and semantic segmentation (d) respectively.

4.3. Quantitative comparison with state-of-the-art models

This section provides a comparison of our proposed model with state-of-the-art models, including OracleRetr3D [6], Scan2Cap [6], TransCap,[3] MORE [27] and REMAN [28]. Since our experiments were conducted on the datasets with ground-truth annotations for bounding boxes and semantic segmentation, in order to ensure a fair comparison, we retrained the SOTA models under the same experimental conditions and annotated datasets to reproduced the results.

Table 1 presents the quantitative comparison results on the ScanRefer dataset. The content marked with (*) indicates the results reproduced in our local environment.

It is shown that our proposed method stands out compared to other models, thanks to using the strategy of multi-view projections which can acquire comprehensive and accurate features. Though our overall performance is comparable to that of REMAN, our approach offers enhanced robustness and adaptability in handling intricate scenarios, particularly in addressing point clouds with holes. Meanwhile, it is worth noting that the running time comparison on the ScanRefer dataset shows that our method surpasses most of the SOTA methods, highlighting the effectiveness of our approach for the 3D dense captioning task. Notably, TransCap [26] has lower running time than ours because it uses only the point cloud as input without considering additional 2D inputs. As shown in Fig. 8, the generated captions of objects in eight cases with point clouds containing holes are visualized to verify the robustness and advantages of our method. The eight cases cover various indoor scenarios, which respectively include dining room, recreation room, meeting room, bedroom, living room, printing room, meeting rest area and kitchen. Compared to other SOTA works including Scan2Cap [6], TransCap [26], MORE [27], REMAN [28], it is evident that our method produces the most informative and realistic descriptions, closely matching the ground truth. Our method is excellent to capture fine-grained object features and the relationships between objects within these various complex scenes containing holes. These experimental comparisons verify the novelty of our method in 3D dense caption, particularly for the point clouds with holes. During the generation of captions, the description of the category of an object only requires a subset of key points from the point cloud. However, the description of the relationships between objects requires considering a larger region of points. That is the reason about the holes would cause captions with correct category but wrong relations.

Table 2 presents the quantitative comparison results on the Nr3D dataset. Most papers of the comparison methods did not provide relevant experimental results on Nr3D, therefore, this table only summarizes the reproduced results. The results in Table 2 show that our model outperforms all other methods.

As illustrated in Tables 1 and 2, our model outperforms most of the other models in terms of each metric on the ScanRefer dataset. This observation underscores the robustness and suitability of our model for the task, excelling in different aspects of semantic representation and linguistic expression.

Table 3
Ablation Study for proposed modules.

Method	B-4	C	R	M
Baseline	39.79	63.63	62.31	28.84
Baseline+MFEM	41.96	66.98	63.28	29.09
Baseline+mixG	41.68	67.62	63.32	29.28
Baseline+Transformer	40.94	68.95	62.48	29.19
Baseline+MFEM+mixG	43.21	69.28	63.67	29.30
Baseline+MFEM+Transformer	43.08	70.03	63.60	29.31
Baseline+mixG+Transformer	42.31	69.84	64.22	29.53
Baseline+MFEM+mixG+Transformer	**44.00**	**70.92**	**65.28**	**29.8**

4.4. Ablation study

4.4.1. Ablation for proposed modules

In this section, the ablation study is carried out on the ScanRefer dataset, and the results are presented in Table 3. The metrics for the baseline model are displayed in the first row, where none of the three modules proposed in this paper (MFEM, mixG, Transformer) are used. The second row reveals an increase of 2.17, 3.35, 0.97, and 0.25 in B-4, C, R, and M, respectively, upon employing the MFEM, suggesting

[3] Note that TransCap is X-Trans2Cap [26] without teacher–student design in [54].

Fig. 7. Qualitative comparisons on ScanRefer dataset. The green bounding box in the point cloud represents the object that needs to be described. The blue and brown text in the description indicate the correct and wrong description respectively.

Table 4
Ablation Study for mixG.

