Explore Contextual Information for 3D Scene Graph Generation

Yuanyuan Liu, Chengjiang Long, Zhaoxuan Zhang, Bokai Liu, Qiang Zhang, Baocai Yin, and Xin Yang*

Abstract—3D scene graph generation (SGG) has been of high interest in computer vision. Although the accuracy of 3D SGG on coarse classification and single relation label has been gradually improved, the performance of existing works is still far from being perfect for fine-grained and multi-label situations. In this paper, we propose a framework fully exploring contextual information for the 3D SGG task, which attempts to satisfy the requirements of fine-grained entity class, multiple relation labels, and high accuracy simultaneously. Our proposed approach is composed of a Graph Feature Extraction module and a Graph Contextual Reasoning module, achieving appropriate information-redundancy feature extraction, structured organization, and hierarchical inferring. Our approach achieves superior or competitive performance over previous methods on the 3DSSG dataset, especially on the relationship prediction sub-task.

Index Terms—scene understanding, context exploration, graph skeleton, scene graph generation.

1 INTRODUCTION

THE Scene Graph (SG) organizes the content of a scene into a graph-based representation, which encodes objects as nodes, connected via pairwise relationships as edges, thus representing complex scene knowledge as a compact graphical structure. Scene graph generation (SGG) enables adequate perception and comprehensive understanding of scenes, especially for 3D real-world scenes, and therefore is beneficial to widespread applications, *e.g.*, robot navigation [1], task planning [2], and scene modification and manipulation [3], [4].

One of the main challenges that current models face is the existence of multi-granularity of objects and multi-relationships between objects, as illustrated in Fig. 1(b). The multi-granularity of objects depends not only on the diverse appearances of objects, but also on the surrounding environment of objects. For example, when chairs surround a `table`, it most likely becomes a `dining table`. On the other hand, the interaction between two objects is hard to define with a single relationship, which leads to multi-relationships between objects. For example, there can be both a spatial relationship `left` and a semantic relationship `same as` between chairs.

A common approach to address this problem is to define effective visual patterns and collect sufficient contextual information before classifying entities. While a series of SGG methods [5], [6], [7], [8], [9], [10] have achieved great success in 2D scenes by using the union region as the visual pattern and message passing between all the entities, we
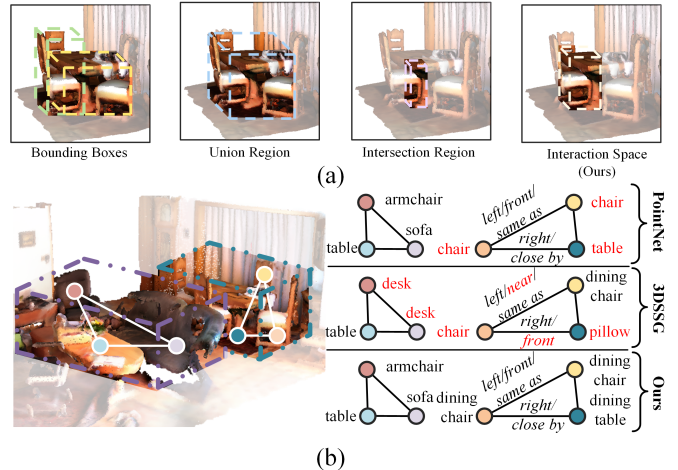

Fig. 1. **(a) Different visual patterns for relationship representation.** Corresponding modeling regions for the relationship between a table and a chair. **(b) Multi-granularity of objects and multi-relationships between objects.** Without the message propagation phase, PointNet cannot infer `table` as `dining table`. The full-graph-based message propagation blurs the discrimination of the delicate features, which leads 3DSSG to perform worse than PointNet. With structured organization rules and hierarchical object labels for message propagation, our method accurately predicts fine-grained entities and multiple relation labels. The word marked in red color denotes the wrong prediction.

cannot apply these 2D methods to 3D scenes directly. The reason is that, compared with 2D image pixels, 3D point clouds are highly unstructured and irregular [11], and due to the inevitable noise in the data, it is more difficult to perform appropriate information extraction and fusion. In a 3D scene, using union regions as visual patterns will lead to noise multiplication. The union region refers to the union of the object bounding boxes, so every time a relationship is modeled, the object region is remodeled (see Fig. 1(a)). The redundant information generated by the repeated modeling of the identical region confuses the learning ability of the

• *Yuanyuan Liu, Zhaoxuan Zhang, Bokai Liu, Qiang Zhang, Baocai Yin and Xin Yang are with the Department of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, 116024, China. E-mail: Liuyy990415@gmail.com; zhangzx@mail.dlut.edu.cn; liubokai2021@mail.dlut.edu.cn; zhangq@dlut.edu.cn; ybc@dlut.edu.cn and xinyang@dlut.edu.cn.*
• *Chengjiang Long is currently a Research Scientist at Meta Reality Labs, Burlingame, CA, 94010, USA. E-mail: clong1@fb.com.*
• ** Xin Yang (xinyang@dlut.edu.cn) is the corresponding authors.*
• *Code is available at: https://github.com/YYLiuDLUT/3D_SCENEGRAPH*

algorithm to a large extent.

In the message passing stage, the features extracted by the visual pattern are used as initial features, which will be propagated indiscriminately among all possible neighbors. The unreasonable information fusion process leads to the explosive propagation of noise. In theory, with a certain depth, all nodes' representations will converge to a stationary point, leading to feature consistency [12], [13]. Resulting in the prediction performance of the fused features being even lower than the original features (see Fig. 1(b)). These two factors lead to the fact that although the recent 3D SGG works [14], [15] take extra class-agnostic instance information as input, they still cannot obtain satisfactory results with both fine-grained classification and multiple relation labels.

Based on the observations of humans' scene understanding process. People will adaptively adjust the information redundancy [16] when entering an unknown environment by extracting efficient visual patterns. These visual patterns, as well as environmental contexts, are then structurally organized and hierarchically inferred in the prefrontal cortex (PFC) [17]. So scene knowledge can be rapidly formed in the brain through optimal processing of small amounts of information.

Inspired by this, we propose a framework for exploring contextual information in 3D SGG, as shown in Fig. 2. The Graph Feature Extraction module extracts entity and relation features with appropriate information redundancy. And the Graph Contextual Reasoning module structurally organizes and hierarchically infers these features, as shown in Fig 4. As a result, we can generate an accurate scene graph with fine-grained entity classes and multiple relation labels.

In the Graph Feature Extraction module, we use the intersection space of the object bounding boxes to replace the union region to reduce the repeated modeling of the identical area. It has been confirmed that the intersection region of the 2D object bounding box proposed by [18] has a better relationship representation effect than the union region. However, applying it directly to the 3D bounding boxes usually yields inaccurate and unreasonable predictions. To extract rich contextual information between the 3D objects, we present the *"interaction space"* (see Fig. 3) together with bounding box position information as a new visual pattern for 3D SSG task. In particular, while we reduce the redundancy by adjusting the union region as intersection space, information about objects, relative relationships, and surrounding contextual information is missing in relation features due to the variation of the receptive field. While keeping the information redundancy constant, we recover the over-removed information by region expansion and position information encoding to ensure that the extracted features can cover the underlying properties of the relationship.

In the Graph Contextual Reasoning module, we take a multi-task learning approach by introducing the Graph Skeleton Learning (GSL) block and the Hierarchy Object Learning (HOL) block on top of the Message Passing block. We use graph skeleton information to represent the correlation between object pairs, where graph skeleton information represents the ground truth of the scene graph with the label information removed. The GSL block reconstructs the fully connected graph into an edge-weighted skeleton graph (see Fig. 5), and captures the contextual information by propagating node messages in the graph. By jointly training the GSL and message passing blocks, the joint action between the two blocks is strengthened, preventing the GSL blocks from converging independently in static space. GSL utilizes the skeleton to incorporate the structured organization process into the contextual information fusion stage, reducing ineffective information exchange and thus preventing noise propagation.

The GSL block can be regarded as a binary classification task with a simple form and high accuracy. Yet, due to the imbalance of relation and non-relation data among nodes, GSL may block the communication between most nodes. This requires us to use limited contextual information for efficient reasoning to achieve fine-grained classification of entities. Therefore, we design the HOL module, which builds a two-level hierarchical tree using coarse-grained and fine-grained object labels (see in Fig. 6). Coarse-grained object labels supervise initial features that contain only attribute information, and fine-grained object labels supervise features for context fusion. Therefore, HOL reduces the task's difficulty by hierarchically decomposing the object classification task. As demonstrated in our experimental results, the proposed SGG approach significantly outperforms existing state-of-the-art methods quantitatively and qualitatively.

