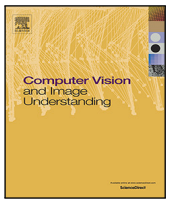
Computer Vision and Image Understanding 259 (2025) 104405

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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



Fast self-supervised 3D mesh object retrieval for geometric similarity

Kajal Sanklecha *, Prayushi Mathur, P.J. Narayanan

Center of Visual Information Technology, International Institute of Information Technology, Hyderabad, India

ARTICLE INFO

MSC:
41A05
41A10
65D05
65D17

Keywords:
Object retrieval
3D triangle mesh
Self-supervision
Embedding space
Ranked retrieval
Mesh analysis

ABSTRACT

Digital 3D models play a pivotal role in engineering, entertainment, education, and various domains. However, the search and retrieval of these models have not received adequate attention compared to other digital assets like documents and images. Traditional supervised methods face challenges in scalability due to the impracticality of creating large, labeled collections of 3D objects. In response, this paper introduces a self-supervised approach to generate efficient embeddings for 3D mesh objects, facilitating ranked retrieval of similar objects. The proposed method employs a straightforward representation of mesh objects and utilizes an encoder–decoder architecture to learn the embedding. Extensive experiments demonstrate the competitiveness of our approach compared to supervised methods, showcasing its scalability across diverse object collections. Notably, the method exhibits transferability across datasets, implying its potential for broader applicability beyond the training dataset. The robustness and generalization capabilities of the proposed method are substantiated through experiments conducted on varied datasets. These findings underscore the efficacy of the approach in capturing underlying patterns and features, independent of dataset-specific nuances. This self-supervised framework offers a promising solution for enhancing the search and retrieval of 3D models, addressing key challenges in scalability and dataset transferability.

1. Introduction

In the realm of digital design, visualization, and communication, 3D models hold a pivotal role, finding extensive use across diverse fields like virtual and augmented reality, gaming, medical imaging, design, scientific visualization, and education. The trajectory of their application signals a growing demand for models characterized by increased variety and intricate detail. The genesis of these rich 3D models predominantly stems from advanced authoring tools, complemented by computer vision algorithms adept at capturing complex models through the lenses of regular and depth cameras. While the triangle (or polygon) mesh stands out as the prevailing and versatile representation for 3D models, alternative methodologies like point clouds (Qi et al., 2017a,b; Li et al., 2018; You et al., 2018; Liu et al., 2019a), implicit surfaces, voxels (Chen et al., 2003; Wu et al., 2015), and signed distance fields have found noteworthy applications.

While there are advanced tools for creating, processing, and analyzing 3D models, the efforts to find and retrieve similar objects from a collection of 3D models are still in the early stages. In comparison to other areas like text retrieval using search engines, the methods for 3D models are not as developed. There are many 3D models available on the internet and specific databases, such as Thingiverse, GrabCAD, Sketchfab, TurboSquid and 3D Warehouse. Stressing the importance of efficient search and retrieval, various open 3D datasets like ModelNet40 (Wu et al., 2015), ShapeNet (Chang et al., 2015), Objaverse (Deitke et al., 2023), CADSketchNet (Manda et al., 2021) and ObjectNet3D (Xiang et al., 2016) highlight the need for methods similar to those used for other types of data.

Historically, early retrieval methods depended on hand-designed descriptors derived from geometric data, such as Spin Images (Johnson, 1997) and Heat Kernel Signatures (Sun et al., 2009). The contemporary landscape, however, witnesses a paradigm shift with descriptors grounded in neural network activations gaining prominence, attributed to their discriminative power, compact representation, and search efficiency. Despite this transition, the field of 3D retrieval methods is still in its formative stages. The associated challenges of classification and recognition of 3D models have attracted increased attention, especially as supervised methods grapple with scalability concerns linked to annotation efforts. This paper charts a course into unexplored territory, advocating for the development of a self-supervised system for 3D retrieval, inspired by methodologies employed in image retrieval (Caron et al., 2020; Hjelm et al., 2018; Wu et al., 2018; Chen et al., 2020) and classification, document retrieval and summarization (Dai et al., 2015; Wang et al., 2019).