Config	Layers of KNN	Layers of DBSCAN	B-4	C	R	M
–	2	0	39.79	63.63	62.31	28.84
–	0	2	40.79	65.38	63.25	29.04
serial	1	1	41.11	66.32	**63.40**	29.14
serial	2	2	41.37	65.86	63.19	29.07
parallel	1	1	**41.68**	**67.62**	63.32	**29.28**
parallel	2	2	41.35	66.76	63.28	29.19
parallel	1	2	41.43	67.02	63.29	29.24
parallel	2	1	41.37	66.92	63.26	29.22

that the multi-view CLIP feature further augments the feature representation of point clouds. As indicated in the third row, when the feature interaction module (mixG) based on mixed graph convolution is added to the baseline model, the B-4, C, R, and M of the model are improved by 1.89, 3.99, 1.01, and 0.44 respectively. This indicates that merely generating a graph structure using KNN is insufficient, and the clustering effect of DBSCAN complements the original graph structure. When only Transformer is used (row 4), the four indexes of the model are improved by 1.15, 5.32, 0.17, and 0.35 respectively. Compared with the baseline model, the usage of Transformer has greatly improved the metric C, which shows that the results of Transformer have stronger linguistic expressiveness than the traditional recurrent neural network. When different modules are paired in combinations (rows 5–7), all metrics improve to varying degrees, and the model achieves the best performance when all three modules are used simultaneously (row 8).

4.4.2. Ablation for mixg

In this section, we explore the impact of various graph convolution modules, different numbers of layers within modules, as well as serial and parallel structures on performance. The results are presented in Table 4. The experiment is divided into three parts. The first experiment investigates the performance when only one graph convolution network is used (*i.e.,* either KNN or DBSCAN graph convolution). By comparing rows 1 and 2 in Table 4, it is observed that the effect of using DBSCAN graph convolution alone surpasses that of using KNN graph convolution alone. This shows that the graph structure of DBSCAN clustering is more in line with the natural scene than the KNN graph structure. The second experiment uses the serial structure of KNN and DBSCAN mixed graph network with different numbers of graph convolution layers. When the number of graph convolution layers of KNN and DBSCAN graphs is both 1 (row 3), the performance is better than that of the model with 2 layers (row 4). The serial structure in row 4 can be regarded as executing graph convolution four times, but too many layers do not aid in the model's performance. When configuring the two modules into a parallel structure, we first conduct comparative experiments between the parallel and serial structures, keeping the number of layers for KNN and DBSCAN identical. Regardless of whether the number of layers is set to 1 or 2 (rows 5–6), the performance of the parallel structure consistently surpasses that of the serial structure. Within the parallel structure, we further conduct control variable experiments to determine the optimal number of layers for each module (rows 5–8). Although the differences in results across various metrics are minimal, it is evident that, overall, the best performance is achieved when both of the KNN and DBSCAN modules are set to 1 layer. This paper finally adopts a parallel structure with the number of graph convolution layers set to 1, thus integrating the advantages of KNN and DBSCAN.

4.4.3. Ablation for various features

In this section, various point features are utilized to enhance the point cloud data, and the ablation experiment results are presented in Table 5. Here, XYZ refers to the use of point cloud coordinates, NORMAL refers to the use of point cloud normal vectors, RGB refers to

Fig. 8. Qualitative comparison between our method and the SOTA works including Scan2Cap [6], TransCap [26], MORE [27], REMAN [28]) on point clouds containing holes, which are marked with red circles.

the use of point cloud colors, and MV refers to the use of planar multi-view scanning features based on ENET. It is observed that performance is as its worst when only point cloud coordinates are used because only coordinates cannot describe the color and texture characteristics of objects. The model performs best when XYZ+NORMAL+MV is used, which is also the scheme employed in the baseline model Scan2Cap. In this paper, the CLIP feature is added on this basis, which further improves the metric performance of the model.