In summary, our contributions are three-fold as follows:

- We utilize a new visual pattern based on interaction region and bounding box position information to extract relation features with appropriate redundancy.
- We propose a multi-task learning strategy in our Graph Contextual Reasoning module, which structurally organizes and hierarchically infers the information, predicting multiple labels from similar features. We are the first method to use the graph skeleton information as supervision information.
- Extensive experiments have demonstrated that our 3D SGG has achieved significant performance improvement to state-of-the-art methods on various scene graph benchmarks, especially on relationship prediction, and even better overcoming the long-tail effect.

2 RELATED WORK

2D and 3D SGG Tremendous 2D SGG progress [19], [20] has been made since [21] firstly mentioned the scene graph. In 3D, SGs have only recently gained more popularity [14], [15], [22], [23], [24] thanks to the introduction of the 3DSSG dataset [14], which contains semantically rich scene graphs of 3D scenes. [14] also proposes an end-to-end network that employs a graph convolutional network (GCN) [25] to handle the message passing stage. SGGPoint [15] builds two associated twin interaction mechanisms between nodes and edges to effectively bridge perception and reasoning. However, SGGPoint only retains 27 object classes and 16 relation classes, and converts multi-label relations into single relation. Hence, it only performs well on the SGG task of coarse classification and single relation label. [24]
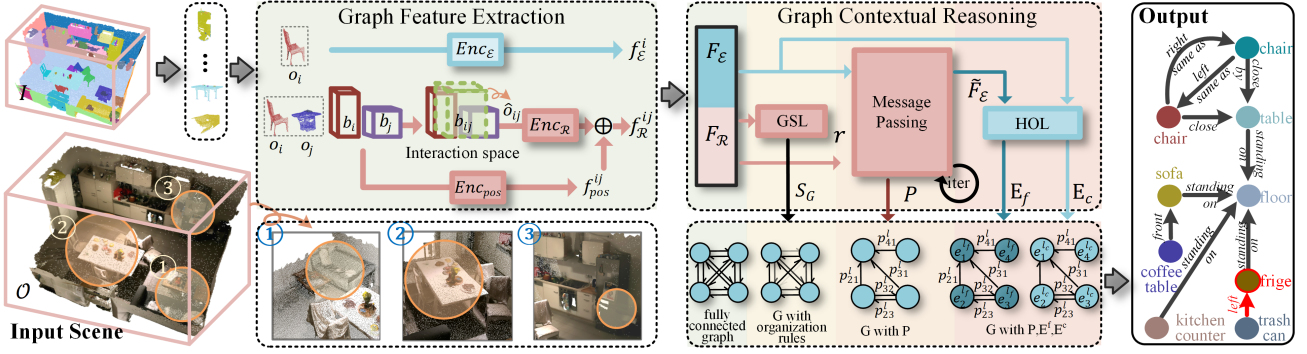
Fig. 2. **Overview of our proposed framework.** We only show partial SGG results corresponding to the circularly marked area of the input scene. Our Graph Contextual Reasoning module introduces a Graph Skeleton Learning (GSL) block and a Hierarchy Object Learning (HOL) block on top of the Message Passing block. We differentiate each block with colors and list the evolution of the scene graph as thumbnails below each block. The internal implementation details are shown in Fig. 4. The generated results demonstrate that our method enables the detection of both `left` (spatial) and `same as` (semantic) relationships between two chairs. And it achieves accurate predictions for both coarse and fine object labels, `table`, and `coffee table` that belong to the same class but with different granularity. The word marked in red color denotes the wrong prediction.

advocates using graph auto-encoder to automatically extract class-dependent representations and topological patterns as prior knowledge to enhance the accuracy of relationship predictions. Unlike other methods [26], [27] that use prior knowledge to guide the contextual fusion stage, perception and prior are naively treated as separate components. They are trained separately from different inputs (from images, triples, or label embeddings), and their predictions are usually fused in a probability space. As [28], we train multiple tasks in parallel to achieve better fusion. Our work successfully builds discriminative features based on feature extraction methods and environmental context fusion, enabling us to map between similar features and multiple labels.

Relationship Feature Extraction [14] uses the same union region as the 2D SGG methods to represent relationship features. [15], [24] use feature engineering [29] and a concatenation scheme to generate relation features from entity features. However, both of them suffer from information redundancy. This redundancy is mainly due to repeated modeling of the same region. Noise in the information is replicated exponentially, reducing features' comprehensibility, ambiguity, and fault tolerance [16]. Most 3D methods try to minimize the noise by reducing the receptive field or locating regions of interest. [30] attempts point set filtering on point clouds to reconstruct noise-free point sets from corresponding noisy inputs. [31] learns and predicts interest points in 3D point clouds using multiple feature descriptors. Therefore, we refer to the intersection region [18] and design a new visual pattern, "interaction space", which has a smaller receptive field and focuses more on the interaction area between objects than the object area. Combined with position feature calculation, our visual pattern can strike a good balance between information content and redundancy.

Graph Contextual Reasoning Context modeling strategies [32], [33], [34], [35], [36] in SGG are mainly used to learn discriminative representation for node and edge prediction, either by designing graph structures or leveraging scene context via various message propagation mechanisms [37], [38], [39]. The most popular graph structure is the fully-connected graph [40], [41], [42]. Recent works [43], [44],

[45], [46] tried to model the context based on the sparse graph structures, using either downstream tasks (*e.g.*VQA) or trimming functions to cut unessential subject-object pairs. These approaches require additional supervised data [47] to realize the downstream tasks, and the loss of information caused by pruning behavior is not reversible. On the other hand, message propagation mechanisms aim to aggregate the contextual information between entities [48] and predicates [49], [50], [51]. Our work borrows ideas from both sides, which can be seen as incorporating the graph structure into the message propagation process. And like [52], we design a hierarchy inferring block for object features, thereby reducing the difficulty of object classification and inference tasks through hierarchical decomposition.

3 APPROACH

3.1 Overview

Given a scene point cloud $\mathcal{O}$ labeled with its class-agnostic instance segmentation $\mathcal{I}$ as input, we first split $\mathcal{O}$ into several object point clouds $\{o_1, o_2, ..., o_n\}$. Our goal is to predict semantic labels for each object and relations among them. To achieve this, we feed all these objects into Graph Feature Extraction module (Sec. 3.2) to obtain entity features $F_\mathcal{E}$ and relation features $F_\mathcal{R}$, which form a fully connected feature graph. Then, these two feature sets $F_\mathcal{E}$ and $F_\mathcal{R}$ will be structurally organized and hierarchically inferred by Graph Contextual Reasoning module (Sec. 3.3), which finally outputs a scene graph denoted as $\mathcal{G} = (E, P)$. E is a semantic labeled entity set as $E = \{e_1^l, e_2^l, ..., e_n^l\}$, and P is a relation labeled predicate set as $P = \{p_{12}^l, p_{13}^l, ..., p_{ij}^l, ..., p_{n,n-1}^l | i \neq j\}$. For each relation p_{ij}^l, it is represented as a binary m-dimension vector where m is the number of all relation labels. If k_{th} dimension's value is 1 means that p_{ij}^l contains k_{th} relation (All-zero vector represents no relationship between e_i and e_j). To be noted that the predicted relations are with multiple relationship labels, so there may be not only one dimension in p_{ij}^l with a value of 1.

3.2 Graph Feature Extraction

Given a scene point cloud $\mathcal{O}$, we extract the point set of each instance i separately, labeled with its class-agnostic instance
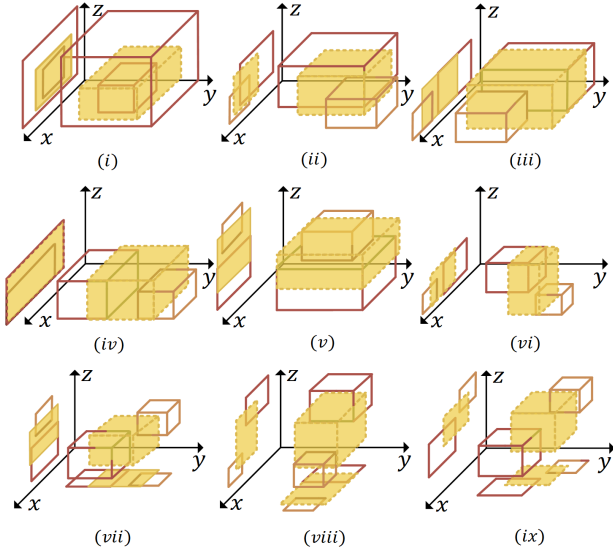
Fig. 3. **Eight cases of interaction space.** Intersectant-(i) Inclusive, Intersectant-(ii) Overlap, (iii) X-direction disjoint , (iv) Y-direction disjoint, (v)Z-direction disjoint, (vi) XY-direction disjoint, (vii) YZ-direction disjoint, ($viii$) XZ-direction disjoint and (ix) XYZ-direction disjoint.

segmentation $\mathcal{I}$, same as [14], [18], [24]. For the object point cloud set $\{o_1, o_2, ..., o_n\}$, we first extract the object feature $f_{\mathcal{E}}^i$ by a PointNet [54] based feature extractor $Enc_{\mathcal{E}}(\cdot)$ for each object o_i, then traverse the following processes for any two objects: 1) calculate the interaction space B_{ij} between bounding box b_i and b_j and encode b_i, b_j to a position feature vector f_{pos}^{ij}; 2) extract the features of the points inside B_{ij} by feature extractor $Enc_{\mathcal{R}}(\cdot)$ and obtain the relation feature $f_{\mathcal{R}}^{ij}$ concatenated with f_{pos}^{ij}. The calculation details of interaction space and position feature are described below.

