This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

Hypergraph-Based Multi-Modal Representation for Open-Set 3D Object Retrieval

Yifan Feng, Shuyi Ji, Yu-Shen Liu, Shaoyi Du, *Member, IEEE,* Qionghai Dai, *Senior Member, IEEE* and Yue Gao*, *Senior Member, IEEE*

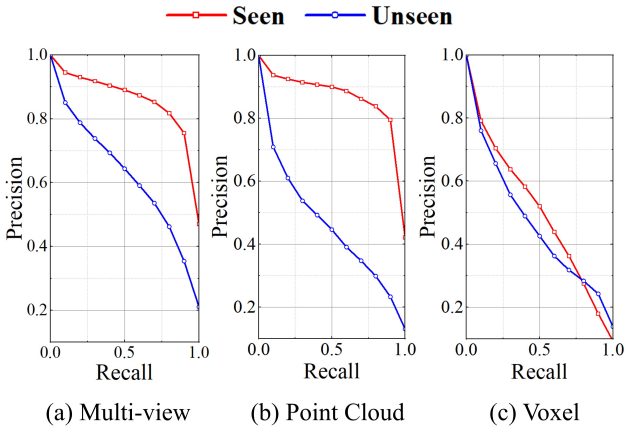
Abstract—The traditional 3D object retrieval (3DOR) task is under the close-set setting, which assumes the categories of objects in the retrieval stage are all seen in the training stage. Existing methods under this setting may tend to only lazily discriminate their categories, while not learning a generalized 3D object embedding. Under such circumstances, it is still a challenging and open problem in real-world applications due to the existence of various unseen categories. In this paper, we first introduce the open-set 3DOR task to expand the applications of the traditional 3DOR task. Then, we propose the Hypergraph-Based Multi-Modal Representation (HGM^{2}R) framework to learn 3D object embeddings from multi-modal representations under the open-set setting. The proposed framework is composed of two modules, *i.e.*, the Multi-Modal 3D Object Embedding (MM3DOE) module and the Structure-Aware and Invariant Knowledge Learning (SAIKL) module. By utilizing the collaborative information of modalities derived from the same 3D object, the MM3DOE module is able to overcome the distinction across different modality representations and generate unified 3D object embeddings. Then, the SAIKL module utilizes the constructed hypergraph structure to model the high-order correlation among 3D objects from both seen and unseen categories. The SAIKL module also includes a memory bank that stores typical representations of 3D objects. By aligning with those memory anchors in the memory bank, the aligned embeddings can integrate the invariant knowledge to exhibit a powerful generalized capacity toward unseen categories. We formally prove that hypergraph modeling has better representative capability on data correlation than graph modeling. We generate four multi-modal datasets for the open-set 3DOR task, *i.e.*, OS-ESB-core, OS-NTU-core, OS-MN40-core, and OS-ABO-core, in which each 3D object contains three modality representations: multi-view, point clouds, and voxel. Experiments on these four datasets show that the proposed method can significantly outperform existing methods. In particular, the proposed method outperforms the state-of-the-art by $12.12\%/12.88\%$ in terms of mAP on the OS-MN40-core/OS-ABO-core dataset, respectively. Results and visualizations demonstrate that the proposed method can effectively extract the generalized 3D object embeddings on the open-set 3DOR task and achieve satisfactory performance.

Index Terms—Hypergraph, Multi-Modal, 3D Object Retrieval, Open-Set, Memory Bank.

1 Introduction

3D objects, as the fundamental elements of the real world, have wide applications like autopilot and scene understanding. To characterize the objects in 3-dimensional space, different modalities (multi-view [9], point cloud [21], voxel [25], etc.) are introduced to provide a comprehensive understanding of 3D objects. Based on those 3D object representations, different 3D object learning tasks have been derived, such as 3D object retrieval (3DOR) [1], which plays a vital role in 3D object understanding and applications. The goal of the 3DOR task is to find the 3D objects that are similar to the query from the target set. Along this line, large efforts [23], [26], [34], [35], [36], [37], [38], [40] have been made in recent years.

One key challenge of the 3DOR task is how to learn discriminative embedding against the diversity across different 3D representations. Given a 3D object, representations under multiple modalities can be generated, such as multi-view, point cloud, and voxel, which are inherently different. For example, the multi-view and voxel are organized by the



(a) Multi-view (b) Point Cloud (c) Voxel

Fig. 1. The performance comparison (Precision-Recall Curve) of different modalities on ModelNet40, with respect to the seen queries and the unseen queries. We notice the gap of the Voxel modality (0.49 mAP on Seen and 0.45 mAP on Unseen) is smaller than the other two modalities, especially for the high-recall case. This is because the model of voxel modality cannot capture the general voxel description for embedding and falls into the over-fitting of biased feature extracting.

- *This work was supported by National Natural Science Funds of China (No. 62088102, 62021002), Open Research Projects of Zhejiang Lab (NO. 2021KG0AB05).*
- *Yifan Feng, Shuyi Ji, Yu-Shen Liu, and Yue Gao are with the School of Software, Tsinghua University, Beijing 100084, China. Yifan Feng, Shuyi Ji, Qionghai Dai, and Yue Gao are with BNRist, THUIBCS, BLBCI, Tsinghua University, Beijing 100084, China. Shaoyi Du is with Institute of Artificial Intelligence and Robotics, College of Artificial Intelligence, Xi'an Jiaotong University, Xi'an 710049, China.*
E-mail: evanfeng97@gmail.com; jisy19@mails.tsinghua.edu.cn; liuyushen@tsinghua.edu.cn; dushaoyi@xjtu.edu.cn; daiqionghai@tsinghua.edu.cn; gaoyue@tsinghua.edu.cn; (Corresponding author: Yue Gao)

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

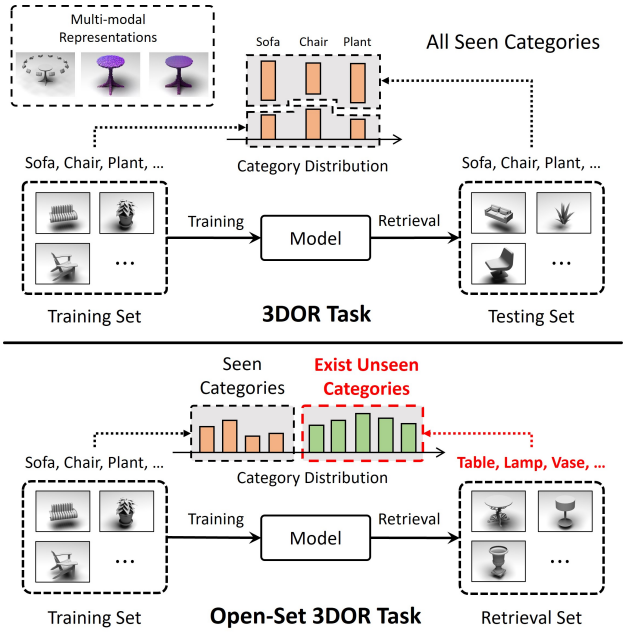


Fig. 2. The comparison between the traditional 3DOR task and the open-set 3DOR task. The top-left part of this illustration shows the multi-modal representations extracted from a given 3D object.

regular structure with different dimensions, while the point cloud is formulated by the irregular structure composed of some discrete points with (x, y, z) coordinates. Previous studies [35], [36], [40] first extract the modality features via modality-specific backbone (*e.g.*, MVCNN [9] and PointNet [21]), and then simply fuse the modality features with weighted concatenation or pull them closer in vector space via optimization objectives like cross-modal center loss [40]. However, direct fusion in this way cannot utilize the potential collaborative information of representations derived from the same 3D object.

Although recent years have witnessed the rapid development of 3DOR, existing solutions are mainly under the closed-set setting, which means the category information of the objects in the testing set is also seen in the training stage. Unfortunately, existing methods trained under the closed-set setting may achieve inferior performance in the condition that some unseen categories appear in the testing set. We have designed an experiment to explore how much influence the open-set setting can have. For the widely used ModelNet40 [7] dataset, we split the 40 categories into two parts: 20 as the seen categories and the other 20 as the unseen categories. Here, "seen" indicates that those categories of 3D objects are used for training, while "unseen" means that they are not used in the training stage. Three typical methods, *i.e.*, MVCNN [9] for the multi-view data, PointNet [21] for the point cloud data, and ShapeNets [25] for the voxel data, are used for comparison. As shown in Fig. 1, we can clearly observe the performance gaps between the seen queries and unseen queries. In particular, in the case of the multi-view and the point cloud modalities, the performance disparities are rather large. Under such circumstances, existing 3DOR methods may suffer from the open-set scenario in practice, and how to deal with the open-set scenario becomes an urgent and important task.

To tackle these issues, in this paper, we first introduce the open-set 3DOR task. Unlike existing open-set recognition task [42], [43], learning 3D objects under the open-set setting is more difficult. Usually, each 3D object is associated with multiple modalities (multi-view, point cloud, voxel, etc.), and the potential correlations among different modalities of 3D objects are hard to accurately model and learn. As depicted in Fig. 2, compared with the traditional 3DOR task under the closed-set setting, the open-set 3DOR task aims to provide robust retrieval performance given both seen and unseen categories by extracting more discriminative representations for generalized 3D objects, which can significantly expand their applications in the real world.

We then propose a Hypergraph-Based Multi-Modal Representation (HGM^{2}R) framework for the open-set 3DOR task, which is composed of two components, *i.e.*, the Multi-Modal 3D Object Embedding (MM3DOE) and the Structure-Aware and Invariant Knowledge Learning (SAIKL). The MM3DOE module includes multiple auto-encoders to learn the latent modality code for different modalities. Subsequently, the Homology Loss and the Bi-reconstruction Loss are introduced to pull the modality codes that are derived from the same 3D objects closer together and enhance the generalization ability of auto-encoders, respectively. In the SAIKL module, the hypergraph structure and hypergraph convolution are first introduced to model the implicit high-order correlation among 3D objects. Then, we propose the memory bank, which includes L memory anchors with each memory anchor denoting a typical representation of 3D objects. Next, to incorporate the invariant knowledge, each 3D object embedding is aligned with the memory anchors to generate the final aligned embedding. In this way, the SAIKL module can distill the invariant knowledge from seen categories to anticipate the unseen categories in the open-set 3DOR task. We formally prove that hypergraph modeling has better representative capability on data correlation than graph modeling. As there is no existing dataset under the open-set setting, we generate four multi-modal datasets toward the open-set 3DOR task, including OS-ESB-core, OS-NTU-core, OS-MN40-core, and OS-ABO-core, to evaluate the performance of the proposed HGM^{2}R framework. Experimental results show that the proposed method significantly outperforms other baseline methods on the open-set 3DOR task. In particular, HGM^{2}R surpasses the strongest baseline by $12.12\%/12.88\%$ in terms of mAP on the OS-MN40-core/OS-ABO-core dataset. Ablation studies and visualizations are also provided to validate the effectiveness of the proposed method. The main contributions of this paper are summarized as follows:

1) We introduce the open-set 3DOR task to expand the applications of the traditional 3DOR task, and release four multi-modal open-set 3DOR datasets, *i.e.*, the OS-ESB-core, OS-NTU-core, OS-MN40-core, and OS-ABO-core datasets for benchmarking.
2) We propose a HGM^{2}R framework for the open-set 3DOR task. The framework is composed of Multi-Modal 3D Object Embedding (MM3DOE) and Structure-Aware and Invariant Knowledge Learning (SAIKL), which are devised to overcome the divergence of different modalities from a given 3D object and fine-tune the 3D object embeddings to-

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

ward the open-set setting, respectively.

3) We utilize the hypergraph to model the high-order correlation of multi-model representations from seen and unseen categories and formally prove that hypergraph modeling has better representative capability on data correlation than graph modeling.
4) Experimental results on four open-set 3DOR datasets under general 3DOR setting, separate retrieval on seen and unseen categories, different input data setting, different coarse splittings setting, cross-dataset setting, and real 3D world setting indicate that the proposed method achieves significant improvements in the open-set 3DOR task compared to existing methods.

The remainder of this paper is organized as follows. We briefly review the related works on 3D object learning and retrieval in Sec. 2. In Sec. 3, we provide the definition of the closed-set 3DOR task and the open-set 3DOR task. Then, we introduce the proposed HGM^{2}R framework in Sec. 4. Experiments, visualizations, and discussions are provided in Sec. 5. We conclude this paper in Sec. 6.

2 Related Work

Since there is no literature specifically developed for the open-set 3D object retrieval task, we introduce some related work on 3D object representation, 3D object retrieval, and open-set learning in this section.

2.1 3D Object Representation

A 3D object can be represented by different modalities, *i.e.*, multi-view, point cloud, and voxel. The multi-view modality can be denoted by several 2D images, which can be learned by well-studied convolution neural networks like MVCNN [9] and GVCNN [10]. The point cloud modality is represented by a $P \times 3$ matrix, where P is the number of points in the point cloud. It is an irregular structure, which can be learned by the MLP/graph-based methods like PointNet [21], PointNet++ [22]. Furthermore, DGCNN [24] integrates the graph neural network [3] into the learning of the point cloud, which is achieved by aggregating the neighbors' features into the center points'. The voxel modality can be regarded as the dimension-expanded image, which can be addressed by 3D convolution neural networks such as ShapeNets [25] and VoxNet [26].