4.4.4. *Ablation for different methods for cropping point cloud*

As presented in Section 3.1.2, there are two ways to crop the input point cloud to obtain a single object's point cloud. One is to crop the point cloud according to the bounding box of the object detection (Vote), and the other is based on the results of semantic segmentation (Instance). The comparison results of the above schemes are shown in Table 6. From the comparison between row 2, row 4, and row 1, it can be found that the results of clipping by bounding box is obviously

Table 5
Ablation Study for various features.

Point features	B-4	C	R	M
XYZ	37.68	60.17	60.19	27.91
XYZ+RGB	40.29	64.10	63.52	28.76
XYZ+NORMAL+RGB	42.20	67.25	64.05	29.49
XYZ+NORMAL+MV	42.01	69.84	64.22	29.53
XYZ+NORMAL+MV+CLIP	**44.00**	**70.92**	**65.28**	**29.81**

Table 6
Ablation Study for different approaches to crop a single object's point cloud.

Objects features	CLIP	B-4	C	R	M
VoteNet w/ Vote	No	42.31	69.84	64.22	29.53
VoteNet w/ Vote	Yes	41.9	68.89	63.19	29.49
PointNet++ w/ Instance	No	41.38	69.99	62.84	29.3
PointNet++ w/ Instance	Yes	**44**	**70.92**	**65.28**	**29.81**

worse than that of semantic segmentation. This may be attributed to the fact that the point cloud cropped by the bounding box may include surrounding objects, which is not the case with semantic segmentation. Since the model in this paper uses the result of semantic segmentation as the input for CLIP, in order to demonstrate that the final promotion is not only led by the usage of semantic segmentation results, this section replaces the CLIP features in the complete model with the semantic segmentation features encoded by PointNet++ (without using CLIP features in this setting) and combines them with VoteNet features as input for the graph convolution model. The results are shown in the third row of the table. Compared with the first row, the direct usage of semantic segmentation features does not lead to significant improvements in the model. Although the CLIP module in this paper uses the semantic segmentation results (row 4), it does not directly use its point cloud features. This shows that the CLIP model by a multi-view feature extraction mechanism has a good ability to deal with noise, and this ability is not derived from using semantic segmented data. Specific cases will be shown in the visualization experiment (Section 4.5.2).

4.5. Visualization

4.5.1. Visualization for KNN and DBSCAN

In this section, we visualize the graph structures generated by the KNN and DBSCAN methods, as illustrated in Fig. 5. The graph structure produced by KNN is on the left, while DBSCAN's output is on the right. KNN includes all the K nearest objects in the graph structure, even if the lower and farther curtains in Fig. 5 (a1) are far away from the bed, they are still marked as nearby objects around the bed. In Fig. 5(b1), the objects in the room are obviously lower left and upper right, but there is no such difference in the graph structure generated by KNN. On the other hand, the graph structure generated by DBSCAN shows a better analytical ability for the above scenes. In Fig. 5(a2), far objects on the rightmost side and the bottommost side are excluded, and in Fig. 5(b2), another cluster of objects is excluded. That is because DBSCAN is influenced by cluster density and cluster radius. However, different annotators may define different scales for surrounding objects, leading to instances where DBSCAN results indicate clutter (no surrounding objects) while the annotators consider that it is related to the surrounding objects. Therefore, KNN and DBSCAN complement each other, and combining them can improve the performance of the model.

4.5.2. Visualization of different methods for cropping point cloud

As illustrated in Fig. 6, the point cloud cropped by the bounding box has a large noise when there are objects at a close distance around it. When the point cloud is cropped by the bounding box of the table (see Fig. 6(c)), if there are many chairs under the table, they will also be cropped, which makes it difficult to recognize that this is a table. The result of semantic segmentation (Fig. 6(d)) can clearly identify the shape of the table because it does not include other objects. Even if some points in the angular parts are offset, it does not affect the overall recognition. This also explains why using the point cloud cropped by the bounding box to extract CLIP features will not improve the performance because the CLIP model is mainly used for the classification task of a single object. If the overlap and occlusion are very serious, using CLIP to extract features may not show good performance. Compared with object detection, semantic segmentation can naturally adapt to the limitations of the CLIP model, and even if a part of the object is missing, it can still be successfully recognized from some angles.