Interaction Space Calculation Traditional 3D SGG [14], [15], [24] methods represent the relationships that use either union regions or feature engineering. From the perspective of a single pair of objects, these visual patterns contain interaction information, relative position information, surrounding scene information, *etc*, which can cover more contextual information in the relationship features. But looking at the whole scene, the regions of entities are repeatedly modeled by relations, and the relations between several pairs of adjacent objects contain similar or even the same information, resulting in information redundancy. In the subsequent message passing phase, this redundant information will continue to diffuse and grow with iterations, eventually making the features indistinguishable. Therefore, in the feature extraction stage, we design a new visual pattern with reference to the intersection region of [18], which is highly coincident with the 3D points in the Interaction Region (IR) [55] computed using the Interaction Bisector Surfaces (IBS) plane, hence the name Interaction Space (InS). We successfully control the information redundancy by reducing the perceptual regions of relational visual patterns.

To fit the 3D case, we use three auxiliary conditions Eq. 1, Eq. 2 and Eq. 3 to classify the relative position between b_i and b_j into eight cases: intersectant (inclusive, overlap), X-direction disjoint, Y-direction disjoint, Z-direction disjoint, XY-direction disjoint, YZ-direction disjoint, XZ-direction

disjoint and XYZ-direction disjoint, as shown in Fig. 3.

$$|x_c^i - x_c^j| \geq \frac{l^i + l^j}{2}, \tag{1}$$

$$|y_c^i - y_c^j| \geq \frac{w^i + w^j}{2}, \tag{2}$$

$$|z_c^i - z_c^j| \geq \frac{h^i + h^j}{2}, \tag{3}$$

where (x_c^i, y_c^i, z_c^i) denote the center coordinates of bounding box b_i, and l^i, w^i, h^i denote b_i's length, width, and height, respectively. The three auxiliary conditions are mainly used to determine whether the projection planes of b_i and b_j in the x, y, and z directions intersect.

We first initialize the interaction space B_{ij} with the union bounding box of b_i and b_j:

$$B_{ij} = [x_1^{U_{ij}}, y_1^{Uij}, z_1^{U_{ij}}, x_2^{U_{ij}}, y_2^{U_{ij}}, z_2^{U_{ij}}] \tag{4}$$

The lower left corner $(x_1^{U_{ij}}, y_1^{Uij}, z_1^{U_{ij}})$ and the upper right corner $(x_2^{U_{ij}}, y_2^{U_{ij}}, z_2^{U_{ij}})$ are used to represent the union bounding box of b_i and b_j. Judging according to Eq. 1, Eq. 2 and Eq. 3 in sequence, if a certain auxiliary condition is satisfied, we replace the coordinates of B_{ij} in this direction as below:

$$B_{ij} = \begin{cases} x_1^{U_{ij}} \to \min(x_c^i, x_c^j), x_2^{U_{ij}} \to \max(x_c^i, x_c^j), & \text{Eq.1,} \\ y_1^{U_{ij}} \to \min(y_c^i, y_c^j), y_2^{U_{ij}} \to \max(y_c^i, y_c^j), & \text{Eq.2,} \\ z_1^{U_{ij}} \to \min(z_c^i, z_c^j), z_2^{U_{ij}} \to \max(x_c^i, x_c^j), & \text{Eq.3,} \end{cases} \tag{5}$$

For intersectant (inclusive and overlap) case, which no auxiliary condition is met, we first calculate the intersection space of b_i and b_j:

$$\mathbf{B}_{ij}^{InT} = [\max(x_1^i, x_1^j), \max(y_1^i, y_1^j), \max(z_1^i, z_1^j), \\ \min(x_2^i, x_2^j), \min(y_2^i, y_2^j), \min(z_2^i, z_2^j)] \tag{6}$$

We use the coordinates of the lower left corner(x_1^i, y_1^i, z_1^i) and the upper right corner(x_2^i, y_2^i, z_2^i) to represent box b_i. It is worth mentioning that, as for the intersectant case, we perform region expansion to obtain more environmental context. As the overlap areas of the object bounding boxes are often obscured or inaccessible, there are no or few points in these areas, which makes it difficult to extract relationship features. The interaction space after expansion can be expressed as:

$$B_{ij} = [\frac{x_1^{ij} + x_1^{U_{ij}}}{2}, \frac{y_1^{ij} + y_1^{U_{ij}}}{2}, \frac{z_1^{ij} + z_1^{U_{ij}}}{2}, \\ \frac{x_2^{ij} + x_2^{U_{ij}}}{2}, \frac{y_2^{ij} + y_2^{U_{ij}}}{2}, \frac{z_2^{ij} + z_2^{U_{ij}}}{2}] \tag{7}$$

where $\left[x_1^{ij}, y_1^{ij}, z_1^{ij}, x_2^{ij}, y_2^{ij}, z_2^{ij}\right]$ denotes the intersection bounding box $\mathbf{B}_{ij}^{InT}$ of b_i and b_j, calculated by Eq. 6.

Position Feature Calculation The interaction space can effectively reduce information redundancy by narrowing the perception region. However, according to the observation, it can be found that the interaction space is more concerned with the area between objects than the object area. This means that there is a lack of information about objects,
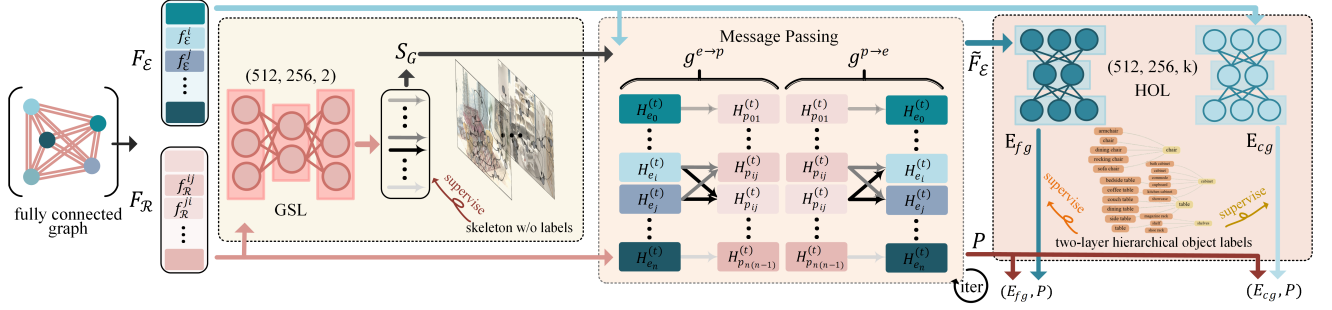
Fig. 4. **Graph Contextual Reasoning Module.** The GSL block is supervised by unlabeled graph skeleton information. Coarse and fine-grained entity labels supervise the HOL block. We show part of the labels in the form of a tree structure, where the word with orange color indicates the fine-grained class categories, and the word with yellow color indicates the coarse-grained NYU40 [53] object categories.

relative relationships, and surrounding contextual information in relationship features. Among them, the object and surrounding contextual information can be supplemented in subsequent message passing stages. But the relative relationships are challenging to learn directly from the features due to the disorder of 3D points. Therefore, we need to additionally encode objects' relative positional relationship, which is of great significance for distinguishing the subject-object relationship between objects [56]. Thus we sort the object 3D bounding box coordinates in a subject-object order as [57]:

$$f_{pos}^{ij} = Enc_{pos}([p_{min}^i, p_{max}^i, p_{min}^j, p_{max}^j]), \quad (8)$$

$$p_{min}^i = \left[\frac{x_{min}^{b_i} - x_{min}^{U_{ij}}}{l^{U_{ij}}}, \frac{y_{min}^{b_i} - y_{min}^{U_{ij}}}{w^{U_{ij}}}, \frac{z_{min}^{b_i} - z_{min}^{U_{ij}}}{h^{U_{ij}}} \right], \quad (9)$$

where i denotes the subject, j the object, $x_{min}^{b_i}$ the minimum value for b_i's coordinates on X-direction, $x_{min}^{U_{ij}}$ the minimum value for union region's coordinates on X-direction. For brevity, we only report the calculation of p_{min}^i, the other items are calculated by exchanging min to max or i to j. Therefore, we obtain a 12-d bounding box position vector and then extract the position feature f_{pos}^{ij} by a fully connected network $Enc_{pos}(\cdot)$. In general, the feature initialization process for fully connected scene graph can be expressed as:

$$F_{\mathcal{E}} = \{f_{\mathcal{E}}^1, f_{\mathcal{E}}^2, ..., f_{\mathcal{E}}^n\} : f_{\mathcal{E}}^i = Enc_{\mathcal{E}}(o_i), \quad (10)$$

$$F_{\mathcal{R}} = \{f_{\mathcal{R}}^{12}, f_{\mathcal{R}}^{13}, ..., f_{\mathcal{R}}^{n,n-1}\} : f_{\mathcal{R}}^{ij} = Enc_{\mathcal{R}}(\hat{o}^{ij}) \oplus f_{pos}^{ij}, \quad (11)$$

where $\hat{o}^{ij}$ denotes the points inside intersection space b_{ij}, $\oplus$ the concatenation.