2.2 3D Object Retrieval

Existing 3D object retrieval methods can be mainly divided into two categories, *i.e.*, single-modality 3D object retrieval methods, and multi-modal 3D object retrieval methods. As for the single-modality 3D object retrieval, [11] and [12] aggregate information from multiple views via aligning distribution and correlations in the view-based graph, respectively, to retrieve 3D objects in view-based 3D representation. [34] propose a metric-based loss function that pulls objects from the same category together and pushes objects from different categories apart to yield discriminative representations in the 3D object retrieval task. Bai et al. propose the GIFT model [13] for real-time 3D object search, which projects 3D objects into view modality and captures a local distribution of 3D shapes in the feature manifold. Besides, some correlation-based methods [4], [14], [15] are proposed to model the complex correlations, which have the potential to model the complex high-order correlations among 3D objects. Feng et al. propose the HGNN [4] for view-based 3D object recognition, which models the high-order correlation among 3D objects with the hypergraph structure. To flexibly adjust the importance of high-cardinality hyperedges and high-degree vertices, HNHN [14] is proposed to apply nonlinear activation functions applied to both hyperedge message aggregation and vertex message aggregation. Besides, HGAT [15] computes the attention scores between the hyperedge and the associated vertices to adaptively aggregate messages from hyperedges with different importance.

As for the multi-modal 3D object retrieval, existing methods such as MMJM [35] and MIFN [36] adopt different backbones to extract modality-specific features, and fuse the multi-modal representations by the weighted concatenation. In the MMJM [35] framework, the discrimination loss is adopted to minimize the Euclidean distance between different modalities to learn discriminative embeddings. PVNet [18] and PVRNet [19] propose the joint frameworks that fuse the multi-view and point cloud with concatenation and attention mechanisms, respectively. Besides, Long et al. further propose the CMCL [40], which designs the cross-modal center loss to reduce the difference across different 3D representations by matching the modality center.

2.3 Open-Set Learning

In this subsection, we introduce some related open-set recognition works on 2D images and open-set learning methods on 3D objects, respectively. Vaze et al. [16] introduce a benchmark for open-set recognition, named Semantic Shift Benchmark (SSB), which better respects the task of detecting semantic novelty. Besides, they demonstrate that the ability of a classifier to make the 'none-of-above' decision is highly correlated with its accuracy on the closed-set classes. With slight modification on the existing classifier layer, PlaceholdeRs for Open-SEt Recognition (PROSER) [42] expands the closed-set classifier to detect whether the sample belongs to the seen categories or not. To reduce the empirical classification risk on the labeled known data and potential unknown data, Chen et al. [17] propose an adversarial-based method, named Adversarial Reciprocal Point Learning (ARPL), to minimize the overlap of known distribution and unknown distributions without loss of known classification accuracy.

Following existing open-set recognition works [16], [28], [29], [42] on 2D images, researchers attempt to investigate open-set 3D object learning. Alliegro *et al.* [30] also introduces a benchmark for open-set 3D object learning, which is collected by sampling points from synthetic 3D objects and segmenting points from the scanned scene point clouds. The authors develop baselines based on the pointnet++ [22] and DGCNN [24] and conduct many experimental settings for comparison, including the synthetic-to-synthetic, real-to-real, and synthetic-to-real open-set detection. To withstand the negative effect of the point clouds from unseen categories, Ma *et al.* [33] propose the PISVM framework

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

that adopts the Conditional Random Field (CRF) to capture inherent spatial relationships and appearance similarities between objects, and employs a Probability of Inclusion Support Vector Machine (PISVM) to estimate an unseen likelihood for each training class. Shi *et al.* [32] further formulate open-set semi-supervised point clouds learning as a bi-level optimization problem, and proposes a weight predictor network to estimate per-sample weights for unseen data. Besides, open-set recognition is also applied in point-clouds-based 3D object detection. Cen *et al.* [31] utilizes metric learning to exploit the detected 3D objects with low confidence and adopts unsupervised clustering to refine the bounding boxes of unknown 3D objects. However, existing works [30], [31], [32], [33] on the open-set 3D object learning only focus on a single modality: point clouds, and are designed only to detect whether the 3D object belongs to the unseen categories. Therefore, they cannot generate general 3D object features from multi-modal 3D representation. The general and discriminative 3D object features are important for real-world 3D applications, where 3D objects from unseen categories are very common. Thus, in this paper, we proposed the open-set 3D object retrieval task toward multi-modal 3D object representations.

3 Problem Definition

In this section, we first review the definition of the widely employed closed-set 3D object retrieval task. Then, we further provide the definition of the open-set 3DOR task.

3.1 Closed-Set 3D Object Retrieval

In traditional closed-set 3D object retrieval, the retrieval methods are designed using the training set $\mathcal{D}_{tra} = \{(o_i, y_i)\}_{i=1}^{L}$ and then used to search similar objects of the query in the testing set $\mathcal{D}_{tes} = \{(o_i, y_i)\}_{i=1}^{N}$. In a typical setting, the 3D object representation and the distance metrics are trained using the training set. Here, L and N denote the numbers of samples in the training set and testing set, respectively. We note that in some cases there is no training set, and all the data in the testing set is used for retrieval. The $o_i = \{m_k\}_{k=1}^{M}$ denotes a 3D object, which can be represented with M modalities, *i.e.*, multi-view, point cloud and voxel. The $y_i \in \mathcal{Y} = \{c_j\}_{j=1}^{Y}$ indicates the category associated with the 3D object o_i. Y is the total number of categories. In the closed-set assumption, $\mathcal{D}_{tra}$ and $\mathcal{D}_{tes}$ share the same category space and are drawn from the same distribution $\mathcal{D}$, which means that in the retrieval phase, all categories of objects in the testing set have been seen in the training phase. The goal of the retrieval task in the closed set is to minimize the expected risk:

$$f^* = \underset{f\in\mathcal{H}}{argmin}\, \mathbb{E}_{(D_i,D_j)\sim(\mathcal{D}_{tes},\mathcal{D}_{tes})} \Big[\mathbb{I}(y_i = y_j)\left(1 - e^{-\mathbb{D}(f(o_i),f(o_j))}\right) + \mathbb{I}(y_i \neq y_j) e^{-\mathbb{D}(f(o_i),f(o_j))}\Big], \tag{1}$$

where $D_i = (o_i, y_i)$ and $D_j = (o_j, y_j)$ are samples both drawn from the $\mathcal{D}_{tes}$. $\mathbb{I}(\cdot)$ is the indicator function, which returns 1 if the expression holds and 0 otherwise. $f := o_i \to x_i$ is the model that maps the 3D object o_i into a semantic embedding $x_i \in \mathbb{R}^d$. $\mathcal{H}$ is the hypothesis space of map $f(\cdot)$. $\mathbb{D}(x_i, x_j)$ is the distance metric function, which could measure the distance between two 3D object embeddings. In the closed-set setting, an expected 3D retrieval method can extract the features that minimize the distance of 3D objects from the same category and maximize the distance of 3D objects from different categories. However, the unseen categories are ubiquitous in practice. Therefore, the strong assumption (all categories in the retrieval phase should be seen in the training phase) on the closed-set 3DOR setting limits its generalizability.

3.2 Open-Set 3D Object Retrieval

Different from the traditional closed-set setting, we consider a more common condition that there are many unseen categories in the retrieval task. Under this circumstance, the retrieval model is still trained using the training set $\mathcal{D}_{tra} = \{(o_i, y_i)\}_{i=1}^{L}$, but is applied to search similar 3D objects in the retrieval set $\mathcal{D}_{ret} = \{(o_i, \hat{y}_i)\}_{i=1}^{R} = \{\mathcal{D}_q, \mathcal{D}_t\}$, where $\mathcal{D}_q$ and $\mathcal{D}_t$ are the query set and the target set, respectively. L and R denote the numbers of samples in the training set and the retrieval set, respectively. Different from the traditional closed-set 3D object retrieval, the category spaces of the training set and the retrieval set are not the same indicating $y_i \in \mathcal{Y} = \{c_j\}_{j=1}^{Y}$, $\hat{y}_i \in \hat{\mathcal{Y}} = \{\hat{c}_j\}_{j=1}^{\hat{Y}}$, and $\mathcal{Y} \neq \hat{\mathcal{Y}}$. Y and $\hat{Y}$ denote the numbers of categories in the training set and the retrieval set, respectively. Thus, the training set $\mathcal{D}_{tra}$ and retrieval set $\mathcal{D}_{ret}$ may have their individual distributions, and the query set $\mathcal{D}_q$ as well as the target set $\mathcal{D}_t$ are two subsets of the retrieval set $\mathcal{D}_{ret}$. It indicates that the query under the open-set setting could either belong to the object categories from the training set or not, while the answer for the closed-set setting is yes. The open-set 3DOR task aims to minimize the expected risk:

$$f^* = \underset{f\in\mathcal{H}}{argmin}\, \mathbb{E}_{(D_i,D_j)\sim(\mathcal{D}_q,\mathcal{D}_t)} \Big[\mathbb{I}(\hat{y}_i = \hat{y}_j)\left(1 - e^{-\mathbb{D}(f(o_i),f(o_j))}\right) + \mathbb{I}(\hat{y}_i \neq \hat{y}_j) e^{-\mathbb{D}(f(o_i),f(o_j))}\Big], \tag{2}$$

where $D_i = (o_i, \hat{y}_i)$ and $D_j = (o_j, \hat{y}_j)$ are samples drawn from the $\mathcal{D}_q$ and $\mathcal{D}_t$, respectively. Since the retrieval set may include many unseen categories, models trained in the open-set 3DOR task are able to handle more real-world applications compared with those trained in the closed-set 3DOR task.

4 Methodology

In this section, we first introduce the proposed Hypergraph-Based Multi-Modal Representation (HGM^{2}R) framework for open-set 3DOR task. The following two subsections will detail the two core components of the proposed framework, which are designed for learning the multi-modal 3D object embedding and learning the invariant knowledge for open-set retrieval, respectively.

4.1 Framework Overview

The overall framework of HGM^{2}R is illustrated in Fig. 3. HGM^{2}R is composed of the Multi-Modal 3D Object Embedding module and the Structure-Aware and Invariant Knowledge Learning module. Given a 3D object associated with multiple modality representations (multi-view, point

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

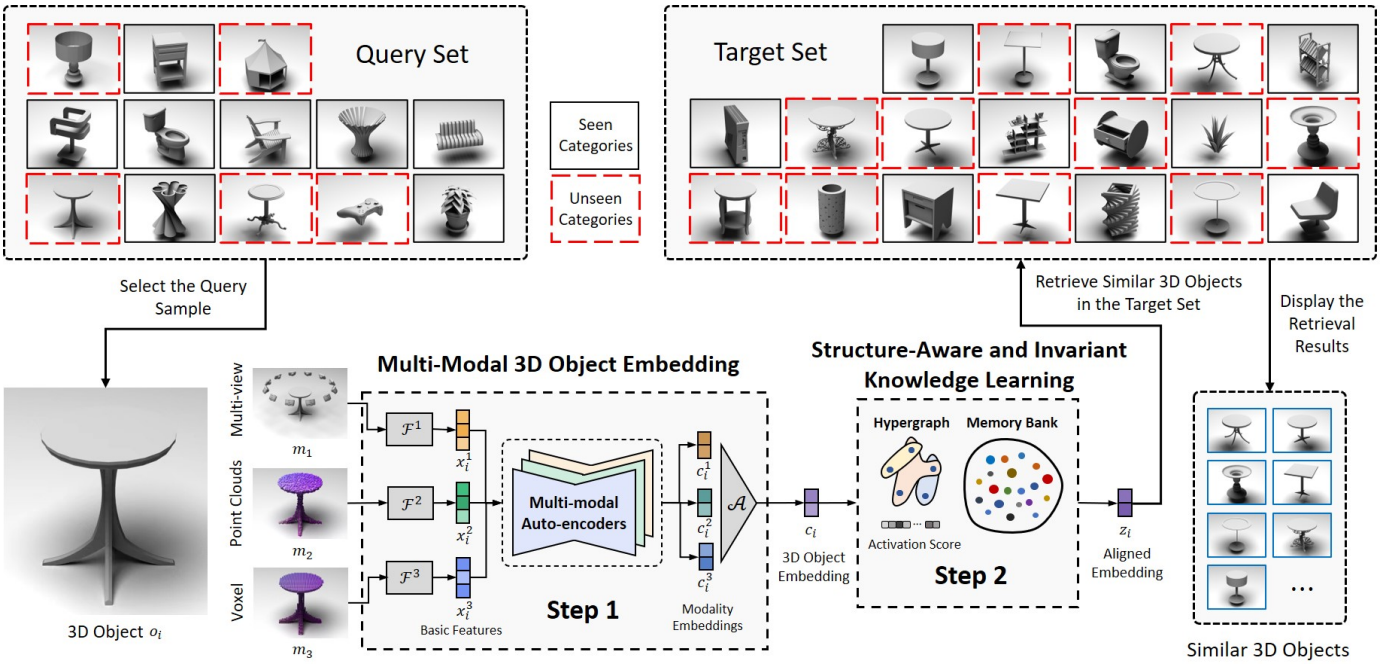


Fig. 3. The proposed HGM^{2}R framework. In the Query set and the Target Set, the 3D object enclosed by a red box indicates that the category of the object is not seen in the training stage, and the 3D object enclosed with a black box means its category is seen in the training stage. $\mathcal{F}^k$ is the basic feature representation methods for modality m_i, where m_i can be multi-view, point cloud, voxel, etc.

cloud, and voxel), typical feature representation methods are employed to extract the basic features for each modality. Then, the **Multi-modal 3D Object Embedding** (MM3DOE) is introduced to generate the unified 3D object embedding from the basic multi-modal semantic features. Next, in the **Structure-Aware and Invariant Knowledge Learning** (SAIKL) phase, the hypergraph convolution and memory bank are combined to distill the high-order correlation and invariant knowledge from seen categories to generate the aligned embeddings. Aligning the final embedding with the memory bank tries to anticipate the open-set categories, thus transforming the closed-set training into the open-set training. Finally, the aligned embeddings can be used in the retrieval task and other downstream tasks.