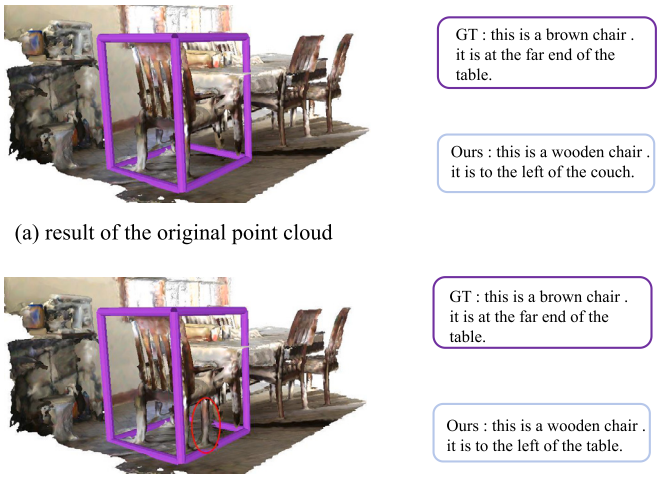


Fig. 9. The results of manually edited low quality point cloud.

4.5.3. Visualization for cases

Fig. 7 shows the visualization results of some cases of our model. In Fig. 7(a), our model successfully identifies "shelf", while the Scan2Cap model considers it as "cabinet". As illustrated in Fig. 7(b), color attributes are described in addition to Scan2Cap's results. Our model gives a more accurate description in Fig. 7(c). In Fig. 7(d), Scan2Cap incorrectly identified the object as "towel". The above results are attributed to the good representations extracted by CLIP and the decoding ability of Transformer. In Fig. 7(e) and 7(f), our model successfully describes the relationship between door and freezer, couch and table respectively, while Scan2Cap describes irrelevant objects. This is the benefits caused by the feature interaction module based on mixed graph convolution.

4.5.4. Visualization for low-quality scene

We test our model on the manually edited low-quality scene of point cloud. As shown in Fig. 9, we added noise to a point cloud from ScannetV2 by using Meshlab software.

Comparing Fig. 9(a) and Fig. 9(b), we added a chair leg as noise. Although there have been few changes in the description, the meaning of the caption is the same as the original one. The results demonstrate the robustness of our model in dealing with scenes with small changes such as noises. Note that low-quality point clouds like the one in Fig. 9 are not used in the training process.

4.5.5. Visualization for failure cases

Fig. 10 shows that our model gives some wrong predictions in some scenarios. In Fig. 10(a), the relationship between the bed and the window is described as "left", but in the real annotation, their relationship is "under". This is because the model does not clearly define this relationship. In Fig. 10(b), the refrigerator is mistakenly