3.3 Contextual Reasoning for Scene Graph Generation

The entity and relation features extracted by the graph feature extraction module form a fully connected graph. Message passing propagates messages through the graph to incorporate contextual information into each node. The contextual information here refers to the underlying dependencies and relations existing in 3D point clouds, which are highly noisy relative to 2D images, so the indiscriminate information communication will aggravate noise propagation. Therefore, we utilize the GSL block to reconstruct the fully connected graph into a skeleton graph, which

effectively organizes contextual information by assigning weights to different nodes in the adjacent node set. After the message passing phase, the HOL block establishes long-term connections between initial features and contextual representations by forming coarse-grained and fine-grained labels into a two-level hierarchical tree. We adopt a multi-task learning approach, introduce GSL and HOL blocks on top of the message passing block, and utilize structured organization and hierarchical inferring to achieve context-based reasoning. The implementation details of the three blocks are shown in the Fig. 4.

In particular, the relation features $F_{\mathcal{R}}$ are first fed into Graph Skeleton Learning (GSL) block to generate an edge weighted graph skeleton, or in other words a structured organization rule set $\mathcal{S}_G = \{r_{12}, r_{13}, ..., r_{n,n-1}\}$. Then entity features $F_{\mathcal{E}}$, relation features $F_{\mathcal{R}}$ and $\mathcal{S}_G$ are then feed into the Message Passing block, a graph neural network [58], [59], to obtain structurally organized $\tilde{F}_{\mathcal{E}}$ and $\tilde{F}_{\mathcal{R}}$. $\tilde{F}_{\mathcal{R}}$ will then be fed into a predicate predictor to obtain the final predicate set P. Finally, we predict the coarse-grained and fine-grained labels for each entity by inputting $F_{\mathcal{E}}$ and $\tilde{F}_{\mathcal{E}}$ to Hierarchy Object Learning block. The details of each module are described below.

Graph Skeleton Learning The message passing block implements contextual representation learning by aggregating features between associated nodes. To compute the associations between nodes, we extract the graph skeleton information, which consists of the scene graph ground truth with node and edge labels removed. As shown in Fig. 5, under the supervision of the graph skeleton, the GSL block can be viewed as a binary classification task of judging whether there is a relationship between nodes. Its classification confidence can intuitively reflect the associations between nodes. By replacing the edges with the classification confidence, the fully connected graph is reconstructed as an edge-weighted skeleton graph. Our GSL module is implemented by a 3-layer fully connected network with ReLU non-linearity between each layer. It directly takes the initial relation features $F_{\mathcal{R}}$ as input and defines organizational rules for each edge on the fully connected graph. In particular, for predicate p_{ij} from entity e_i to e_j, GSL block takes $f_{\mathcal{R}}^{ij}$ as input, and predicts its organizational rule r_{ij}. We only take the positive predictive value as the final confidence score and normalize it with $softmax$ option. To achieve hard control for high or low confidence scores, we then feed these scores into
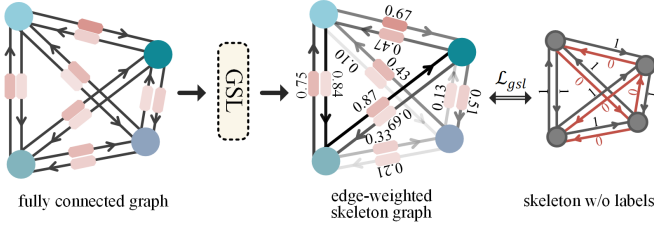
Fig. 5. **Edge-weighted Skeleton Graph.** The graph skeleton information consists of the scene graph ground truth with labels removed. After passing through the GSL block, the edges in the fully connected graph are replaced by the classification confidence. Colors of different edges represent confidence, dark colors represent high confidence, and vice versa.
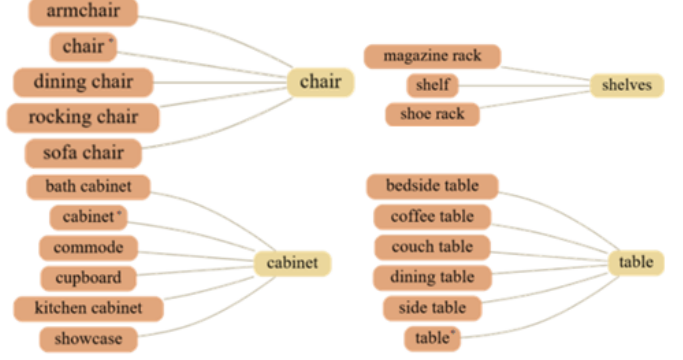

Fig. 6. **Two-layer Hierarchical Tree.** We show part of the labels in the form of a tree structure, where the word marked in orange color indicates the fine-grained class categories, and the word marked in yellow color indicates the coarse-grained NYU40 [53] object categories.

the gating function $\tau(\cdot)$ proposed by [60]. This function can make the predicted organization closer to GT scene graph skeleton. The above process can be described as follows:

$$r_{ij} = \tau\left(softmax(GSL(f_{\mathcal{R}}^{ij}))\right),$$

$$\tau(x) = \begin{cases} 0 & x \le \beta \\ \alpha x - \alpha\beta & \beta < x < 1/\alpha + \beta \\ 1 & x \ge 1/\alpha + \beta \end{cases}, \quad (12)$$

where α and β are two learnable hyperparameters.

Message Passing After computing the organizational rules $\mathcal{S}_G$, we propagate information from each entity to its relevant predicates and vice versa under the guidance of these rules. To this end, we use a multi-stage bipartite graph neural network to propagate the contextual feature on the fully-connected graph within the constraints of organizational rules to get discriminative feature representations. More specifically, the organizational rule r_{ij} affects the information flow of both entity-to-predicate $g^{e\to p}$ and predicate-to-entity $g^{p\to e}$. At time step t, we denote the hidden state of the entity node o_i as $H_{e_i}^{(t)}$ and the hidden state of the predicate node o_{ij} as $H_{p_{ij}}^{(t)}$. We use the feature vector $f_{\mathcal{E}}^i$ and $f_{\mathcal{R}}^{ij}$ obtained by the feature extraction module to initialize the hidden state.

$$H_{e_i}^{(0)} = f_{\mathcal{E}}^i, \quad H_{p_{ij}}^{(0)} = f_{\mathcal{R}}^{ij} \quad (13)$$

Our visual pattern narrows the perceptual area to reduce repetitive modeling of the entity area. Therefore, more attention is paid to the area between objects than to the object area. This means that there is a lack of information about objects and surrounding contextual information in relation features. Therefore, this information needs to be supplemented in the message passing phase. Each predicate aggregates messages from its neighbors according to organizational rules, with the formula:

$$M_{p_{ij}}^{(t)} = g^{e\to p}(r_{ij}, H_{e_i}^{(t)}, H_{e_j}^{(t)})$$
$$= mean\left(g^s\left(r_{ij} \cdot H_{e_i}^{(t)}\right) + g^o\left(r_{ij} \cdot H_{e_j}^{(t)}\right)\right), \quad (14)$$

where $M_{p_{ij}}^{(t)}$ denotes predicate p_{ij}'s message at iteration t, g^s and g^o are independent multi-layer perceptron (MLP) for subject and object, along with a mean operation $mean(\cdot)$.