4.2 Multi-Modal 3D Object Embedding

To overcome the inherent semantic gap across different multi-modal representations of 3D objects and distill unified 3D object embedding from multi-modal representations, the MM3DOE module is designed here. Specifically, the MM3DOE utilizes auto-encoder to encode the basic feature into a latent code space for each modality. Then, the homology loss pulls the different modality codes together to guarantee that the modalities derived from the same object are closer than other objects' modalities. Besides, the intra-modal and cross-modal reconstruction loss is introduced to decrease the information loss in the compressing process.

4.2.1 Multi-modal Auto-encoders for 3D Objects

Given N 3D objects $\{o_i\}_{i=1}^N$ and M modality-specific feature extractors $\{\mathcal{F}^k\}_{k=1}^M$, the basic feature matrices $\{\mathbf{X}^k\}_{k=1}^M$ can be generated, where $\mathbf{X}^k = \mathcal{F}^k(\{o_i\}_{i=1}^N)$ and $\mathbf{X}^k \in \mathbb{R}^{N\times d_0}$. As shown in Fig. 4, the auto-encoders always compress the input feature from modality m_k into the latent space m_o, *i.e.*, 3D object embedding space, for better representation, which is defined as follows:

$$\begin{cases} \Psi^k := m_k \to m_o \\ \Phi^k := m_o \to m_k \end{cases}, \tag{3}$$

where $\Psi^k(\cdot)$ is the encoder for modality m_k that maps the feature space of modality m_k into the 3D object embedding space m_o, and $\Phi^k(\cdot)$ is the decoder that maps the features from 3D object embedding space m_o to the modality m_k. Given the basic feature $x_i^k \in \mathbb{R}^{d_0}$ for the 3D object o_i with modality m_k, the estimated 3D object embedding from modality m_k can be denoted as $c_i^k = \Psi^k(x_i^k)$, $c_i^k \in \mathbb{R}^{d_c}$, and the reconstructed modality feature from c_i^k can be denoted as $\hat{x}_i^k = \Phi^k(c_i^k)$, $\hat{x}_i^k \in \mathbb{R}^{d_0}$.

4.2.2 Loss Function for MM3DOE

To compress the modality embedding and leverage the collaborative information across modalities, the Homology Loss $\mathcal{L}_{homo}$ and Bi-reconstruction Loss $\mathcal{L}_{br}$ are developed, respectively.

Homology Loss. The Homology Loss is designed to pull the distance among the estimated 3D object embeddings $\{c_i^k\}_{k=1}^M$ from different modalities closer, which is defined as follows:

$$\mathcal{L}_{homo} = \frac{2}{M(M-1)} \sum_{k=1}^{M} \sum_{l=k+1}^{M} \|c_i^k - c_i^l\|_2, \tag{4}$$

where $\|\cdot\|$ is the $\mathcal{L}_2$ norm, c_i^k and c_i^l are both the estimated 3D object embeddings but from different modalities. The $\mathcal{L}_{homo}$ can restrict the modalities representations of the same 3D object as close as possible to each other to construct a compact latent space.

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

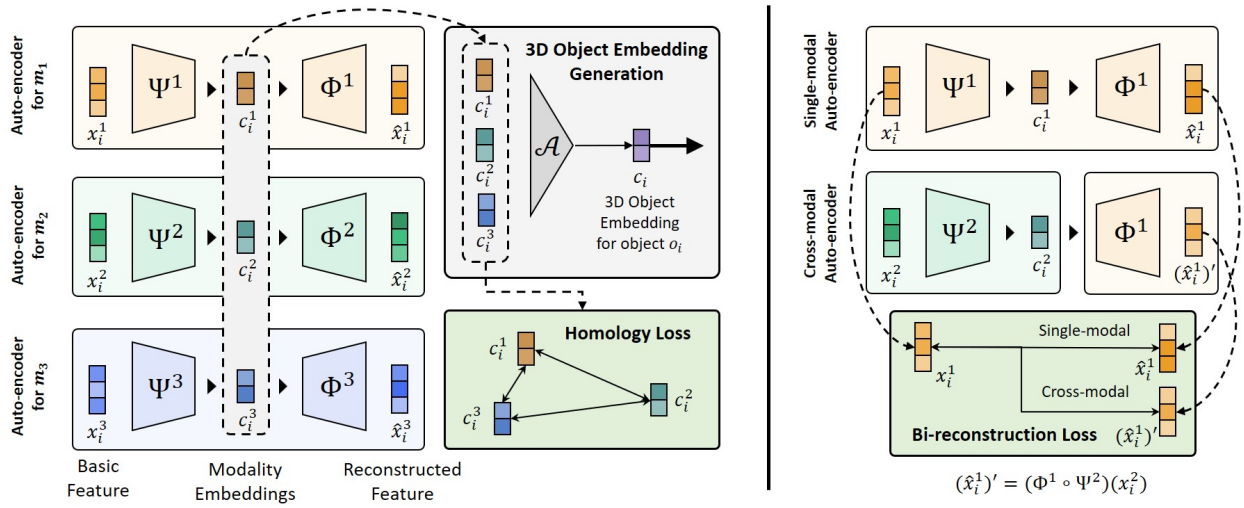


Fig. 4. The illustration of the Multi-modal 3D Object Embedding Module. The 3D object embedding c_i is generated by aggregating modality embeddings $\{c_i^k\}_{k=1}^M$, which can be used in many downstream tasks. The homology loss is designed to learn the shared information across modalities, and the Bi-reconstruction loss is devised to promote the generalization ability of embeddings from each modality.

Bi-reconstruction Loss. To promote the generalization ability of the encoder $\{\Psi^k(\cdot)\}_{k=1}^M$ and decoder $\{\Phi^k(\cdot)\}_{k=1}^M$ for the different modalities of 3D objects, we further propose the Bi-reconstruction Loss. Motivated by [2], the $\mathcal{L}_{br}$ is composed of the single-modal reconstruction restriction and the cross-modal reconstruction restriction, which is formulated as follows:

$$\mathcal{L}_{br} = \frac{1}{M}\sum_{k=1,l\neq k}^{M}\Big(\tau\|x_i^k - \hat{x}_i^k\|_2 + (1-\tau)\|x_i^k - (\Phi^k \circ \Psi^l)(x_i^l)\|_2\Big), \quad (5)$$

where x_i^k and $\hat{x}_i^k$ are the basic feature and reconstructed feature of modality m_k for object o_i, respectively. $f \circ g$ denotes the "function composition" of function $f(\cdot)$ and function $g(\cdot)$. In Eq (5), the former term is designed for the single-modal reconstruction, and the latter one is for the cross-modal reconstruction, which is restricted by $l \neq k$. τ is the hyper-parameter to balance the strengths of the reconstruction ability and generalization ability of the auto-encoders.

Joint Optimization. By combining Eq. (4) and Eq. (5), the overall loss function for multi-modal 3D object embedding is given as:

$$\mathcal{L}_{ae} = \lambda\mathcal{L}_{homo} + (1-\lambda)\mathcal{L}_{br}, \quad (6)$$

where λ is the hyper-parameter to trade-off between the homology loss and the bi-reconstruction loss.

4.2.3 3D Object Embedding Generation

After previous processes, M 3D object embeddings $\{c_i^k\}_{k=1}^M$ estimated from different modalities are generated for 3D object o_i. Since those embeddings have been enhanced and mapped into the same latent space by $\mathcal{L}_{homo}$, an aggregation function $\mathcal{A}(\cdot)$ is proposed to generate the unified 3D object embedding, which can be formulated as:

$$\mathbf{C} = \mathcal{A}\left(\{\mathbf{C}^k\}_{k=1}^M\right), \quad (7)$$

where $\mathbf{C}^k \in \mathbb{R}^{N\times d_c}$, and $c_i^k \in \mathbb{R}^{d_c}$ is the i-th row of the matrix $\mathbf{C}^k$. $\mathbf{C} \in \mathbb{R}^{N\times d_c}$ is the generated unified 3D object embedding matrix, which integrates the information from all modalities.

4.3 Structure-Aware and Invariant Knowledge Learning

To endow the 3D object embedding with the generalization that anticipates unseen categories, we proposed the Structure-Aware and Invariant Knowledge Learning (SAIKL) module as shown in Fig. 5. Specifically, the hypergraph structure and hypergraph convolution are employed to utilize the collaborative high-order information among seen and unseen 3D objects, and the memory bank is able to generalize the invariant knowledge from seen categories to achieve unbiased feature generation for unseen categories.

4.3.1 Hypergraph Generation and Hypergraph Convolution

Although objects from these unseen categories are unlabeled, the potential information among them can also increase the generalization of the 3D object embeddings. To bridge the correlation among 3D objects from seen categories and unseen categories, the hypergraph structure is adopted here. The degree-free hyperedge in a hypergraph can naturally model the high-order correlation among vertex compared with the edge in a simple graph.

A hypergraph can be represented as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$, where $\mathcal{V}$ and $\mathcal{E}$ are the vertex set and the hyperedge set, respectively. $\mathbf{W} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ is a diagonal matrix, where $\mathbf{W}_{i,i}$ denotes the weight of the i-th hyperedge. Here, each 3D object with multi-modal representation is treated as the vertex. Through the MM3DOE module, the unified 3D object embedding matrix $\mathbf{C} \in \mathbb{R}^{N\times d_c}$ is generated, which is treated as the feature associated with each vertex. Then, we construct the hyperedges via the k-nearest neighbors (KNN) algorithm to model the collaborative high-order information among 3D objects. Specifically, for each vertex, we construct a hyperedge to link it and its $K-1$ neighbor vertices. In this way, N hyperedges can be constructed.

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

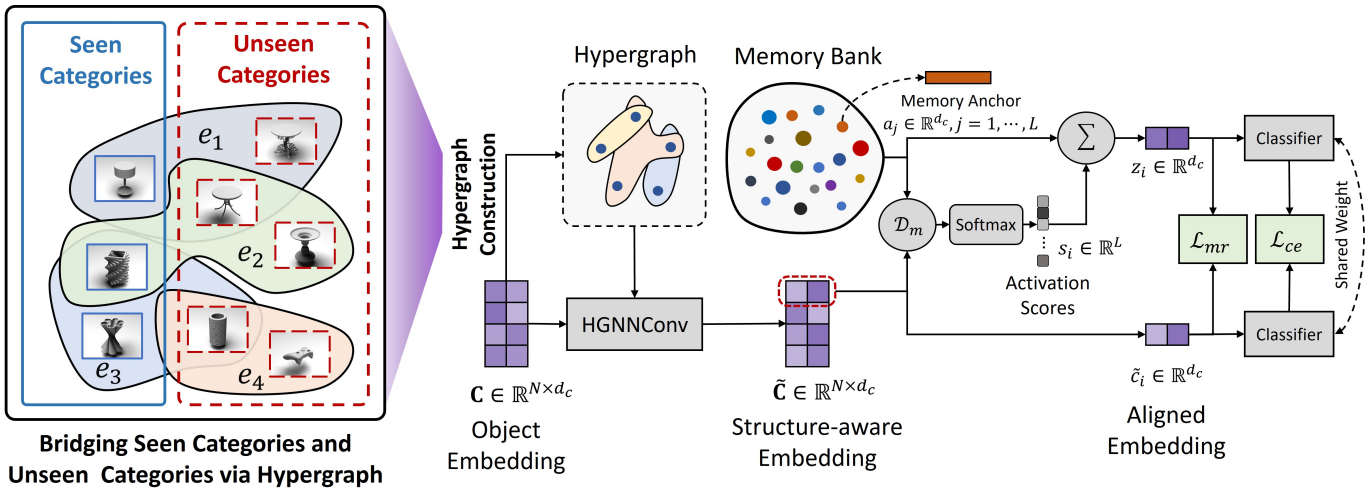


Fig. 5. Illustration of the Structure-Aware and Invariant Knowledge Learning module. The hypergraph is adopted to capture the latent high-order correlation among 3D objects from seen and unseen categories. The memory anchor is developed to improve the generalized ability of the multi-modal 3D object embeddings and resists the over-fitting of learning on seen categories.

For the convenience of computation, the hypergraph can be represented by the incidence matrix $\mathbf{H} \in \{0,1\}^{|\mathcal{V}| \times |\mathcal{E}|}$, where the i-th hyperedge is the i-th column of $\mathbf{H}$, and $\mathbf{H}(v,e) = 1$ if the hyperedge e contains the vertex v. To learn the structure-aware embedding $\tilde{\mathbf{C}} \in \mathbb{R}^{N \times d_c}$ from the hypergraph structure, the hypergraph convolution [4] (HGNNConv) is adopted, which is formulated as follows:

$$\tilde{\mathbf{C}} = \sigma\left(\mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-\frac{1}{2}}\mathbf{C}\mathbf{\Theta}\right), \tag{8}$$

where $\mathbf{D}_v$ and $\mathbf{D}_e$ are the diagonal degree matrices for vertex and hyperedge, respectively. $\mathbf{\Theta} \in \mathbb{R}^{d_c \times d_c}$ is the trainable parameter for the HGNNConv layer. By conducting HGNNConv on the hypergraph constructed with all 3D objects, the generated structure-aware embeddings $\tilde{\mathbf{C}}$ can learn the potential collaborative information from both seen and unseen categories.