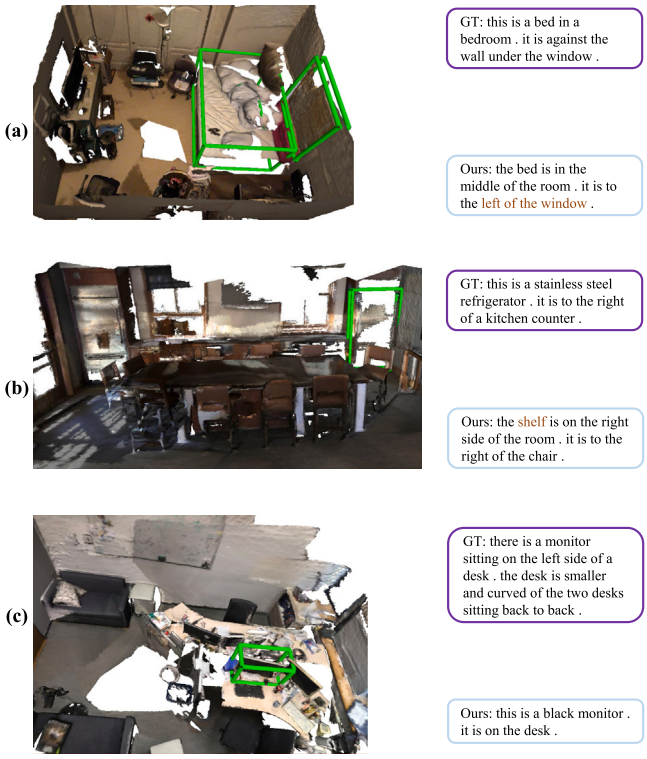


Fig. 10. The failure cases of our model in some scenes.

recognized as a cabinet because the quality of the scanned scene is too low, and the refrigerator only has a small area left, which is difficult even for humans to identify. In Fig. 10(c), although the correct relationship is described, the generated description is still too simple and not rich enough compared with the real manual annotation.

5. Conclusion and future works

In this paper, a multi-level mix encoding model is proposed for the task of 3D dense captioning. On the basis of VoteNet features, our model projects objects from the original point cloud with multi-view transformation and uses the CLIP model to extract multi-view features, which helps to alleviate the problem of gaps and holes in the point cloud and makes the object features more accurate. In the construction of graph relationships, our model uses KNN and DBSCAN to build graphs, which effectively extract the relationships from each object to the surrounding objects. Finally, our model replaces the recurrent neural network in the caption generation module with Transformer, which enhances the semantic generation ability. We present experiment results to verify the effectiveness of the proposed model on ScanRefer and Nr3D datasets. Although our method achieves good results, there is still room for improvement. In future work, we will try to define the spatial relationship in detail and study the diversity of models to generate more complex and detailed descriptions. Furthermore, considering the widespread use of large language models (LLMs), exploring the enhancement of our method's dense captioning performance using LLMs presents a promising direction for future research.

CRediT authorship contribution statement

Zijing Ma: Writing – review & editing, Validation, Resources, Investigation. **Zhi Yang:** Writing – original draft, Software, Methodology, Conceptualization. **Aihua Mao:** Writing – review & editing, Writing – original draft, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Shuyi Wen:** Writing – review & editing, Validation, Investigation. **Ran Yi:** Writing – original draft, Supervision, Resources, Formal analysis, Data curation. **Yongjin Liu:** Writing – original draft, Resources, Methodology, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515011573, 2024A1515012791, Guangzhou Municipal Science and Technology Program key projects under Grant 2023B01J1001, National Natural Science Foundation of China under grant 62302297, Shanghai Sailing Program under grant 22YF1420300, Young Elite Scientists Sponsorship Program by CAST under grant 2022QNRC001, Beijing Hospitals Authority Clinical Medicine Development of special funding support under Grant ZLRK202330.

Data availability

No data was used for the research described in the article.