For each entity node e_i, its neighbor nodes will be divided into two categories: N_{i*} and N_{*i}, respectively corresponding to the neighbor nodes where e_i is the subject and

e_i is the object. Message from different relationship nodes is aggregated according to their corresponding organizational rules, using the formula:

$$M_{e_i}^{(t)} = g^{p\to e}(r_{ij}, r_{ji}, H_{p_{ij}}^{(t)}, H_{p_{ji}}^{(t)})$$
$$= mean\left(\sum_{j\in N_{i*}} g^p\left(r_{ij} \cdot H_{p_{ij}}^{(t)}\right)\right.$$
$$\left. + \sum_{j\in N_{*i}(i)} g^p\left(r_{ji} \cdot H_{p_{ji}}^{(t)}\right)\right), \quad (15)$$

where $M_{e_i}^{(t)}$ denotes entity e_i's message at iteration t, g^p is a independent multi-layer perceptron (MLP) for predicate. After the message passing process, we update the feature using two Gated Recurrent Units (GRU) [61]:

$$H_{e_i}^{(t+1)} = GRU_e\left(H_{e_i}^{(t)}, M_{e_i}^{(t)}\right), \quad (16)$$
$$H_{p_{ij}}^{(t+1)} = GRU_p\left(H_{p_{ij}}^{(t)}, M_{p_{ij}}^{(t)}\right). \quad (17)$$

We fuse the information u times to get the final updated contextual-fused object feature $\tilde{F}_{\mathcal{E}} = \{H_{e_1}^{(u)}, H_{e_2}^{(u)}, ..., H_{e_n}^{(u)}\}$ and relation feature $\tilde{F}_{\mathcal{R}} = \{H_{p_{12}}^{(u)}, H_{p_{13}}^{(u)}, ..., H_{p_{n,n-1}}^{(u)}\}$. The organizational rules generated by GSL based on the initial features remain unchanged in the subsequent u fusion processes to prevent the fused relation features from strengthening the association between nodes. After feeding $\tilde{F}_{\mathcal{R}}$ into the predicate predictor, we obtain the relation labels $P = \{p_{12}^l, p_{13}^l, ..., p_{n,n-1}^l\}$.

Hierarchy Object Learning Due to the imbalance of relation and non-relation data, most communication between nodes will be blocked by GSL, and entity nodes can only obtain limited environmental information. Therefore, we utilize the HOL block to perform a hierarchical decomposition of the entity classification task. The HOL block uses coarse-grained and fine-grained entity labels to form a two-level hierarchical tree, as shown in Fig. 6. We construct hierarchical constraints for features containing attribute and contextual information, establishing long-term connections between features and enabling context-based reasoning on objects. The inputs of the HOL block are the initial entity feature $F_{\mathcal{E}}$ and information-fused feature $\tilde{F}_{\mathcal{E}}$. There are two

entity predictors in HOL block, each entity predictor is implemented by a 3-layer fully connected network with ReLU non-linearity between each layer, which predict coarse and fine entity labels separately:

$$l_{e_i}^c = Dec_{\mathcal{E}}^{coarse}(f_{\mathcal{E}}^i), \quad l_{e_i}^f = Dec_{\mathcal{E}}^{fine}(H_{e_i}^{(u)}). \quad (18)$$

After labeling e_i with $l_{e_i}^c$ and $l_{e_i}^f$, we obtain coarse labels $E^c = \{e_1^{l_c}, e_2^{l_c}, ...e_n^{l_c}\}$ and fine labels $E^f = \{e_1^{l_f}, e_2^{l_f}, ...e_n^{l_f}\}$. Combined with predicate set P, we finally generate the scene graph $\mathcal{G}$ from point cloud $\mathcal{O}$.

3.4 Loss Functions

The supervisory signal in the traditional SGG methods is deformed into a triplet form (e_i^l, p_{ij}^l, e_j^l), instead of scene graphs in the intuitive sense, which lacks structure and hierarchy. Under this deformation, the intuitive structured information is transformed into an implicit representation, leading to unsatisfactory performance. Thus we propose GSL block to generate graph skeleton $\mathcal{S}_G$ and supervised it with ground truth skeleton $\mathcal{S}_{GT} = \{b_{12}, b_{13}, ...b_{n,n-1}\}$, where $b_{ij} = 1$ indicates that there is an edge from e_i to e_j and $b_{ij} = 0$ the edge does not exist:

$$\mathcal{L}_{gsl} = BCE(\mathcal{S}_G, \mathcal{S}_{GT}), \quad (19)$$

where $BCE(\cdot)$ is a binary cross entropy loss function. Note that our graph skeleton data does not contain any label of nodes and edges, which is intended to limit the model to focus on learning structured information.

Regarding the HOL block, two granularity object labels were applied as supervision information. Hence, we split the ground truth scene graph into three supervisory signals: 1) skeleton graph; 2) coarse and fine-grained object labels, and 3) predicates. We train our model end-to-end, and our total loss function is described below:

$$\mathcal{L}_{total} = \mathcal{L}_{gsl} + \mathcal{L}_p + \lambda_c \mathcal{L}_e^c + \lambda_f \mathcal{L}_e^f, \quad (20)$$

where λ_c and λ_f are weighting factors, $\mathcal{L}_p$ denotes the per-class binary cross entropy loss, $\mathcal{L}_e^c$ and $\mathcal{L}_e^f$ are the standard cross entropy loss for multi-class classification task with coarse and fine-grained labels. By jointly training multiple tasks, the modules can be prevented from converging independently in a static space, resulting in better incorporation of structured information and hierarchical labels into visual information.

3.5 Implementation Details

We adopt PointNet [54] as the feature extraction network, which processes the points inside the object bounding box or interaction space with three channels (x, y, z) and outputs a final 256-d feature vector. Our message passing block iterates $u = 3$ times. The total number m of relation labels is 26. In our model, all entity and predicate predictors are composed of three fully connected layers followed by batch normalization and ReLU activation. The α, β are initialized as 2.2 and 0.025 as suggested by [60]. The hyperparameters λ_c, λ_f in our loss function are both set as 0.1. Adam is chosen as our model optimizer with a learning rate of 10^{-4}.

4 EXPERIMENTS

4.1 Dataset and Evaluation Metrics

3DSSG [14] is a large-scale 3D dataset extended from the 3RSCAN dataset [62] with scene graph annotations. It features 1482 scene graphs, which contain 534 classes of objects and 40 relationships. We take the same 160 object categories and 26 predicate labels as in [14]. The dataset provides a variety of object categories with different coarse and fine grains by mapping categories to NYU40 [53], RIO27 [62] and Eigen [63]. In this paper, NYU40 is selected as the coarse-grained object label, and the specific mapping relationship is provided in the supplementary material.

For evaluation, we applied the same scene-level split specified in [14] on the point cloud representations. Following [14], [15], [23], the scene graph prediction performance is evaluated upon the three perspectives using the top-k recall metric, namely object class prediction, predicate prediction, and relationship prediction. Of which the relationship level confidence scores are obtained by multiplying each respective score of the subject, predicate, and object in order.

4.2 Comparison with State-of-the-art Approaches

We evaluate our proposed method against state-of-the-art scene graph generation methods: MSDN [64], KERN [65], 3DSSG [14], BGNN [60], SGGPoint [15]. In addition to these methods, we also design a simple PointNet [54] based method by directly adding the same entity and predicate predictors to justify whether the discrimination of the features will be reduced. Among them, MSDN, KERN, and BGNN are 2D SSG methods. For a fair comparison, we removed the 2D object detector and added the same PointNet based feature extractor as ours (see more comparison results in the supplementary material).

Quantitative Results As shown in Tab. 1, our method is the only one that outperforms PointNet on all three sub-tasks without using any prior knowledge. It shows that our defined interaction space minimizes each node's information redundancy, and the contextual reasoning process retains the discrimination of features even after multiple iterations. Our method outperforms PointNet with significant margin of **6.7** and **5.1** on object class prediction. Besides, despite the absence of prior statistical co-occurrence knowledge, our method is slightly inferior to KERN by 0.2 and 0.5 on predicate prediction. Ultimately, our method outperforms the others on the relationship prediction sub-task by a large margin.

It is worth mentioning that to alleviate the severe object class imbalance problem in the SGG task, SGGPoint only retains 27 object classes and 16 relation classes (3DSSG-O27R16) in the 3DSSG dataset. And it also combines multi-label relationships between nodes into one relationship. Resulting in its network not being able to understand complex scenes with fine-grained objects and multiple relationships well (see our results on 3DSSG-O27R16 in the supplementary material). Our method can accurately classify entities at a fine-grained level, allowing the construction of scene graphs with more scene knowledge. To more directly compare the performance improvements of fine-grained object classes with MSDN, we further demonstrate the R@10 improvements for some entities in Fig. 7.

TABLE 1
Quantitative comparisons of our method against existing methods on 3DSSG [14] dataset. † denotes results reproduced with the authors' code.

Model	Object Class Prediction		Predicate Prediction		Relationship Prediction	
	R@5	R@10	R@3	R@5	R@50	R@100
PointNet† [54]	63.39	74.54	89.07	96.03	50.05	55.73
MSDN† [64]	61.07	72.41	85.99	93.60	46.55	53.20
KERN† [65]	66.58	76.52	**90.13**	**96.61**	51.36	58.49
3DSSG [14]	66.41	77.26	82.58	94.34	51.16	56.48
BGNN† [60]	71.19	81.98	86.98	93.80	55.20	60.85
SGGPoint [15]	27.82	35.85	68.18	87.32	7.94	9.91
Ours	**73.40**	**82.59**	89.90	96.10	**61.94**	**68.24**

TABLE 2
Quantitative comparisons of our method against existing methods by mR@K.