4.3.2 Aligning Embeddings to Memory Bank

Except for bridging the correlations among 3D objects from seen categories and unseen categories with hypergraph structure, we also attempt to distill some typical representations from 3D objects from seen categories that are shared by both seen categories and unseen categories. To learn the invariant information, we leverage the memory bank to store a large amount of typical representations during the training stage. The memory bank $\mathcal{M}$ is the structure that contains L invariant memory anchors a_j of 3D objects, and each anchor can store one typical representation of 3D objects, which can be formulated as:

$$\mathcal{M} = \{a_j \in \mathbb{R}^{d_c} \mid j = 1, \cdots, L\}. \tag{9}$$

Given the structure-aware embedding $\tilde{c}_i$ of the 3D object o_i, we first compute the activation score $s_{i,j}$ for each memory anchor a_j in the memory bank. The activation score denotes the similarity between the original 3D object embedding and different typical representations in the memory bank, which can be formulated as follows:

$$s_{i,j} = \mathcal{D}_m(\tilde{c}_i, a_j), \tag{10}$$

where $\mathcal{D}_m(\cdot,\cdot)$ is the distance metric function. Then, we normalize the L activation score $s_{i,j}$ with the softmax function, and rebuilt the object embedding with the normalized activation score $s'_{i,j}$ and all memory anchors $\{a_j\}_{j=1}^L$ in the memory band, which can be formulated as follows:

$$\begin{cases} s'_{i,j} = \dfrac{e^{s_{i,j}}}{\sum_{j=1}^{L} e^{s_{i,j}}} \\ z_i = \sum\limits_{j=1}^{L} s'_{i,j} a_j \end{cases}, \tag{11}$$

where $z_i \in \mathbb{R}^{d_c}$ is the aligned embedding for 3D object o_i. In Sec. D, we provide the visualization of the memory anchors, which demonstrates that the memory bank can store the invariant and meaningful semantic knowledge of 3D objects. Compared with the original 3D object embeddings, aligning embeddings to the typical memory anchors can alleviate the problem of over-fitting and learning the invariant knowledge for the open-set setting.

4.3.3 Loss Function of SAIKL

To train the hypergraph convolution and the learnable memory anchors, we adopt two loss functions, *i.e.*, Memory Reconstruction Loss $\mathcal{L}_{mr}$ and the common Cross-entropy Loss $\mathcal{L}_{ce}$.

Memory Reconstruction Loss. The memory reconstruction can not only keep the similarity between the original 3D object embedding and the aligned embedding but also update the related memory anchors to distill the invariant and typical knowledge toward generalized 3D object representations, which can be formulated as follows:

$$\mathcal{L}_{mr} = \left\|\tilde{c}_i - z_i\right\|_2. \tag{12}$$

Cross-Entropy Loss. To guide the learning of 3D object embedding upon the seen categories, the common cross-entropy loss function is adopted, which can be defined as follows:

$$\mathcal{L}_{ce} = -\sum_{k=1}^{Y}\Big(y_{i,k}\log(p_{i,k}) + y_{i,k}\log(\tilde{p}_{i,k})\Big), \tag{13}$$

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

where $p_{i,k} = \frac{e^{z_{i,k}}}{\sum_{m=1}^{Y} e^{z_{i,m}}}$ and $\tilde{p}_{i,k} = \frac{e^{\tilde{c}_{i,k}}}{\sum_{m=1}^{Y} e^{\tilde{c}_{i,m}}}$ are the predicted probability score of the given 3D object o_i in k-th seen category for the aligned embedding z_i and the structure aware embedding $\tilde{c}_i$, respectively. $y_{i,k}$ is the k-th value of the one-hot encoded ground truth label of the 3D object o_i, and Y is the number of the seen categories.

Joint Optimization. In structure-aware and invariant knowledge learning stage, the combined loss function of $\mathcal{L}_{mr}$ and $\mathcal{L}_{ce}$ is given as:

$$\mathcal{L}_{mb} = \alpha \mathcal{L}_{mr} + (1-\alpha)\mathcal{L}_{ce}, \tag{14}$$

where α is the hyper-parameter for balance.

4.4 Why Hypergraph, Not Graph?

In this subsection, we further mathematically prove that hypergraph modeling has better representative capability on data correlation than graph modeling. We can define a map from hypergraphs to graphs as $\phi : \mathcal{H}^n \rightarrow \mathcal{G}^n$, where the $\mathcal{H}^n$ and $\mathcal{G}^n$ are the hypergraph space and graph space with specified n vertices, respectively. Since a graph can be regarded as a 2-uniform hypergraph, we have $\mathcal{G}^n \subset \mathcal{H}^n$. In this paper, we aim to model the k-neighbor correlation for each 3D object. For a fair comparison, for graph-based modeling, we connect every pair of the k-neighbor of a given 3D object, and for hypergraph-based modeling, we directly utilize a hyperedge to connect the 3D objects from the k-neighbor of a given 3D object. Based on this, we can implement the map as follows:

$$\phi(\mathbf{H}) = \begin{cases} \mathbf{H} & \text{if} \quad \mathbf{H} \in \mathcal{G}^n \\ \mathbf{H}\mathbf{H}^\top & \text{otherwise} \end{cases}, \tag{15}$$

where $\mathbf{H} \in \mathbb{N}^{n \times m}$ is the incidence matrix of an arbitrary n-vertex hypergraph.

Theorem 1. *Given an n-vertex hypergraph and the structure mapping function $\phi : \mathcal{H}^n \rightarrow \mathcal{G}^n$, then the projection ϕ from the hypergraph to graph is non-injection and surjection.*

Proof. For the surjection, we know that any graph can be embedded in the hypergraph space. That is, any graph can be regarded as a special hypergraph with the hyperedge connecting at most two vertices. As for the non-injection, we find a case that the mapping is a many-to-one projection. As shown in Fig. 6, given the graph with adjacency matrix $\mathbf{A} \in \mathbb{N}^{n \times n}$, we have two inverse images $\mathbf{H}$ and $\mathbf{H}'$ such that $\mathbf{A} = \mathbf{H}\mathbf{H}^\top = \mathbf{H}'\mathbf{H}'^\top$. It indicates that the mapping ϕ is a non-injection. □

Remark. *Supposing the number of vertices is fixed to n, graph-based modeling will lose information compared to hypergraph-based modeling.*

The correlation among 3D objects is inherently high-order since more than two 3D objects may share the same property. Considering the graph space $\mathcal{G}^n$ is a sub-space of the hypergraph space $\mathcal{H}^n$ for a specific vertex number n. Graph-based modeling may confuse those similar hypergraphs, as shown in Fig. 6. Therefore, graph-based modeling may lose information, and hypergraph-based modeling is more powerful.

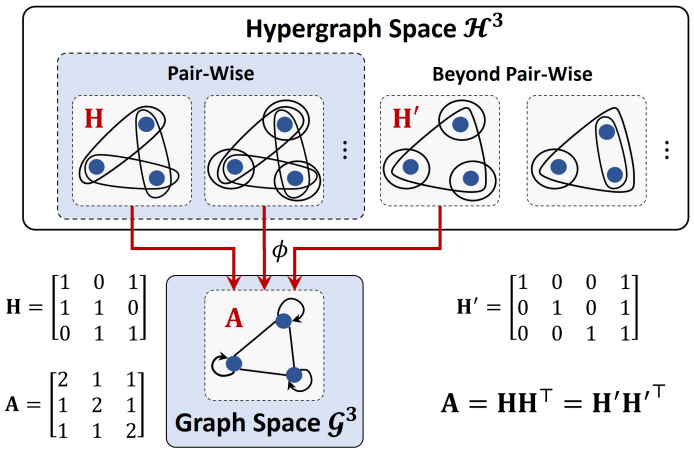


Fig. 6. The comparison of Hypergraph Space $\mathcal{H}^3$ and Graph Space $\mathcal{G}^3$.

5 Experiments

In this section, we first introduce the experimental settings. Then, we conduct experiments on our generated open-set 3DOR datasets, including OS-ESB-core, OS-NTU-core, OS-MN40-core, and OS-ABO-core. Then, we provide various experiments, including separate retrieval on seen and unseen categories, experiments with different input data, experiments under different category splittings, experiments of cross-dataset retrieval, and experiments of real and virtual 3D worlds to verify the effectiveness of the proposed method under different cases. Moreover, ablation studies on the proposed two core components, *i.e.*, the Multi-Modal 3D Object Embedding module and the Structure-aware and Invariant Knowledge Learning module, are presented. Finally, the visualizations on the memory bank and the open-set 3DOR are also provided.

5.1 Experimentalal Settings

Open-Set 3DOR Dataset. We generate four multi-modal open-set 3DOR datasets, including OS-ESB-core, OS-NTU-core, OS-MN40-core, and OS-ABO-core, which are created from the public datasets ESB [5], NTU [6], ModelNet40 [7], and ABO [8], respectively. As shown in Tab. 1, these datasets consist of seen categories for training and unseen categories for retrieval, and three modalities, including multi-view, voxel, and point cloud, are extracted for each 3D object in the three datasets. The generation of open-set 3DOR datasets is provided in Appendix A. Specifically, the detailed descriptions of the multi-modal data generation are shown in Appendix A.2, and the corresponding open-set setting of the four datasets are provided in Appendix A.3.

Multi-Modal Feature Extraction. In our experiments, three modality representations, including multi-view (12 views), point cloud (1024 points), and voxel (32 dimensions), are adopted for a given 3D object. Then, the corresponding feature extraction model: MVCNN [9] for multi-view modality, PointNet [21] for point cloud modality, and 3D Shapenets [25] for voxel modality are employed, respectively. Besides, a simple auto-encoder is applied to avoid the curse of dimensionality in the pre-processing for voxel modality. For more details, refer to Appendix B.

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

TABLE 1
The statistics of the released datasets.

		OS-ESB-core	OS-NTU-core	OS-MN40-core	OS-ABO-core
Categories	Categories	41	67	40	21
	Seen Categories	17	13	8	4
	Unseen Categories	24	54	32	17
Number of Objects	3D Objects	670	1919	12310	6622
	Average Number of Objects per Category	16	28	307	315
Multi-Modal Data	Voxel Data	32-d	32-d	32-d	32-d
	Multi-view Data	24 views	24 views	24 views	24 views
	Point Cloud Data	1024 points	1024 points	1024 points	1024 points
	Real-world Image	False	False	False	True
Retrieval Data Splitting	Number of Objects for Training (OT)	98	378	2821	1082
	Number of Objects for Retrieval (OR)	572	1541	9489	5540
	Number of Query Objects in OR	120	270	160	85
	Number of Target Objects in OR	452	1271	9329	5455

"32-d" denotes generating voxel of $32 \times 32 \times 32$ dimensions.

TABLE 2
Experimental settings on four datasets.

	OS-ESB-core	OS-NTU-core	OS-MN40-core	OS-ABO-core
Dimension of the Multi-view Feature	512	512	512	512
Dimension of the Point Cloud Feature	512	512	512	512
Dimension of the Original Voxel Feature	6912	6912	6912	6912
Dimension of the Compressed Voxel Feature	512	512	512	512
"K" of KNN for Hypergraph	10	10	50	50

Hypergraph Construction. Given the absence of inherent connection structure among 3D objects from both the seen and unseen categories, the widely used K-nearest-neighbor algorithm is adopted to estimate high-order correlations. We employ the grid search to find the most appropriate K for different datasets, as shown in Tab. 2. The time complexity of constructing a hypergraph is $O(N^2C)$, which is independent of "K". In practice, given feature matrix $X \in \mathbb{R}^{N \times C}$ and the hyper-parameter "K", the hypergraph is constructed by the K-nearest-neighbor algorithm, which can be further divided into three steps: computing the distance matrix, sorting neighbors and building hypergraph incidence matrix. Assuming the inner product is adopted for measuring the distance between two objects, the time complexity of computing the distance matrix is $O(N^2C)$. The time complexity of sorting neighbors is $O(N^2 \log(N))$. In most cases, we have $\log(N) \ll C$. The third step is building the hypergraph incidence matrix according to the nearest neighbors, and the time complexity of it is $O(NK)$. In summary, the time complexity of building a hypergraph is $O(N^2C)$.

Training Strategy. For a fair comparison, we fix the random seed as 2022 for all experiments in this paper. The SGD optimizer is adopted for all methods. The learning rate is set to 0.1 and 0.001 for step 1 and step 2, respectively. Besides, the epoch number and the weight decay are set to 120 and 0.0005, respectively. Step 1 and step 2 are trained separately. As for compared methods, the reported hyper-parameter configuration in their paper/code is adopted.

Other Settings. In our experiments, the memory bank contains 128 memory anchors with dimension $a_j \in \mathbb{R}^{256}$, and the 3D object embedding, structure-aware embedding, and aligned embedding all are vectors with 256 dimensions. Each memory anchor is initialized with the normal distribution and can be trained in the data forward and backward processes. The 3D object embedding generation function $\mathcal{A}(\cdot)$ in MM3DOE is implemented by the average pooling function. The hyper-parameters τ, λ, and α in Eq. (5), Eq. (6), and Eq. (14) are set to 0.5, 0.6 and, 0.1, respectively. As for the evaluation metric, the commonly used mAP, NDCG, ANMRR, and Precision-Recall Curve are adopted.