References

[1] Wu J, Chen T, Wu H, Yang Z, Luo G, Lin L. Fine-grained image captioning with global-local discriminative objective. IEEE Trans Multimed 2021;23:2413–27. http://dx.doi.org/10.1109/TMM.2020.3011317.
[2] Wang D, Hu Z, Zhou Y, Hong R, Wang M. A text-guided generation and refinement model for image captioning. IEEE Trans Multimed 2023;25:2966–77. http://dx.doi.org/10.1109/TMM.2022.3154149.
[3] Aafaq N, Mian A, Akhtar N, Liu W, Shah M. Dense video captioning with early linguistic information fusion. IEEE Trans Multimed 2023;25:2309–22. http://dx.doi.org/10.1109/TMM.2022.3146005.
[4] Zhang Z, Xu D, Ouyang W, Zhou L. Dense video captioning using graph-based sentence summarization. IEEE Trans Multimed 2021;23:1799–810. http://dx.doi.org/10.1109/TMM.2020.3003592.
[5] Antol S, Agrawal A, Lu J, Mitchell M, Batra D, Lawrence Zitnick C, et al. Vqa: Visual question answering. In: Proceedings of the IEEE international conference on computer vision. 2015, p. 2425–33.
[6] Chen Z, Gholami A, Nießner M, Chang AX. Scan2cap: Context-aware dense captioning in rgb-d scans. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, p. 3193–203.
[7] Ye S, Chen D, Han S, Liao J. 3D question answering. 2021, arXiv preprint arXiv:2112.08359.
[8] Chen DZ, Chang AX, Nießner M. Scanrefer: 3d object localization in rgb-d scans using natural language. In: European conference on computer vision. 2020, p. 202–21.
[9] Dudani SA. The distance-weighted k-nearest-neighbor rule. IEEE Trans Syst Man Cybern 1976;(4):325–7.
[10] Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, et al. Learning transferable visual models from natural language supervision. In: International conference on machine learning. 2021, p. 8748–63.
[11] Qi CR, Litany O, He K, Guibas LJ. Deep hough voting for 3d object detection in point clouds. In: Proceedings of the IEEE/CVF international conference on computer vision. 2019, p. 9277–86.
[12] Ester M, Kriegel H-P, Sander J, Xu X. Density-based spatial clustering of applications with noise. In: Int. conf. knowledge discovery and data mining, vol. 240, no. 6. 1996.
[13] Achlioptas P, Abdelreheem A, Xia F, Elhoseiny M, Guibas L. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In: European conference on computer vision. 2020, p. 422–40.
[14] Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhudinov R, et al. Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. 2015, p. 2048–57.
[15] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst 2017;30.

[16] Anderson P, He X, Buehler C, Teney D, Johnson M, Gould S, et al. Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 6077–86.

[17] Cornia M, Stefanini M, Baraldi L, Cucchiara R. Meshed-memory transformer for image captioning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, p. 10578–87.

[18] Deng C, Ding N, Tan M, Wu Q. Length-controllable image captioning. In: Computer vision–ECCV 2020: 16th European conference, glasgow, UK, August 23–28, 2020, proceedings, part XIII 16. 2020, p. 712–29.

[19] Zhang Z, Wu Q, Wang Y, Chen F. Exploring pairwise relationships adaptively from linguistic context in image captioning. IEEE Trans Multimed 2022;24:3101–13. http://dx.doi.org/10.1109/TMM.2021.3093725.

[20] Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). 2019, p. 4171–86.

[21] Lu J, Batra D, Parikh D, Lee S. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. Adv Neural Inf Process Syst 2019;32.

[22] Su W, Zhu X, Cao Y, Li B, Lu L, Wei F, et al. VL-BERT: Pre-training of generic visual-linguistic representations. In: International conference on learning representations. 2019.

[23] Song P, Guo D, Cheng J, Wang M. Contextual attention network for emotional video captioning. IEEE Trans Multimed 2023;25:1858–67. http://dx.doi.org/10.1109/TMM.2022.3183402.

[24] Qi CR, Yi L, Su H, Guibas LJ. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Adv Neural Inf Process Syst 2017;30.

[25] Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. In: Proceedings of the 34th international conference on machine learning, vol. 70. 2017, p. 1263–72.

[26] Yuan Z, Yan X, Liao Y, Guo Y, Li G, Li Z, et al. X-Trans2Cap: Cross-modal knowledge transfer using transformer for 3D dense captioning. 2022, arXiv preprint arXiv:2203.00843.

[27] Jiao Y, Chen S, Jie Z, Chen J, Ma L, Jiang Y-G. More: Multi-order relation mining for dense captioning in 3d scenes. In: Computer vision–ECCV 2022: 17th European conference, tel aviv, Israel, October 23–27, 2022, proceedings, part XXXV. 2022, p. 528–45.

[28] Mao A, Yang Z, Chen W, Yi R, Liu Y-j. Complete 3D relationships extraction modality alignment network for 3D dense captioning. IEEE Trans Vis Comput Graphics 2023.

[29] Qi CR, Liu W, Wu C, Su H, Guibas LJ. Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 918–27.

[30] Xu D, Anguelov D, Jain A. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 244–53.