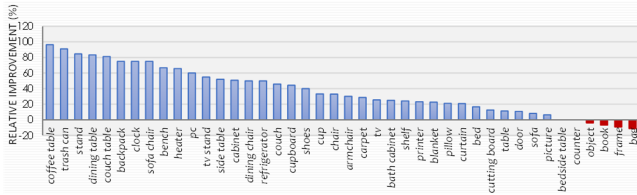| Model	Object Class Prediction		Predicate Prediction		Relationship Prediction	
	mR@5	mR@10	mR@3	mR@5	mR@50	mR@100
MSDN [64]	23.59	35.51	47.41	62.10	44.61	50.17
KERN [65]	23.48	35.89	45.68	61.97	43.46	49.14
3DSSG [14]	23.33	34.43	45.82	63.93	51.16	52.21
BGNN [60]	28.49	41.79	45.15	58.98	49.08	54.21
SGGPoint [15]	10.54	12.37	25.65	47.59	1.04	3.52
Ours	**33.87**	**45.18**	**47.10**	**64.16**	**53.21**	**61.50**

Fig. 7. **Quantitative prediction results about fine-grained object labels.** The R@10 improvement of different entities of our methods to the MSDN.
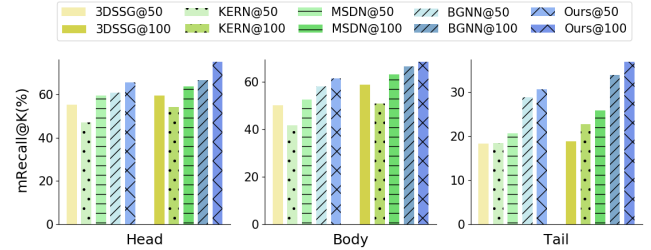

Fig. 8. **The performance of overcoming the long tail effect.** The per-group mR@50 and mR@100 results on each long-tail category groups in relationship prediction subtask.

It is well-known that SGG models trained on biased datasets have low performances for less frequent categories. Therefore, we additionally introduce mean recall (mR@K) of [66] to examine how well the method learns for uncommon predicates. The metric independently computes the recall for each predicate category and averages the results. So, each category contributes equally. Since our method does not rely on labels or prior knowledge, both of which are biased information, the long-tail effect can be mitigated. As shown in Tab. 2, our method outperforms all three sub-tasks, effectively reducing the impact of some common but meaningless predicates, such as `on`, `near by`, and paying equal attention to those less common predicates, such as `build in`, `belonging to`, which are more valuable for high-level reasoning.

Moreover, we divided the relationship categories into three disjoint groups according to the instance number in the training split: head (more than 10^4), body ($10^3 \sim 10^4$), and tail (less than 10^3). As shown in Fig. 8, we compute the mean recall on each long-tail category group in the relationship prediction sub-task and find our method significantly outperforms the prior works on the tail group. As a result, we achieve the highest mean recall over all categories.

Qualitative Results As shown in Fig. 9, we chose three categories of indoor scenes, kitchen, cafe, and study room, to verify that the proposed method can accurately explore contextual information to predict scene entities and relationships. In complex environments like kitchens, our method can identify small objects like `plates` and `stoves` and accurately distinguish between `kitchen cabinet` and `kitchen counter` with similar characteristics. For relations, we successfully predict `stove-built in-kitchen counter` when other methods only can predict `lying on`,

which requires a deeper semantic understanding of the relationship.

Our method has a stable prediction for scenes with certain environmental patterns, such as a cafe with multiple sets of tables and chairs. Meanwhile, the prediction result also reflects the drawback of KERN: when the object labels are predicted incorrectly, the relation prediction will be greatly affected, as wrong object labels will bring invalid co-occurrence knowledge to the process of feature information organization. Since KERN relies heavily on prior knowledge, the final SGG results are inconsistent with actual visual information, which is also present in other methods that simply utilize prior knowledge to organize environmental information (see more results in the supplementary materials). Compared with other methods, our result has an excellent performance in predicting the bidirectional relationship between objects. Our method can predict multiple labels accurately from similar features by constructing more discriminative features through effective feature extraction and contextual information fusion (see more qualitative results in the supplementary material). We can also predict incomplete objects in the study scene, such as `side table` and `commode`, by supplementing the surrounding contextual information.

4.3 Ablation Study

Model Components As shown in Tab. 3, we first verify the effectiveness of each component by incrementally adding each one of them to a common baseline MSDN [64], denoted as M0. It is worth mentioning that, MSDN uses the union region as the visual pattern, and transfers information indiscriminately between entities and relationships based on a bipartite graph neural network.

Fig. 9. **Comparisons against the state-of-the-arts.** We only show the results for a local area of the scene. We use gray boxes to indicate the entities and underlines to indicate the relations. The different colors of entity names and relation labels are the same as the colors of the class-agnostic instance labels, except that red indicates an incorrect prediction result.
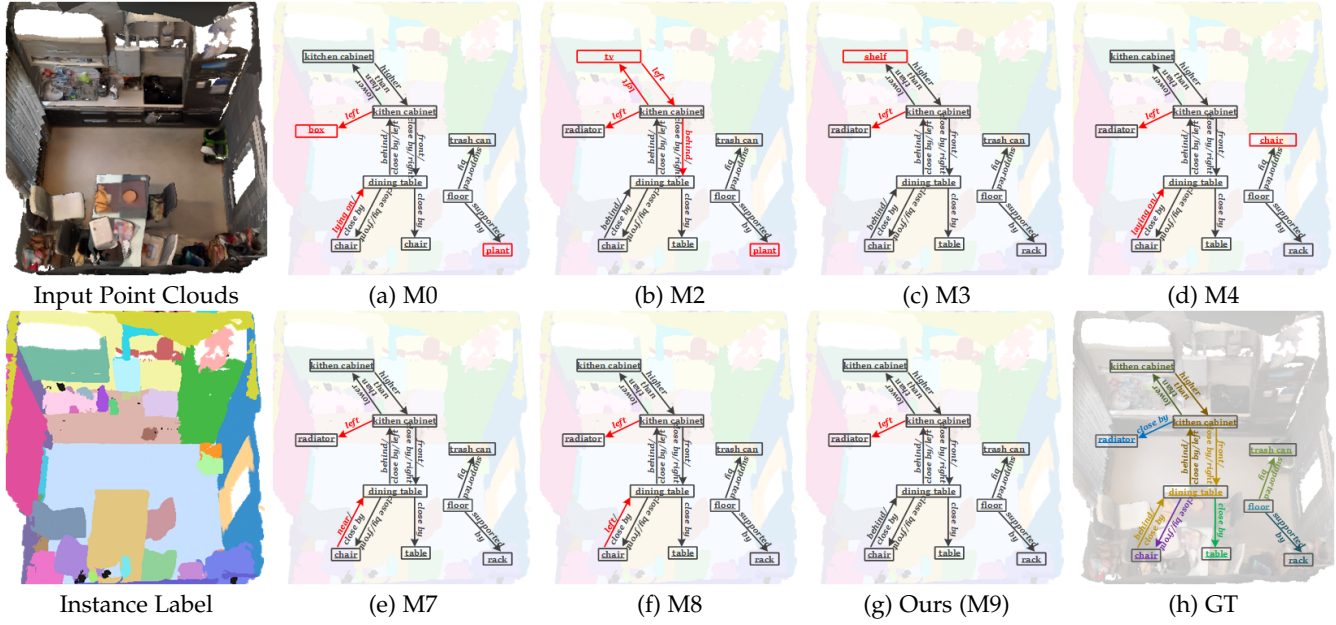
Fig. 10. **Visualization results for ablation study.** For clarity, we only show the results for a locally representative area of the scene. The rest of the annotation is the same as described above.

TABLE 3
Ablation study of our proposed network. InS: Interaction space; PoF: Position feature; MsP: Message Passing block; GSL: Graph Skeleton Learning block; HOL: Hierarchy Object Learning block. →**X** indicates replacing the corresponding component with **X**. ✓ and ✗ represent yes and no, respectively. The best results are marked in bold.