5.2 Compared Methods

Since there is no method specifically designed for the open-set 3DOR task yet, we re-produce the current state-of-the-art methods of the 3DOR task and the open-set recognition task for comparison. The compared methods can be divided into two categories, *i.e.*, the traditional 3DOR methods (MMJM [35], TCL [34], SDML [39], and CMCL [40]) and knowledge-based methods (MMSAE [41] and PROSER [42]):

MMJM [35]: MMJM is a multi-modal joint network, which adopts weighted fusion to combine multi-modal features for learning.

TCL [34]: TCL is a typical metric-learning-based method, which incorporates the triplet loss and center loss to learn discriminative 3D object embeddings.

SDML [39]: SDML is a metric-learning-based cross-modal retrieval model, which trains the models for different modalities independently and learns different projection matrices to bridge different modalities.

CMCL [40]: CMCL designs the cross-modal center loss to pull modality representations from the same categories together and push those from the different categories apart, which is also a metric-learning-based method.

MMSAE [41]: MMSAE is an auto-encoder-based cross-modal retrieval method, which trains auto-encoders with

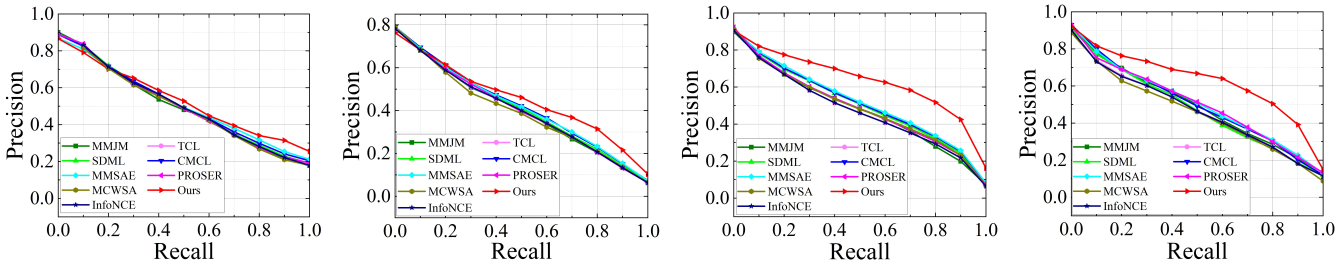
Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

TABLE 3
Experimental results on the OS-ESB-core, OS-NTU-core, OS-MN40-core, and OS-ABO-core datasets.

	OS-ESB-core			OS-NTU-core			OS-MN40-core			OS-ABO-core		
	mAP↑	NDCG↑	ANMRR↓	mAP↑	NDCG↑	ANMRR↓	mAP↑	NDCG↑	ANMRR↓	mAP↑	NDCG↑	ANMRR↓
TCL	0.4931	0.2189	0.5268	0.3937	0.2123	0.6100	0.4811	0.6383	0.52301	0.4933	0.5386	0.5105
MMJM	0.4894	0.2174	0.5299	0.3924	0.2121	0.6122	0.4736	0.6329	0.5303	0.4701	0.5218	0.5324
SDML	0.4959	0.2175	0.5236	0.4016	0.2152	0.6049	0.5075	0.6570	0.5022	0.4744	0.5279	0.5242
CMCL	0.5001	0.2197	0.5306	0.4108	0.2172	0.5943	0.5138	0.6598	0.4975	0.4983	0.5089	0.5024
MMSAE	0.4988	0.2206	0.5369	0.4085	0.2170	0.5999	0.5208	0.6657	0.4900	0.5051	0.5380	0.5049
MCWSA	0.4948	0.2134	0.5375	0.3922	0.2069	0.6214	0.4878	0.6385	0.5195	0.4561	0.5105	0.5470
PROSER	0.4869	0.2113	0.5395	0.3947	0.2124	0.6096	0.4900	0.6454	0.5166	0.5033	0.5327	0.5034
InfoNCE	0.5026	0.2191	0.5263	0.4003	0.2119	0.6109	0.4737	0.6331	0.5302	0.4683	0.5214	0.5350
Ours	**0.5174**	**0.2273**	**0.5128**	**0.4488**	**0.2281**	**0.5667**	**0.6420**	**0.7291**	**0.3827**	**0.6339**	**0.5796**	**0.3796**

For the mAP and NDCG metric, the higher score is better. For the ANMRR metric, the lower score is better.



(a) PR Curve on OS-ESB-core. (b) PR Curve on OS-NTU-core. (c) PR Curve on OS-MN40-core. (d) PR Curve on OS-ABO-core.

Fig. 7. The precision-recall curves of the proposed method and compared methods on four datasets, respectively.

reconstruction loss function to project embeddings from different modalities into the same latent space.

MCWSA [44]: MCWSA is a multi-channel weight-sharing auto-encoder method that fuses modalities with the multi-head attention mechanism for multi-modal data representation learning.

PROSER [42]: PROSER is a typical open-set recognition method, which expands the closed-set classifier to detect whether the sample belongs to the seen categories or not.

InfoNCE [45]: InfoNCE is a widely used contrastive learning method for general representation learning. It induces the latent space to capture information that is highly informative for predicting future samples and improving the embedded representation.

For a fair comparison, we fix the parameters of the modality-specific feature extractor after pre-training in the training set. That is, the same basic features for all modalities are used for experiments for all compared methods and the proposed HGM^{2}R . Besides, to ensure fairness in comparison to methods such as TCL [34], InfoNCE [45], and PROSER [42], which solely utilize single-modality data as input, we incorporate a tiny multi-modal fusion network as the former network to guarantee the same multi-modal input as other compared methods.

5.3 Results and Discussions

To demonstrate the effectiveness of the proposed HGM^{2}R framework, we compare it with other state-of-the-art methods. Experimental results of the open-world 3DOR task are shown in Tab. 3. As shown in the results, the proposed method outperforms the state-of-the-art methods on all four datasets. In particular, on the OS-MN40-core and OS-ABO-core dataset, our method achieves $0.6420/0.6339$ mAP with about $12\%/13\%$ improvements compared with the second-best method. We also provide the Precision-Recall (PR) Curve to evaluate the performance of the proposed HGM^{2}R framework and other compared methods, as illustrated in Fig. 7. The larger area below the curve indicates better performance. From the results, we can observe that our method outperforms all other compared methods. The better performance indicates that by the structure-aware and invariant knowledge learning, the proposed method has the potential to understand the generalized 3D object representation and achieve better open-set retrieval performance as more 3D object data is fed.

Besides, as shown in those results from Tab. 3 and Fig. 7 that the performance improvements achieved on the OS-MN40-core dataset and OS-ABO-core dataset are more significant than that on the OS-ESB-core and OS-NTU-core dataset. Specifically, as the average number of objects per category arises (16 of OS-ESB-core $\rightarrow$ 28 of OS-NTU-core $\rightarrow$ 307 of OS-MN40-core $\rightarrow$ 315 of OS-ABO-core), the corresponding performance will significantly increase. This is because, compared with the OS-ESB-core and OS-NTU-core datasets, the OS-MN40-core and OS-ABO-core datasets have more than ten times samples per category for the model to train, in which the proposed SAIKL module can aggregate more high-order correlations from similar 3D objects, and store more invariant knowledge in memory anchors.

Additionally, we observe that, in traditional 3DOR methods, CMCL, SDML, and TCL perform better than the MMJM. The result indicates that metric learning is more suitable for learning discriminative embeddings compared with direct modality concatenation. In contrast, MMJM simply adopts the weighted concatenation to fuse different modalities, which ignores the modality gap and cannot

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

TABLE 4
Results of separate retrieval on seen and unseen categories.

	On Seen Categories		On Unseen Categories	
	mAP↑	Recall@100↑	mAP↑	Recall@100↑
TCL	0.9350	0.8214	0.7392	0.7176
MMJM	0.9199	0.8078	0.7307	0.7138
SDML	0.8850	0.7850	0.7469	0.7239
CMCL	0.9099	0.7960	0.7521	0.7249
MMSAE	0.8872	0.7861	0.7603	0.7294
MCWSA	0.8570	0.7683	0.7289	0.7056
PROSER	0.8771	0.7778	0.7493	0.7256
InfoNCE	0.9365	0.8219	0.7392	0.7164
Ours	**0.9410**	**0.8247**	**0.8223**	**0.7821**

TABLE 5
Experimental results with different input data.

	Multi-view		Point Cloud		Voxel	
	4-v	12-v	1024-p	2048-p	32-d	64-d
MMJM	0.4681	**0.4811**	0.4811	**0.4863**	**0.4811**	0.4809
TCL	0.4646	**0.4736**	0.4736	**0.4864**	0.4736	**0.4738**
SDML	0.4960	**0.5075**	0.5075	**0.5157**	**0.5075**	0.5067
CMCL	0.5006	**0.5138**	0.5138	**0.5243**	0.5138	**0.5132**
MMSAE	0.5085	**0.5209**	0.5209	**0.5289**	**0.5209**	0.5199
MCWSA	0.4728	**0.4878**	0.4878	**0.4911**	**0.4878**	0.4738
PROSER	0.4845	**0.4900**	0.4900	**0.4955**	0.4900	**0.4902**
InfoNCE	0.4646	**0.4737**	0.4737	**0.4865**	0.4737	**0.4739**
Ours	0.6336	**0.6420**	0.6420	**0.6439**	0.6420	**0.6434**

"4-v" and "12-v" denote capturing $4/12$ horizontal views from the 3D object. "1024-p" and "2048-p" denote sampling $1024/2048$ points from the surface of a given 3D object. "32-d" and "64-d" denote generating voxel of $32 \times 32 \times 32/64 \times 64 \times 64$ dimensions.
Note that this table only provides the mAP score for comparison.

utilize the collaborative information across modalities, thus performing the worst. Moreover, the superior performance of CMCL and SDML compared to TCL can be attributed to the adaptability of their loss functions in the multi-modal setting. In knowledge-based methods, the same reason enables MMSAE to perform better than PROSER. As for the auto-encoder-based methods, the MCWSA performs worse than the MMSAE, which indicates that the multi-head attention cannot yield a more general representation confronting objects from unseen categories.

5.4 Separate Retrieval on Seen and Unseen Categories

In the open-world retrieval setting, it is essential to consider both seen and unseen categories separately. In this subsection, we conduct experiments to investigate whether our method can reduce the gap in retrieving 3D objects from seen and unseen categories separately, as depicted in Fig. 1. We first split the categories of ModelNet40 datasets into two sets: D_S (for seen categories) and D_U (for unseen categories), as described in Tab. S3. Each set contains 20 categories, and 3D objects in each category are further divided into training set D_S^{tr}/D_U^{tr} (80%) and retrieval set D_S^{re}/D_U^{re} (20%). In this setting, the models are trained on the D_S^{tr}, and evaluated on the seen categories D_S^{re} and unseen categories D_U^{re} separately, as shown in Tab. 4.

In Tab. 4, we observe a significant gap between the results for "On Seen Categories" and "On Unseen Categories", with a decline of about 0.16 *w.r.t.* mAP on average. This is because of the inherent challenge of retrieval confronting 3D objects from unseen categories. Despite the superior performance of TCL, MMJM, and InfoNCE on seen categories, they also have apparent performance degradation when facing unseen categories. In contrast, our method achieves the best performance in retrieval on both seen and unseen categories. In particular, on unseen categories, our method exhibits a gain of about 0.09 in mAP. It obviously reduces the gap in retrieving 3D objects from seen and unseen categories separately. We attribute the superior performance of our method to hypergraph-based 3D embedding learning. By modeling high-order correlations among 3D object embeddings, the hypergraph can bridge the 3D objects from seen categories and unseen categories towards a universal 3D object correlation modeling. Besides, the invariant knowledge learning module can further help extract the general representation of 3D objects and resist over-fitting on seen categories.

5.5 Experiments with Different Data Generation

In this subsection, experiments on the OS-MN40-core dataset with different data generation strategies are provided to verify the consistent effectiveness of the proposed HGM^{2}R . Specifically, for each setting, we change the data generation strategy of one modality while keeping that of the other two modalities the same as the main experiment. The data generation strategy is altered as follows: the number of input views is set to 4 or 12, the number of points sampled from the surface is set to 1024 or 2048, and the resolution of the voxel data is set to $32 \times 32 \times 32$ or $64 \times 64 \times 64$. For example, in the "4-v" setting, the multi-modal input is configured as 4 views for multi-view modality, 1024 points for point cloud modality, and $32 \times 32 \times 32$ for voxel modality. Note that the "4-v" setting for multi-view modality is generated by capturing views from 4 cameras with 90-degree intervals from each other. Other settings are the same as the main experiment in Sec. 5.1.

The experimental results are shown in Tab. 5. For the multi-view modality, as we decrease the number of views from 12 to 4, the performance of all methods decreases, and our method still shows the most superior performance with a large margin from the second-best methods. A similar trend is also shown in the point cloud modality. However, the increment of resolution of the voxel modality cannot lead to robust performance improvement. The main reason is the limited expressive ability to represent a 3D object with a three-dimensional binary tensor.