[31] Chen R, Han S, Xu J, Su H. Point-based multi-view stereo network. In: Proceedings of the IEEE/CVF international conference on computer vision. 2019, p. 1538–47.

[32] Sindagi VA, Zhou Y, Tuzel O. Mvx-net: Multimodal voxelnet for 3d object detection. In: 2019 International conference on robotics and automation. 2019, p. 7276–82.

[33] Song S, Xiao J. Sliding shapes for 3d object detection in depth images. In: Computer vision–ECCV 2014: 13th European conference, Zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13. 2014, p. 634–51.

[34] Shuran S, Jianxiong X. Deep sliding shapes for amodal 3d object detection in rgb-d images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 808–16.

[35] Li X, Shi B, Hou Y, Wu X, Ma T, Li Y, et al. Homogeneous multi-modal feature fusion and interaction for 3D object detection. In: Computer vision–ECCV 2022: 17th European conference, tel aviv, Israel, October 23–27, 2022, proceedings, part XXXVIII. 2022, p. 691–707.

[36] Li L, Gan Z, Cheng Y, Liu J. Relation-aware graph attention network for visual question answering. In: Proceedings of the IEEE/CVF international conference on computer vision. 2019, p. 10313–22.

[37] Huang Q, Wei J, Cai Y, Zheng C, Chen J, Leung H-f, et al. Aligned dual channel graph convolutional network for visual question answering. In: Proceedings of the 58th annual meeting of the association for computational linguistics. 2020, p. 7166–76.

[38] Chen H, Huang Y, Takamura H, Nakayama H. Commonsense knowledge aware concept selection for diverse and informative visual storytelling. In: Proceedings of the AAAI conference on artificial intelligence, vol. 35, no. 2. 2021, p. 999–1008.

[39] Vo DM, Luong Q-A, Sugimoto A, Nakayama H. A-CAP: Anticipation captioning with commonsense knowledge. 2023, arXiv preprint arXiv:2304.06602.

[40] Speer R, Chin J, Havasi C. Conceptnet 5.5: An open multilingual graph of general knowledge. In: Proceedings of the AAAI conference on artificial intelligence, vol. 31, no. 1. 2017.

[41] Yang X, Peng J, Wang Z, Xu H, Ye Q, Li C, et al. Transforming visual scene graphs to image captions. 2023, arXiv preprint arXiv:2305.02177.

[42] Battaglia PW, Hamrick JB, Bapst V, Sanchez-Gonzalez A, Zambaldi V, Malinowski M, et al. Relational inductive biases, deep learning, and graph networks. 2018, arXiv preprint arXiv:1806.01261.

[43] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. 2014, arXiv preprint arXiv:1406.1078.

[44] Zhang R, Guo Z, Zhang W, Li K, Miao X, Cui B, et al. Pointclip: Point cloud understanding by clip. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, p. 8552–62.

[45] Hartigan JA, Wong MA. Algorithm AS 136: A k-means clustering algorithm. J Royal Statist Soc Series C (Appl Statist) 1979;28(1):100–8.

[46] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.

[47] Dai A, Chang AX, Savva M, Halber M, Funkhouser T, Nießner M. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 5828–39.

[48] Paszke A, Chaurasia A, Kim S, Culurciello E. Enet: A deep neural network architecture for real-time semantic segmentation. 2016, arXiv preprint arXiv:1606.02147.

[49] Papineni K, Roukos S, Ward T, Zhu W-J. Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the association for computational linguistics. 2002, p. 311–8.

[50] Vedantam R, Lawrence Zitnick C, Parikh D. Cider: Consensus-based image description evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, p. 4566–75.

[51] Lin C-Y, Hovy E. Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics. 2003, p. 150–7.

[52] Banerjee S, Lavie A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization. 2005, p. 65–72.

[53] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In: International conference on learning representations.

[54] Hinton G, Vinyals O, Dean J, et al. Distilling the knowledge in a neural network. 2015, arXiv preprint arXiv:1503.02531 2(7).