Method	Graph Feature Extraction		Graph Contextual Reasoning			Object Class Prediction		Predicate Prediction		Relationship Prediction	
	InS	PoF	MsP	GSL	HOL	R@50	R@100	R@50	R@100	R@50	R@100
M0	→ Union	✗	✓	✗	✗	61.07	72.41	85.99	93.60	46.55	53.20
M1	→ InT	✗	✓	✗	✗	61.62	73.02	83.72	92.99	46.98	53.92
M2	✓	✗	✓	✗	✗	63.18	74.67	84.96	93.57	48.92	56.25
M3	✓	✓	✓	✗	✗	69.65	79.75	91.41	**96.70**	60.92	66.24
M4	→ Union	✗	✓	✓	✗	71.97	81.54	87.44	94.26	58.97	65.05
M5	→ Union	✗	✓	→ KERN	✗	66.54	76.53	89.10	95.31	51.32	57.98
M6	→ Union	✗	✓	→ BGNN	✗	70.10	80.28	84.59	92.40	54.18	59.60
M7	→ Union	✗	✓	✗	✓	65.69	76.44	86.24	93.23	50.32	56.78
M8	→ Union	✗	✓	✓	✓	72.87	81.92	87.62	94.92	60.41	66.16
M9	✓	✓	✓	✓	✓	**73.40**	**82.59**	**91.43**	96.49	**61.94**	**68.24**

We first make quantitative ablation studies on each component. As we claimed in our paper, repeated modeling and meaningless information communication in the contextual extraction and fusion stages are the main causes for the indistinguishable features. **InS** reduces the repetitive modeling of the object area by narrowing the perception area (M2: results in object accuracy improved by 2 points). On the other hand, **PoF** provides InS with information such as the size, location, and subject-predicate relationship of objects that it cannot perceive (M3: results in improved object and predicate accuracy, and relationship prediction task improved for **13** points). Another source of noise is indiscriminate information fusion. Therefore, **GSL** reduces meaningless information transmission in the message passing process, and **HOL** imposes hierarchical constraints on the features before and after fusion (M8: relationship accuracy improved by **12** points).

Visual Patterns As shown in Tab. 3, we additionally provide the quantitative comparison results between our interaction space and the intersection region (InT-M1) of [18]. To fit the data characteristics of 3D point clouds, we expand the interaction space, effectively improving the quality of relation features and the accuracy of relationship prediction by 2 points.

Graph Skeleton Learning Block Our GSL module provides rules for organizing information at the message passing stage. Moreover, as shown in Tab. 3, our rules can better reduce meaningless information communication compared with the organization rules established by using prior knowledge (KERN-M5) or relationship labels (BGNN-M6), thereby reducing the noise in features.

Qualitative Results As shown in Fig. 10, the specific analysis of each case is as follows:

(1) M2: Interaction space is often tiny and covers less environmental context than the union region, which causes lower relationship prediction accuracy. However, due to the reduction of information redundancy, the accuracy of object prediction is improved. As shown in Fig. 10(a) and Fig. 10(b), the entity label radiator has been successfully predicted, but the number of incorrectly predicted relations

has also increased.

(2) M3: PoF compensates for the missing position information in InS. Because InS cannot perceive object positions and subject-object orders, resulting in the same features on the bidirectional edge between objects. As shown in Fig. 10(b), model predict the bidirectional edges as `left` and cannot perceive `higher than` and `lower than` relationships because the lack of object positions. By adding PoF, as shown in Fig. 10(c), the incorrect relation predictions are corrected while maintaining the accuracy of object prediction.

(3) M4: Compared to baseline MSDN, the additional structured organization rules enable the algorithm to handle the information with high redundancy, even if it has been iterated many times, and still retain sufficient feature discrimination. As shown in Fig. 10(a) and Fig. 10(d), structured organization rules improve the prediction results for both entities and relations.

(4) M7: By processing information hierarchically, we realize the process of classifying objects from coarse to fine-grained, and substantially improve object recognition accuracy. As shown in Fig. 10(a) and Fig. 10(e), the prediction of entities is significantly improved.

(5) M8: The GSL and HOL blocks together with the Message Passing block form the Graph Contextual Reasoning module, which structurally organizes the information and hierarchically infers the coarse to fine features, as shown in Fig. 10(f).

5 CONCLUSION

In this work, we propose a framework for the 3D scene graph generation. It explores contextual information via a well-designed graph feature extraction module and a graph contextual reasoning module. First, we propose a new visual pattern with appropriate information redundancy. Second, the features extracted from visual patterns are contextually fused by structured organization and hierarchical inferring, retaining the discrimination of features. The experiments demonstrate that the proposed method significantly outperforms the state-of-the-arts methods. In the future, we will try to perform scene understanding for more incomplete scenes through scene completion [67] or object reconstruction [68].

REFERENCES

[1] Y. Lv, N. Xie, Y. Shi, Z. Wang, and H. T. Shen, "Improving target-driven visual navigation with attention on 3d spatial relationships," *arXiv preprint arXiv:2005.02153*, 2020.

[2] U.-H. Kim, J.-M. Park, T.-J. Song, and J.-H. Kim, "3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents," *IEEE Transactions on Cybernetics*, vol. 50, no. 12, pp. 4921–4933, 2019.

[3] Y. Zhou, Z. While, and E. Kalogerakis, "Scenegraphnet: Neural message passing for 3d indoor scene augmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7384–7392.

[4] H. Dhamo, F. Manhardt, N. Navab, and F. Tombari, "Graph-to-3d: End-to-end generation and manipulation of 3d scenes using scene graphs," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 352–16 361.

[5] G. Yang, J. Zhang, Y. Zhang, B. Wu, and Y. Yang, "Probabilistic modeling of semantic ambiguity for scene graph generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 527–12 536.

[6] M. Suhail, A. Mittal, B. Siddiquie, C. Broaddus, J. Eledath, G. Medioni, and L. Sigal, "Energy-based learning for scene graph generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 936–13 945.

[7] R. Herzig, M. Raboh, G. Chechik, J. Berant, and A. Globerson, "Mapping images to scene graphs with permutation-invariant structured prediction," *Advances in Neural Information Processing Systems*, vol. 31, pp. 7211–7221, 2018.

[8] A. Zareian, S. Karaman, and S.-F. Chang, "Bridging knowledge graphs to generate scene graphs," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 606–623.

[9] A. Newell and J. Deng, "Pixels to graphs by associative embedding," *arXiv preprint arXiv:1706.07365*, 2017.

[10] J. Zhang, K. J. Shih, A. Elgammal, A. Tao, and B. Catanzaro, "Graphical contrastive losses for scene graph parsing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 535–11 543.

[11] Y. Li and G. Baciu, "Sg-gan: Adversarial self-attention gcn for point cloud topological parts generation," *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[12] S. Cai, L. Li, J. Deng, B. Zhang, Z.-J. Zha, L. Su, and Q. Huang, "Rethinking graph neural architecture search from message-passing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6657–6666.

[13] 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, 2020.

[14] J. Wald, H. Dhamo, N. Navab, and F. Tombari, "Learning 3d semantic scene graphs from 3d indoor reconstructions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3961–3970.

[15] C. Zhang, J. Yu, Y. Song, and W. Cai, "Exploiting edge-oriented reasoning for 3d point-based scene graph analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9705–9715.

[16] A. Dieckmann and J. Rieskamp, "The influence of information redundancy on probabilistic inferences," *Memory & Cognition*, vol. 35, no. 7, pp. 1801–1813, 2007.

[17] M. Sarafyazd and M. Jazayeri, "Hierarchical reasoning by neural circuits in the frontal cortex," *Science*, vol. 364, no. 6441, 2019.

[18] W. Wang, R. Wang, S. Shan, and X. Chen, "Exploring context and visual pattern of relationship for scene graph generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8188–8197.

[19] X. Lin, C. Ding, J. Zeng, and D. Tao, "Gps-net: Graph property sensing network for scene graph generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3746–3753.

[20] L. Chen, H. Zhang, J. Xiao, X. He, S. Pu, and S.-F. Chang, "Counterfactual critic multi-agent training for scene graph generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4613–4623.

[21] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3668–3678.

[22] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese, "3d scene graph: A structure for unified semantics, 3d space, and camera," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5664–5673.

[23] S.-C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari, "Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7515–7525.

[24] S. Zhang, A. Hao, H. Qin *et al.*, "Knowledge-inspired 3d scene graph prediction in point cloud," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 620–18 632, 2021.

[25] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[26] J. Gu, H. Zhao, Z. Lin, S. Li, J. Cai, and M. Ling, "Scene graph generation with external knowledge and image reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1969–1978.

[27] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 852–869.

[28] S. Sharifzadeh, S. M. Baharlou, and V. Tresp, "Classification by attention: Scene graph classification with prior knowledge," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 6, 2021, pp. 5025–5033.

[29] 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*, vol. 38, no. 5, pp. 1–12, 2019.

[30] X. Lu, S. Wu, H. Chen, S.-K. Yeung, W. Chen, and M. Zwicker, "Gpf: Gmm-inspired feature-preserving point set filtering," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 8, pp. 2315–2326, 2017.

[31] Z. Shu, S. Xin, X. Xu, L. Liu, and L. Kavan, "Detecting 3d points of interest using multiple features and stacked auto-encoder," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 8, pp. 2583–2596, 2018.