5.6 Experiments under Different Category Splittings

In this subsection, we conduct experiments under different category splittings on the OS-MN40-core dataset. The traditional retrieval task uses the same categories in the training stage and retrieval stage and splits the dataset by selecting objects from each category for training and retrieval. Unlike the traditional retrieval task, the core motivation of the Open-Set 3DOR task is to simulate the case of existing unseen categories in real-world retrieval. Thus, the category splitting for training and retrieval is very important. Here, we adopt two strategies to investigate how the category splitting for the training set and retrieval set affects performance: splitting under coarse classes and under random

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

categories splitting. Detailed training/retrieval splitting can be found in Appendix C.

TABLE 6
The description of eight coarse classes of the OS-MN40-core dataset.

Coarse Classes	Contained Categories
Items supporting people	table, stool, bench, bed, chair, desk
Square items supporting somethings	night stand, mantel, tv stand, dresser, bookshelf
Square items containing somethings	glass box, wardrobe, range hood, tent
Items like cylinder/cone	flower pot, plant, vase, bottle, cone
Flat items	keyboard, guitar, laptop, curtain, door
Electronic products	monitor, laptop, radio, xbox
Items containing water	sink, toilet, bathtub, bowl, cup
Others	airplane, car, person, sofa, stairs, lamp

In our original splitting, we manually split 40 categories into eight coarse classes, as shown in Tab. 6. On the original OS-MN40-core dataset, the mark "8/32" (Under Coarse Splitting) in Tab. 7 means that one category from each coarse class is selected for training, and the rest is used for retrieval. Similarly, "16/24" (Under Coarse Splitting) is the setting where two categories of each coarse class are selected for training and the rest is used for retrieval. "24/16" (Under Coarse Splitting) is similar to the former two splittings. Besides, to eliminate the effect of manual coarse splitting, we further conduct control experiments by randomly splitting. "8/32" (Randomly Splitting) in Tab. 7 is the setting where 8 categories are randomly selected for training and the rest is selected for retrieval. Similarly, "16/24" and "24/16" are generated by the same random splitting.

TABLE 7
Experimental results of models under different category splittings on the OS-MN40-core dataset.

	Under Coarse Splitting			Under Random Splitting		
Settings	**8/32**	**16/24**	**24/16**	**8/32**	**16/24**	**24/16**
MMJM	0.4811	0.5519	0.6569	0.5097	0.6324	0.6794
TCL	0.4736	0.5442	0.6256	0.5251	0.6320	0.6700
SDML	0.5075	0.5586	0.6449	0.5532	0.6580	0.6997
CMCL	0.5138	0.5778	0.6674	0.5702	0.6680	0.7091
MMSAE	0.5209	0.5982	0.6893	0.5766	0.6734	0.7054
MCWSA	0.4878	0.5508	0.6279	0.5594	0.6199	0.6547
PROSER	0.4900	0.5513	0.6455	0.5526	0.6410	0.6752
InfoNCE	0.4737	0.5443	0.6249	0.5251	0.6323	0.6699
Ours	**0.6420**	**0.7196**	**0.8555**	**0.7065**	**0.8004**	**0.8462**

The first number in "Settings" is the number of categories used for training and the second number is the number of categories used for retrieval. For example, "8/24" denotes 8 categories are used for training and 24 categories are used for retrieval.
Note that this table only provides the mAP score for comparison.

Experimental results under different splittings of 3DOR on the OS-MN40-core dataset are presented in Tab. 7. First, the proposed HGM^{2}R yields the best performance under all splittings, which demonstrates that HGM^{2}R is robust to category splittings. Both on two splittings, as the number of categories of training arises, the performance also improves, which indicates that seeing more categories in training can generate more expressive embeddings for 3D objects in unseen categories. Besides, we find that with the same number of categories for training/retrieval, the models under random splitting perform better than those under coarse splitting. We attribute this phenomenon to manual coarse splitting. The eight coarse splittings are biased to the categories of real-world 3D objects, which just cluster some categories with similar features. However, these manual features cannot replace the general feature distribution in the real world. In contrast, random splitting provides a more accurate depiction of the actual feature distribution in the real world, leading to better performance.

5.7 Experiments of Cross-Dataset Retrieval

In this subsection, we conduct experiments for cross-dataset retrieval to investigate how the dataset's intrinsic distribution affects the experimental results. Here, the OS-MN40-core dataset is adopted for training, and the OS-ABO-core dataset is adopted for retrieval. As shown in Tab. 1, OS-MN40-core and OS-ABO-core contain 40 and 21 categories, respectively. Besides, we notice that the categories from the two datasets partially overlap, which means 11 categories in the retrieval stage (OS-ABO-core) may have been seen in the training stage (OS-MN40-core). Therefore, we adopt four settings for comparison: **Excluding Shared Categories**, **Sharing Four Categories**, **Sharing Seven Categories**, and **Including Shared Categories**. The retrieval set of the four settings includes 0, 4, 7, and 11 seen categories from the training stage, respectively. Experimental results are shown in Tab. 8. Specifically, in the first experimental setting, all 40 categories of OS-MN40-core are adopted for training, and ten categories (excluding all shared categories with OS-MN40-core) of OS-ABO-core are adopted for retrieval. In the last setting, all 40 categories of OS-MN40-core are adopted for training, and all 21 categories of OS-ABO-core are adopted for retrieval. Detailed categories splitting of the four settings is provided in Appendix Tab. S4, S5 , S6, and S7. Other experimental settings are the same as the main experiments in Sec. 5.1.

Experimental results of cross-dataset 3DOR are shown in Tab. 8. Based on the results, we have the following observations. First, the proposed HGM^{2}R exhibits the best performance under all four settings. Specifically, in the Excluding Shared Categories setting, our method achieves about 10% performance improvements in terms of mAP compared with the strongest baseline. The experimental results demonstrate the significant advantages of our method when confronting a large number of unseen categories during the retrieval stage. We attribute the improvements mainly to the designed structure-aware and invariant knowledge learning module, which could extract more general features for multi-modal 3D representations. Besides, the constructed hypergraph structure can bridge the high-order correlations of 3D objects from both unseen categories and seen categories to achieve robust retrieval performance when confronting unseen categories. Furthermore, as the ratio of unseen categories gradually rises ($48\% \rightarrow 59\% \rightarrow 71\% \rightarrow 100\%$), the performance improvements achieved by our method in comparison to the second-best method also progressively increase ($3\% \rightarrow 4\% \rightarrow 6\% \rightarrow 10\%$). This suggests that our approach effectively addresses the challenge of retrieving objects from previously unseen categories. The progressive performance improvements also demonstrate the efficacy of our method in adapting to and generalizing

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

TABLE 8
Experimental results of cross-dataset 3D object retrieval.

	Excluding Shared Categories			Sharing Four Categories			Sharing Seven Categories			Including Shared Categories		
	mAP↑	NDCG↑	ANMRR↓	mAP↑	NDCG↑	ANMRR↓	mAP↑	NDCG↑	ANMRR↓	mAP↑	NDCG↑	ANMRR↓
MMJM	0.7093	0.5687	0.3218	0.6273	0.6165	0.393	0.5846	0.5209	0.432	0.5442	0.5375	0.4606
TCL	0.6506	0.5405	0.3775	0.5969	0.6033	0.4260	0.5493	0.5059	0.4621	0.5158	0.5276	0.4927
SDML	0.6708	0.5529	0.3622	0.6115	0.6148	0.411	0.5548	0.5108	0.4538	0.5281	0.5352	0.4807
CMCL	0.6804	0.5571	0.3504	0.6183	0.6106	0.4066	0.5732	0.5173	0.4369	0.5390	0.5379	0.4728
MMSAE	0.6803	0.5599	0.3535	0.6064	0.6004	0.4177	0.5582	0.5060	0.4509	0.5283	0.528	0.4802
MCWSA	0.6289	0.5407	0.3935	0.5718	0.5874	0.4430	0.5283	0.4885	0.4724	0.4920	0.5099	0.5111
PROSER	0.6531	0.5409	0.3710	0.5902	0.6028	0.4296	0.5399	0.4998	0.4676	0.5080	0.5237	0.4973
InfoNCE	0.6507	0.5409	0.3776	0.5968	0.6031	0.4268	0.5489	0.5062	0.4627	0.5163	0.5275	0.4916
Ours	**0.8042**	**0.6088**	**0.2352**	**0.6825**	**0.6309**	**0.3644**	**0.6194**	**0.5288**	**0.4038**	**0.5755**	**0.5414**	**0.4535**

In this setting, 3D objects for training are selected from OS-MN40-core dataset, and 3D objects for retrieval are selected from OS-ABO-core dataset.

across diverse and unfamiliar object classes. Since the 3D objects from seen categories may influence the evaluation of learning the general embedding toward unseen categories in the retrieval set, we suggest removing all 3D objects from seen categories to fully evaluate the open-set retrieval performance of a given method. Besides, we notice that the results of open-set retrieval under "Including Shared Categories" seem lower than that under "Excluding Shared Categories". It can be attributed to the differences in the number of categories within the retrieval set. Specifically, under the "Including Shared Categories" setting, the retrieval set contains samples from 21 categories, while under the "Excluding Shared Categories" setting, the retrieval set contains samples from 10 categories. We further split the data of "Including Shared Categories" into two parts: 3D objects from seen categories, and 3D objects from unseen categories. The performance bottleneck of this setting is the second part of the data. The 3D objects from the seen categories are more compact than those objects from unseen categories since those seen categories are well-studied in the training process. Those 3D objects from seen categories will appear as a large clique in the retrieval process of 3D objects from unseen categories. Thus, the experiment of "Including Shared Categories" with more categories for retrieval will yield worse performance than the experiment of "Excluding Shared Categories".

5.8 Experiments of Real and Virtual 3D Worlds

In this subsection, we further test our models on the real-world open-set retrieval task. We conduct experiments across the OS-MN40-core and OS-ABO-core datasets. The OS-MN40-core dataset only contains the rendered images from synthetic 3D objects, and the OS-ABO-core provides both rendered images from synthetic 3D objects and real images of the corresponding products as illustrated in Fig. S8. Then, we adopt two main settings for evaluation: "OS-MN40-core for training and OS-ABO-core for retrieval (OS-MN40-core → OS-ABO-core)", and "OS-ABO-core for training and OS-MN40-core for retrieval (OS-ABO-core → OS-MN40-core)". For each setting, we further divide it into two sub-settings for convenience of comparison, as shown in Tab. 9. Note that "Rendered-I" denotes "Rendered Image", and "Real-I" denotes "Real Image". As for the "Rendered-I → Real-I" of "(OS-MN40-core → OS-ABO-core)", the training set is rendered images of all 3D objects from the OS-MN40-core dataset, and the retrieval set is real images of all products from the OS-ABO-core dataset. As for the "Real-I → Rendered-I" of "(OS-ABO-core → OS-MN40-core)", the training set is real images of all products from the OS-ABO-core dataset, and the retrieval set is rendered images of all 3D objects from the OS-MN40-core dataset. We further conduct two controlled experiments under two main settings, *i.e.*, the "Rendered-I → Rendered-I", where the rendered images from the two datasets are adopted for the training set or retrieval set, and the object splitting is the same as other experiments under the specific setting.

Since the real-world data only contain image modality, we make some modifications to the proposed HGM^{2}R as follows. First, we remove the MM3DOE module, and the image features extracted by the image modality backbone are directly fed into the proposed SAIKL module to learn the general features under the open-set setting. Besides, the number of input views is fixed to 1 since some products in the dataset only contain one real image. Other experimental settings are the same as in the main experiments in Sec. 5.1.

Experimental results are shown in Tab. 9. From the results, we have the following observations. Firstly, the proposed method outperforms all compared methods, which demonstrates the effectiveness of the proposed method in exploiting the unseen categories and in bridging the gap between the virtual and real 3D worlds. Secondly, with the same 3D object as training and retrieval, all methods perform worse across real and rendered image settings (Rendered-I → Real-I and Real-I → Rendered-I) than the controlled experiments (Rendered-I → Rendered-I). This is because of the inherent gap between the virtual world and the real world, and retrieval with virtual 3D objects is not challenging enough to validate the effectiveness of a method. Thirdly, as for the "OS-MN40-core → OS-ABO-core" setting, we notice that our method achieves about 10% improvement in the "Rendered-I → Rendered-I" sub-setting, while in the "Rendered-I → Real-I" sub-setting the gains are about 5%. The results indicate that our method has the potential to handle open-set retrieval among real-world 3D objects. Finally, we find an interesting point: the performance gap between the two sub-settings under the "OS-MN40-core → OS-ABO-core" and "OS-ABO-core → OS-MN40-core" differs. Here, we take our method and the mAP metric as an example. In the "OS-MN40-core → OS-ABO-core" setting, the margin is about 7%, while the margin under the other setting is about 1%. It indicates that the model trained on real-world data can easily adapt to virtual-world data, whereas the model trained on virtual-world data will yield significant performance degradation when tested on real-world data. This can be attributed to the

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

TABLE 9
Experimental results of retrieval across real and virtual 3D worlds.

	OS-MN40-core → OS-ABO-core						OS-ABO-core → OS-MN40-core					
	Rendered-I → Real-I			Rendered-I → Rendered-I			Real-I → Rendered-I			Rendered-I → Rendered-I		
	mAP↑	NDCG↑	ANMRR↓	mAP↑	NDCG↑	ANMRR↓	mAP↑	NDCG↑	ANMRR↓	mAP↑	NDCG↑	ANMRR↓
MMJM	0.3278	0.3891	0.6605	0.4447	0.4575	0.5521	0.3944	0.5603	0.5979	0.4093	0.5760	0.5844
TCL	0.3341	0.3962	0.6563	0.4422	0.4570	0.5549	0.3963	0.5643	0.5964	0.4112	0.5769	0.5821
SDML	0.3460	0.4043	0.6429	0.4582	0.4678	0.5378	0.4042	0.5658	0.5904	0.4216	0.5838	0.5741
CMCL	0.3492	0.4049	0.6406	0.4558	0.4664	0.5413	0.4041	0.5658	0.5906	0.4216	0.5837	0.5741
MMSAE	0.3591	0.4167	0.6307	0.4524	0.4656	0.5479	0.4012	0.5627	0.5924	0.4180	0.5796	0.5764
MCWSA	0.3263	0.3875	0.6616	0.4278	0.4569	0.5670	0.3878	0.5573	0.6046	0.4072	0.5718	0.5856
PROSER	0.3365	0.3988	0.6533	0.4432	0.4625	0.5519	0.3962	0.5638	0.5968	0.4114	0.5773	0.5808
InfoNCE	0.3348	0.3957	0.6561	0.4422	0.4569	0.5549	0.3963	0.5644	0.5966	0.4112	0.5701	0.5823
Ours	**0.4359**	**0.4597**	**0.5642**	**0.5068**	**0.4807**	**0.5030**	**0.4893**	**0.6250**	**0.5148**	**0.4977**	**0.6422**	**0.5044**

"Rendered-I" denotes "Rendered Image", and "Real-I" denotes "Real Image". "A → B" denotes "A" is adopted for training and "B" is adopted for retrieval.

TABLE 10
Ablation studies of the Multi-modal 3D Object Embedding module on the OS-MN40-core dataset.

		mAP↑	NDCG↑	ANMRR↓
Single-Modal	**Multi-view Feature**	0.4920	0.6472	0.5109
	Point Cloud Feature	0.2989	0.4618	0.6785
	Compressed Voxel Feature	0.2784	0.4539	0.6909
Multi-Modal	**Concat**	0.3117	0.4914	0.6656
	MM3DOE w/o $\mathcal{L}_{br}$	0.5041	0.6570	0.5032
	MM3DOE w/o $\mathcal{L}_{homo}$	0.5126	0.6637	0.4942
	MM3DOE	**0.5168**	**0.6678**	**0.4914**

"w/o" denotes "without".

TABLE 11
Ablation studies of the Structure-aware and Invariant Knowledge Learning Module on the OS-MN40-core dataset.

		mAP↑	NDCG↑	ANMRR↓
On SAL	**MLP-based SAL**	0.4717	0.6298	0.5305
	GCN-based SAL	0.5606	0.6854	0.4535
	SAL	**0.6243**	**0.7260**	**0.3925**
On IKL	**IKL w/o $\mathcal{L}_{mr}$**	0.6269	0.7270	0.3899
	IKL w/o $\mathcal{L}_{ce}$	0.6301	0.7278	0.3845
	SAL + IKL	**0.6339**	**0.7291**	**0.3827**

"w/o" denotes "without".

limited representation of information in virtual-world data, while real-world data can reflect most of the properties of the virtual-world data. Therefore, models trained on real-world data possess a more robust ability to generalize and adapt to virtual-world data. Moreover, in the real 3D world, there are more knowledge and high-order correlations compared with the virtual 3D world, which deserves to be exploited in future works.

5.9 Ablation Study

In this subsection, we conduct ablation studies to verify the effectiveness of the proposed Multi-Modal 3D Object Embedding module and the Structure-Aware and Invariant Knowledge Learning module. Besides, the ablation studies on the hyper-parameter "k" in hypergraph construction in the SAL module are also provided.

5.9.1 On the MM3DOE Module

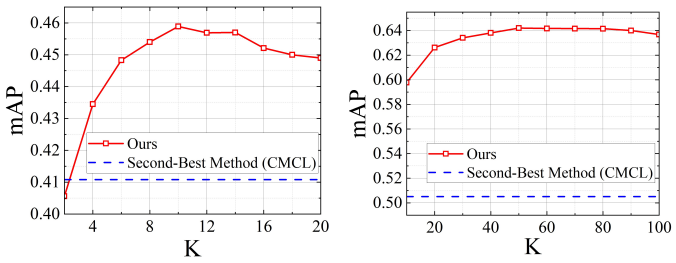
The Multi-modal 3D Object Embedding (MM3DOE) module is designed to overcome the distinction of different modalities and generate unified 3D object embeddings. We consider this ablation from two aspects, *i.e.*, comparison of single-modal features and comparison of multi-modal features, as shown in Tab. 10. It is clear that the proposed MM3DOE outperforms all single-modal features. In the single-modal comparison, basic features from different modalities exhibit different performances. The multi-view feature yields the best performance compared with the other two modalities, which can be attributed to the well-studied 2D convolution on images. As for the multi-modal comparison, we compare the proposed MM3DOE with the **Concat**, MM3DOE without $\mathcal{L}_{br}$, and MM3DOE without $\mathcal{L}_{homo}$. The **Concat** denotes the direct concatenation of multi-modal features. Results show that direct concatenation cannot fully utilize the superiority of multi-modal information and can only achieve borderline performance (not the worst and not the best). MM3DOE with both $\mathcal{L}_{br}$ and $\mathcal{L}_{homo}$ exhibits the best performance, which indicates that the proposed MM3DOE can effectively utilize collaborative information across modalities and overcome the modality distinction.

5.9.2 On the SAIKL Module

To validate the effectiveness of the Structure-aware and Invariant Knowledge Learning (SAIKL) module, we test Structure-aware Learning (SAL) and the Invariant Knowledge Learning (IKL) on the OS-MN40-core dataset, respectively. Note that, in this ablation study, the input of the SAL and its derivatives is the same 3D object embeddings learned from the MM3DOE module, and the input of the IKL and its derivatives is the same structure-aware embeddings learned from the SAL module. In the ablation of the SAL, we replace the hypergraph-based correlation learning with MLP and GCN, where MLP denotes the fully-connected layers and GCN denotes the graph convolution networks [3]. For a fair comparison, the replaced MLP and GCN have the same number of layers and trainable parameters as the original hypergraph-based correlation learning. Besides, the graph is constructed by the KNN algorithm with the same value of K as the hypergraph-based correlation learning. Experimental results are shown in Tab. 11, from which we can observe that the correlation learning on SAL among seen and unseen categories of 3D objects can significantly increase the performance of the

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768



(a) On the OS-NTU-core dataset. (b) On the OS-MN40-core dataset.

Fig. 8. Ablation study on hyper-parameter "K" in hypergraph construction.

TABLE 12
Complexity analysis on four datasets.

		OS-ESB-core	OS-NTU-core	OS-ABO-core	OS-MN40-core
Batch Size		670	1919	6622	12310
Params (M)	Step 1	4.73	4.73	4.73	4.73
	Step 2	0.14	0.14	0.14	0.14
FLOPs (G)	Step 1	4.22	12.07	41.66	77.45
	Step 2	0.09	0.26	0.88	1.68
Inference Time (ms)	Step 1	1.4	2.31	6.14	10.69
	Step 2	0.74	1.35	4.03	8.80

open-set 3DOR. Compared to the edge of the graph, the hyperedge in the hypergraph can model beyond pair-wise correlations, which endows the capability of modeling high-order correlation on the hypergraph. Thus, the proposed hypergraph-based SAL achieves the best performance. In the ablation study of IKL, the IKL without $\mathcal{L}_{ce}$ shows better performance than the IKL without $\mathcal{L}_{mr}$, which implies that the proposed IKL does not heavily rely on the labeled data and has the potential to be expanded to the unsupervised learning paradigm. The combination of SAL and IKL yields the best performance, which demonstrates the proposed SAIKL module can effectively utilize the collaborative high-order information among seen and unseen categories of 3D objects and learn the invariant knowledge to adapt to the open-set setting.

5.9.3 On Hypergraph Construction

From Sec. 5.9.2, we notice that the SAL module contributes the most performance improvements in our method. The core of the SAL module is the hypergraph construction with K-nearest-neighbor (KNN) algorithm. Thus, we further conduct ablation studies on the hyper-parameters "K" to validate the influence of it on hypergraph construction in the SAL Module. Obviously, as shown in Tab. 1 the dataset can be divided into two types from the perspective of "Average Number of Objects per Category": small-scale datasets (OS-ESB-core and OS-NTU-core datasets) and medium-scale datasets (OS-MN40-core and OS-ABO-core datasets). The datasets in the former type have about 20 objects per category, and the datasets in the latter type have about 300 objects per category. Here, we select the OS-NTU-core and OS-MN40-core datasets from the two classes to conduct ablation studies as shown in Fig. 8.

In Fig. 8, the red line illustrates our method, and the blue line is the second-best compared method. From Fig. 8, as the "K" varies, the performance of the proposed method is stable and outperforms that of the compared method in most ranges. Besides, we can find that the small-scale dataset OS-NTU-core and the medium-scale dataset OS-MN40-core have different peak values. The former is about ten, and the latter is about 50 for the hyper-parameter "K" of hypergraph construction. We attribute it to the scale of the datasets. Obviously, OS-NTU-core only has 28 objects per category, while OS-MN40-core has 307 objects per category. More objects per category may have more complex correlations among 3D objects. Thus, the larger number of neighbors leads to better performance in the OS-MN40-core datasets. In the constructed hypergraph, hyperedges that contain more nodes can model more complex correlations. However, it may also lead to over-fitting and performance decrease when the "K" comes too large.

5.10 Visualization

In this subsection, we present visualizations of the retrieved results obtained by our proposed method in the open-set 3DOR task.

To intuitively exhibit the discriminative embeddings extracted by the proposed method, we provide some retrieval examples on the OS-MN40-core dataset in Fig. 9. In the open-set 3DOR dataset, the retrieval set consists of the query set and the target set. In Fig. 9, the query 3D objects are selected from the query set, and the retrieval target is the 3D objects from the target set. The categories of 3D objects from both the query set and target are all unseen in the training stage. Generally speaking, the traditional 3DOR task aims to search the object from a seen category, which is prone to falling into the trap of only mastering distinguishing the category of a given 3D object. In the closed-set 3DOR task, this trap may lead to better performance, but in the open-set 3DOR task, it will limit the generalization of the extracted embeddings and lead to worse performance as some unseen categories exist. As shown in Fig. 9, both results with the higher and lower metric scores indicate that the proposed method can distill the shape representation rather than the category representation. Those retrieval examples demonstrate that the proposed method can break the close-set trap and fetch similar objects from the shape representation perspective in the open-set 3DOR task.

5.11 Complexity Analysis

In this subsection, we conduct a comprehensive analysis of complexity, focusing on three key aspects: the number of parameters (Params), the floating point operations (FLOPs), and the inference time. The statistical results of our method on four datasets are presented in Tab. 12. For a deeper investigation, we split our method into "Step 1" and "Step 2". "Step 1" and "Step 2" denote the processes of the multi-modal 3D object embedding and the structure-aware and invariant knowledge learning, respectively. Since our method needs to construct a big hypergraph that contains seen categories from the training set and unseen categories from the retrieval set, the batch size of the proposed method

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

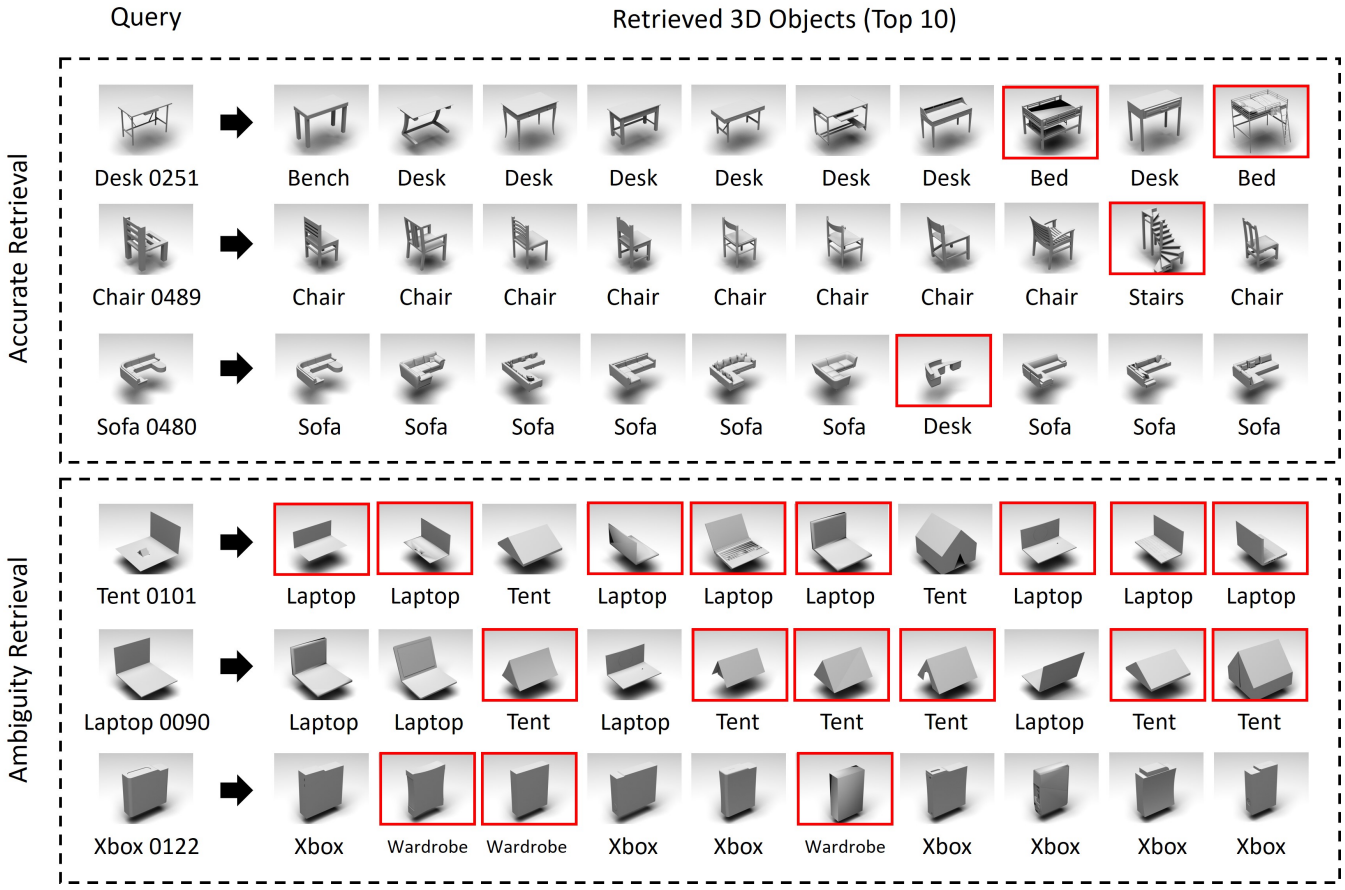


Fig. 9. Visualizations of open-set 3DOR examples on the OS-MN40-core dataset. Note that all categories shown in the illustration are unseen in the training set. The retrieved object is enclosed with a red square if its category is not the same as that of the query object. The text below each query example is its actual name in the OS-MN40-core dataset.