[32] Y. Teng, L. Wang, Z. Li, and G. Wu, "Target adaptive context aggregation for video scene graph generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 688–13 697.

[33] A. Zareian, Z. Wang, H. You, and S.-F. Chang, "Learning visual commonsense for robust scene graph generation," in *European Conference on Computer Vision*. Springer, 2020, pp. 642–657.

[34] S. Khandelwal, M. Suhail, and L. Sigal, "Segmentation-grounded scene graph generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 879–15 889.

[35] Y. Lu, H. Rai, J. Chang, B. Knyazev, G. Yu, S. Shekhar, G. W. Taylor, and M. Volkovs, "Context-aware scene graph generation with seq2seq transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 931–15 941.

[36] Y. Guo, J. Song, L. Gao, and H. T. Shen, "One-shot scene graph generation," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3090–3098.

[37] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5831–5840.

[38] M.-J. Chiou, H. Ding, H. Yan, C. Wang, R. Zimmermann, and J. Feng, "Recovering the unbiased scene graphs from the biased ones," in *Proceedings of the ACM International Conference on Multimedia*, 2021, pp. 1581–1590.

[39] G. Ren, L. Ren, Y. Liao, S. Liu, B. Li, J. Han, and S. Yan, "Scene graph generation with hierarchical context," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 909–915, 2020.

[40] B. Dai, Y. Zhang, and D. Lin, "Detecting visual relationships with deep relational networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3076–3086.

[41] S. Woo, D. Kim, D. Cho, and I. S. Kweon, "Linknet: Relational embedding for scene graph," *arXiv preprint arXiv:1811.06410*, 2018.

[42] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5410–5419.

[43] T.-J. J. Wang, S. Pehlivan, and J. Laaksonen, "Tackling the unannotated: Scene graph generation with bias-reduced models," *arXiv preprint arXiv:2008.07832*, 2020.

[44] X. Yang, K. Tang, H. Zhang, and J. Cai, "Auto-encoding scene graphs for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 685–10 694.

[45] M. Qi, W. Li, Z. Yang, Y. Wang, and J. Luo, "Attentive relational networks for mapping images to scene graphs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3957–3966.

[46] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph r-cnn for scene graph generation," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 670–685.

[47] Y. Yao, A. Zhang, X. Han, M. Li, C. Weber, Z. Liu, S. Wermter, and M. Sun, "Visual distant supervision for scene graph generation," *arXiv preprint arXiv:2103.15365*, 2021.

[48] W. Wang, R. Wang, S. Shan, and X. Chen, "Sketching image gist: Human-mimetic hierarchical scene graph generation," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 222–239.

[49] Y. Cong, H. Ackermann, W. Liao, M. Y. Yang, and B. Rosenhahn, "Nodis: Neural ordinary differential scene understanding," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 636–653.

[50] G. Yin, L. Sheng, B. Liu, N. Yu, X. Wang, J. Shao, and C. C. Loy, "Zoom-net: Mining deep feature interactions for visual relationship recognition," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 322–338.

[51] Y. Li, W. Ouyang, B. Zhou, J. Shi, C. Zhang, and X. Wang, "Factorizable net: an efficient subgraph-based framework for scene graph generation," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 335–351.

[52] W. Zhang, Z. Dong, J. Liu, Q. Yan, C. Xiao *et al.*, "Point cloud completion via skeleton-detail transformer," *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[53] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Proceedings of the European Conference on Computer Vision*, 2012, pp. 746–760.

[54] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[55] R. Hu, C. Zhu, O. van Kaick, L. Liu, A. Shamir, and H. Zhang, "Interaction context (icon) towards a geometric functionality descriptor," *ACM Transactions on Graphics*, vol. 34, no. 4, pp. 1–12, 2015.

[56] N. Dhingra, F. Ritter, and A. Kunz, "Bgt-net: Bidirectional gru transformer network for scene graph generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2150–2159.

[57] Y. Zhan, J. Yu, T. Yu, and D. Tao, "On exploring undetermined relationships for visual relationship detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5128–5137.

[58] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[59] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *arXiv preprint arXiv:1511.05493*, 2015.

[60] R. Li, S. Zhang, B. Wan, and X. He, "Bipartite graph network with adaptive message passing for unbiased scene graph generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 109–11 119.

[61] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[62] J. Wald, A. Avetisyan, N. Navab, F. Tombari, and M. Nießner, "Rio: 3d object instance re-localization in changing indoor environments," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7658–7667.

[63] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Advances in neural information processing systems*, vol. 27, 2014.

[64] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang, "Scene graph generation from objects, phrases and region captions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017, pp. 1261–1270.

[65] T. Chen, W. Yu, R. Chen, and L. Lin, "Knowledge-embedded routing network for scene graph generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6163–6171.

[66] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu, "Learning to compose dynamic tree structures for visual contexts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6619–6628.

[67] X. Han, Z. Zhang, D. Du, M. Yang, J. Yu, P. Pan, X. Yang, L. Liu, Z. Xiong, and S. Cui, "Deep reinforcement learning of volume-guided progressive view inpainting for 3d point scene completion from a single depth image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 234–243.

[68] X. Yang, Y. Wang, Y. Wang, B. Yin, Q. Zhang, X. Wei, and H. Fu, "Active object reconstruction using a guided view planner," *arXiv preprint arXiv:1805.03081*, 2018.

Yuanyuan Liu received the B.Eng. degree in the Computer Science and Technology from Northeast Forestry University, Harbin, China. She is currently working towards the Ph.D. degree with the Department of Computer Science and Technology, Dalian, China. Her research interests include scene understanding and computer vision.

Qiang Zhang was born in Xian, China, in 1971. He received his M.Eng. degree in economic engineering and Ph.D degree in circuits and systems from Xidian University, Xian, China, in 1999 and 2002, respectively. He was a lecturer at the Center of Advanced Design Technology, Dalian University, Dalian, China, in 2003 and was a professor in 2005. His research interests are bio-inspired computing and its applications. He has authored more than 70 papers in the above fields. Thus far, he has served on the editorial board of seven international journals and has edited special issues in journals such as *Neurocomputing* and *International Journal of Computer Applications in Technology*.

Chengjiang Long is currently a Research Scientist at Meta Reality Labs. Prior joining Meta, he worked as a Principal Scientist/Tech Leader in JD Tech R&D Center at Silicon Valley from June 2020 to Dec 2021, and worked as a Computer Vision Researcher/Senior R&D Engineer at Kitware from February 2016 to April 2020. He also worked as an Adjunct Professor at University at Albany, SUNY from August 2018 to May 2020, and was an Adjunct Professor at Rensselaer Polytechnic Institute (RPI) from Jan 2018 to May 2018. He received the M.S. degree in Computer Science from Wuhan University in 2011 and a B.S degree in Computer Science and Technology from Wuhan University in 2009. He got his Ph.D. degree in Computer Science from Stevens Institute of Technology in 2015. During his Ph.D. study, he worked at NEC Labs America and GE Global Research as a research intern in 2013 and 2015, respectively. To date, he has published over 65 papers including top journals such as TOG, TPAMI, IJCV, TIP and TMM, top international conferences such as SIGGRAPH Asia, CVPR, ICCV, AAAI, and ACM MM, and owns 1 patent. He is also the reviewer for more than 20 top international journals and conferences. His research interests involve various areas of Computer Vision, Computer Graphics, Multimedia, Machine Learning, and Artificial Intelligence. He is a member of IEEE and AAAI.

Baocai Yin is a professor and doctoral advisor at Dalian University of Technology. He received his Ph.D. degree in computational mathematics from Dalian University of Technology (1990-1993), where he also received his M.S. degree in computational mathematics (1985-1988) and his B.S. degree in applied mathematics(1981-1985). His research areas include digital multimedia technology, virtual reality and graphics technology, and multi-function perception technology.

Zhaoxuan Zhang received the B.E. degree in School of Mathematical Sciences from Dalian University of Technology, Liaoning, China in 2016, where he is currently pursuing the Ph.D. degree in computer science. His current research interests include computer vision and computer graphics, especially the 3D reconstruction.

Xin Yang is a professor and doctoral advisor at Dalian University of Technology. He received his Ph.D. degree in computer science from Zhejiang University (2007-2012), and his B.S. degree in computer science from Jilin University (2003-2007). His main research interests include computer graphics and vision, intelligent robot technology, focusing on the efficient expression, understanding, perception and interaction of scenes.

Bokai Liu received the B.Eng. degree in the Computer Science and Technology from Taiyuan University of Technology, Taiyuan, China, in 2021. He is currently working toward the M.Sc. degree with the Department of Computer Science and Technology, Dalian University of Technology, Dalian, China. His research interests include scene understanding and computer vision.