is fixed to the number of all 3D objects of the datasets. In other words, one epoch of our method only contains one batch of data. It is evident that the parameters of our method remain consistent on the four datasets, while the FLOPs and the inference time increase as the bath size grows. This is because the number of parameters is independent of the batch size, and all 3D objects share the same parameters. As for the FLOPs and inference time, as the batch size increases, the dimensions of the input feature matrix also increase. Thus, the complexity of the matrix multiplication also increases, which causes an increase in the computation cost and the inference time. Besides, we notice that the complexity metrics of "Step 2" are all lower than that of "Step 1". It is because the input of "Step 1" is multi-modal features (three feature matrices from three modalities), and contains three auto-encoder. The input of "Step 2" is the fused 3D object features (only one feature matrix). Moreover, in the hypergraph convolution of "Step 2", we remove the learnable parameters and only keep the basic high-order message passing to bridge the features of 3D objects from seen and unseen categories to reduce the cost.

6 Conclusion

In this paper, we introduced the open-set 3DOR task to deal with the limitation of the traditional 3DOR task confronting the existence of unseen categories in practice. We then proposed HGM^{2}R framework to bridge the gaps across different modality representations (such as multi-view, point cloud, and voxel) and learn generalized 3D object embeddings toward the open-set setting. Besides, we formally prove that hypergraph modeling has better representative capability on data correlation than graph modeling. We also generated four multi-modal open-set datasets, *i.e.*, the OS-ESB-core, OS-NTU-core, OS-MN40-core, and OS-ABO-core datasets to evaluate the performance of different methods on the open-set 3DOR task. Experimental results on four datasets demonstrate the effectiveness of the proposed method.

However, there are still some issues that remain to be solved in the future. First, bridging distinctions across different modalities can be handled by the auto-encoder or some metric functions, yet the imbalanced performance of different modal-specific methods still exists. As shown in Tab. 10, the multi-view basic feature is significantly better than the other two modalities. Second, heavily relying on the multi-view modality easily leads to a performance bottleneck. As shown in Fig. 9, the Xbox and Wardrobe may be similar from some angles of view, but they can be easily discriminated from the point cloud or voxel modality. Therefore, building a model from balanced single-modal feature extractors has the potential to achieve better performance than that built from imbalanced ones. Besides, as shown in

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

Fig. S3, most unseen categories have the combined activation score distribution, which indicates that 3D objects may consist of multiple typical representations. Exploring how these typical representations build a 3D object is meaningful and needs to be studied in future works.

REFERENCES

[1] Bustos B, Keim D, Saupe D, et al. Content-based 3D object retrieval[J]. IEEE Computer graphics and Applications, 2007, 27(4): 22-27.
[2] Feng F, Wang X, Li R. Cross-modal retrieval with correspondence autoencoder[C]. Proceedings of MM. 2014: 7-16.
[3] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. Proceedings of ICLR, 2016.
[4] Feng Y, You H, Zhang Z, et al. Hypergraph neural networks[C]. Proceedings of AAAI. 2019, 33(01): 3558-3565.
[5] Jayanti S, Kalyanaraman Y, Iyer N, et al. Developing an engineering shape benchmark for CAD models[J]. Computer-Aided Design, 2006, 38(9): 939-953.
[6] Chen D Y, Tian X P, Shen Y T, et al. On visual similarity based 3D model retrieval[C]. Computer graphics forum. 2003, 22(3): 223-232.
[7] Wu Z, Song S, Khosla A, et al. 3d shapenets: A deep representation for volumetric shapes[C]. Proceedings of CVPR. 2015: 1912-1920.
[8] Collins J, Goel S, Deng K, et al. Abo: Dataset and benchmarks for real-world 3d object understanding[C]. Proceedings of CVPR. 2022: 21126-21136.
[9] Su H, Maji S, Kalogerakis E, et al. Multi-view convolutional neural networks for 3d shape recognition[C]. Proceedings of ICCV. 2015: 945-953.
[10] Feng Y, Zhang Z, Zhao X, et al. Gvcnn: Group-view convolutional neural networks for 3d shape recognition[C]. Proceedings of CVPR. 2018: 264-272.
[11] Su Y, Li Y, Nie W, et al. Joint heterogeneous feature learning and distribution alignment for 2D image-based 3D object retrieval[J]. IEEE TCSVT, 2019, 30(10): 3765-3776.
[12] Wei X, Yu R, Sun J. View-gcn: View-based graph convolutional network for 3d shape analysis[C]. Proceedings of CVPR. 2020: 1850-1859.
[13] Bai S, Bai X, Zhou Z, et al. Gift: A real-time and scalable 3d shape search engine[C]. Proceedings of CVPR. 2016: 5023-5032.
[14] Dong Y, Sawin W, Bengio Y. Hnhn: Hypergraph networks with hyperedge neurons[J]. Proceedings of ICML, 2020.
[15] Bai S, Zhang F, Torr P H S. Hypergraph convolution and hypergraph attention[J]. Pattern Recognition, 2021, 110: 107637.
[16] Vaze S, Han K, Vedaldi A, et al. Open-set recognition: A good closed-set classifier is all you need?[J]. arXiv preprint arXiv:2110.06207, 2021.
[17] Chen G, Peng P, Wang X, et al. Adversarial reciprocal points learning for open set recognition[J]. IEEE PAMI, 2021, 44(11): 8065-8081.
[18] You H, Feng Y, Ji R, et al. Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition[C]. Proceedings of MM. 2018: 1310-1318.
[19] You H, Feng Y, Zhao X, et al. PVRNet: Point-view relation neural network for 3D shape recognition[C]. Proceedings of AAAI. 2019, 33(01): 9119-9126.
[20] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. Proceedings of CVPR. 2016: 770-778.
[21] Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation[C]. Proceedings of CVPR. 2017: 652-660.
[22] Qi C R, Yi L, Su H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space[J]. NIPS, 2017, 30.
[23] Feng Y, Feng Y, You H, et al. Meshnet: Mesh neural network for 3d shape representation[C]. AAAI. 2019, 33(01): 8279-8286.
[24] Wang Y, Sun Y, Liu Z, et al. Dynamic graph cnn for learning on point clouds[J]. TOG, 2019, 38(5): 1-12.
[25] Wu Z, Song S, Khosla A, et al. 3d shapenets: A deep representation for volumetric shapes[C]. Proceedings of CVPR. 2015: 1912-1920.
[26] Maturana D, Scherer S. Voxnet: A 3d convolutional neural network for real-time object recognition[C]. IROS. IEEE, 2015: 922-928.
[27] Scheirer W J, Jain L P, Boult T E. Probability models for open set recognition[J]. IEEE PAMI, 2014, 36(11): 2317-2324.
[28] Bendale A, Boult T E. Towards open set deep networks[C]. Proceedings of CVPR. 2016: 1563-1572.
[29] Joseph K J, Khan S, Khan F S, et al. Towards open world object detection[C]. Proceedings of CVPR. 2021: 5830-5840.
[30] Alliegro A, Borlino F C, Tommasi T. Towards Open Set 3D Learning: A Benchmark on Object Point Clouds[J]. arXiv preprint arXiv:2207.11554, 2022.
[31] Cen J, Yun P, Cai J, et al. Open-set 3D Object Detection[C]. IEEE 3DV, 2021: 869-878.
[32] Shi X, Xu X, Zhang W, et al. Open-Set Semi-Supervised Learning for 3D Point Cloud Understanding[J]. arXiv preprint arXiv:2205.01006, 2022.
[33] Ma H, Xiong R, Wang Y, et al. Towards open-set semantic labeling in 3D point clouds: Analysis on the unknown class[J]. Neurocomputing, 2018, 275: 1282-1294.
[34] He X, Zhou Y, Zhou Z, et al. Triplet-center loss for multi-view 3d object retrieval[C]. Proceedings of CVPR. 2018: 1945-1954.
[35] Nie W, Liang Q, Liu A A, et al. MMJN: Multi-modal joint networks for 3D shape recognition[C]. Proceedings of MM. 2019: 908-916.
[36] Liang Q, Xiao M, Song D. 3D shape recognition based on multi-modal information fusion[J]. Multimedia Tools and Applications, 2021, 80(11): 16173-16184.
[37] Zhao S, Yao H, Zhang Y, et al. View-based 3D object retrieval via multi-modal graph learning[J]. Signal Processing, 2015.
[38] Li Y, Yu A W, Meng T, et al. DeepFusion: Lidar-Camera Deep Fusion for Multi-Modal 3D Object Detection[J]. arXiv preprint arXiv:2203.08195, 2022.
[39] Hu P, Zhen L, Peng D, et al. Scalable deep multimodal learning for cross-modal retrieval[C]. Proceedings of SIGIR. 2019: 635-644.
[40] Jing L, Vahdani E, Tan J, et al. Cross-modal center loss for 3D cross-modal retrieval[C]. Proceedings of CVPR. 2021: 3142-3151.
[41] Wu Y, Wang S, Huang Q. Multi-modal semantic autoencoder for cross-modal retrieval[J]. Neurocomputing, 2019, 331: 165-175.
[42] Zhou D W, Ye H J, Zhan D C. Learning placeholders for open-set recognition[C]. Proceedings of CVPR. 2021: 4401-4410.
[43] Fini E, Sangineto E, Lathuilière S, et al. A unified objective for novel class discovery[C]. Proceedings of ICCV. 2021: 9284-9292.
[44] Zheng J, Zhang S, Wang Z, et al. Multi-channel weight-sharing autoencoder based on cascade multi-head attention for multimodal emotion recognition[J]. IEEE TOMM, 2022.
[45] Oord A, Li Y, Vinyals O. Representation learning with contrastive predictive coding[J]. arXiv preprint arXiv:1807.03748, 2018.

Yifan Feng received the BE degree in computer science and technology from Xidian University, Xi'an, China, in 2018, and the MS degree from Xiamen University, Xiamen, China, in 2021. He is currently working toward the PhD degree from the School of Software, Tsinghua University, Beijing, China. His research interests include hypergraph neural networks, machine learning, and pattern recognition.

Shuyi Ji received the B.E. degree from the School of Software, Tsinghua University, Beijing, China, in 2019. She is currently working toward the Ph.D. degree with the School of Software, Tsinghua University, Beijing, China. Her research interests include hypergraph computation, machine learning, and pattern recognition.

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3332768

Yu-Shen Liu (M'18) received the B.S. degree in mathematics from Jilin University, China, in 2000, and the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2006. From 2006 to 2009, he was a Post-Doctoral Researcher with Purdue University. He is currently an Associate Professor with the School of Software, Tsinghua University. His research interests include shape analysis and machine learning.

Shaoyi Du received double Bachelor degrees in computational mathematics and in computer science in 2002 and received his M.S. degree in applied mathematics in 2005 and Ph.D. degree in pattern recognition and intelligence system from Xi'an Jiaotong University, China in 2009. He is a professor at Xi'an Jiaotong University. His research interests include computer vision, machine learning and pattern recognition.

Qionghai Dai received the M.S. and Ph.D. degrees in computer science and automation from Northeastern University, Shenyang, China, in 1994 and 1996, respectively. He is currently a Professor with the Department of Automation and the Director of the Institute for Brain and Cognitive Sciences, Tsinghua University, Beijing.His research interests include computational photography and microscopy, computer vision and graphics, and intelligent signal processing.

Yue Gao is an associate professor with the School of Software, Tsinghua University. He received the B.S. degree from the Harbin Institute of Technology, Harbin, China, and the M.E. and Ph.D. degrees from Tsinghua University, Beijing, China.

Authorized licensed use limited to: Tsinghua University. Downloaded on December 15,2023 at 02:24:23 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.