The application of supervised Siamese network (Taigman et al., 2014) architectures, acknowledged luminaries in image recognition (Taigman et al., 2014), is broadened to encompass the realm of 3D

* Corresponding author.
E-mail address: kajal.sanklecha@research.iiit.ac.in (K. Sanklecha).

https://doi.org/10.1016/j.cviu.2025.104405
Received 17 March 2024; Received in revised form 7 May 2025; Accepted 25 May 2025
Available online 6 June 2025

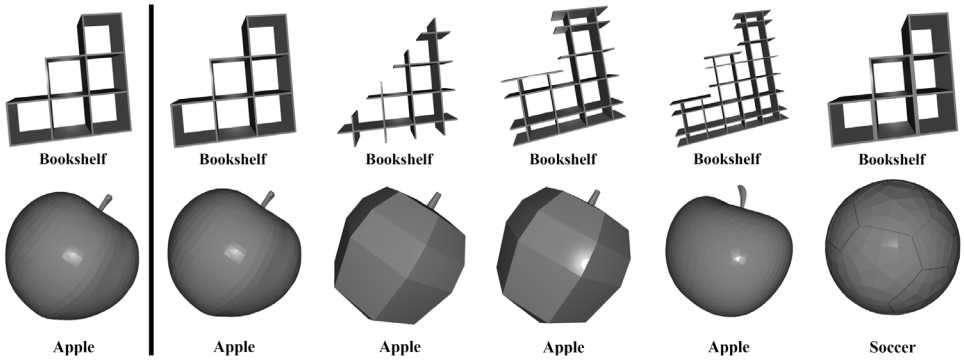


Fig. 1. This figure illustrates the retrieval effectiveness of our self-supervised method. The query object (left) is matched with the top 5 most similar objects from the database (right), ranked by decreasing similarity (leftmost = most similar). Notably, the retrieved objects ("Bookshelf" from ShapeNet (Chang et al., 2015) and "Apple" from Objaverse (Xiang et al., 2016)) exhibit strong geometric resemblance to the query, highlighting the model's ability to capture intrinsic shape. Labels are provided for reference only and are not used during retrieval.

mesh objects. These architectures, operating in a self-supervised manner, obviate the need for object annotations while efficiently learning embeddings. Current 3D retrieval methodologies are broadly classified into view-based and model-based strategies. The former, reliant on multi-view images of the 3D object, grapples with the constraint of capturing only two dimensions of the three-dimensional information, resulting in data loss. On the flip side, model-based 3D retrieval methods center their focus on the shape and structure of objects, employing descriptors, bag-of-features, shape context, graph-based or spectral methods, and deep learning-based approaches such as CNNs for efficient retrieval.

In this research, we introduce a novel self-supervised method for retrieving 3D mesh objects, eliminating the requirement for annotated data. This approach extends the concepts presented in the MORe (Sanklecha et al., 2023) paper, incorporating a simple yet effective encoder–decoder structure. Notably, our model is trained on the ModelNet40 (Wu et al., 2015) dataset, showcasing its adaptability and transferability across different datasets, including ShapeNet (Chang et al., 2015)and Objaverse (Xiang et al., 2016) as shown in Fig. 1. Despite the absence of labeled data, our approach demonstrates comparable performance to traditional supervised methods in terms of mean Average Precision (mAP) scores. Our major contributions are:

- **Self Supervision:** A self-supervised method for 3D mesh object retrieval, eliminating the need for annotated data and emphasizing simplicity in representation.
- **Transferability:** Demonstrated the model's transferability by training on the ModelNet40 dataset and achieving robust performance on larger datasets like ShapeNet and Objaverse.
- **Fast Retrieval:** The system can quickly locate 3D objects in a database, requiring only 0.25 s for retrieval.
- **Orientation-Invariant Retrieval:** The system ensures that 3D object retrieval is not affected by their orientation. It automatically aligns all objects to a consistent angle, enabling accurate identification regardless of their initial position.
- **Dataset Contribution:** Contributed a lightweight preprocessed dataset to the community, fostering collaboration, reproducibility, and future advancements in 3D mesh object retrieval research.

2. Related works

The realm of 3D object processing and retrieval is vast, encompassing literature on classification, retrieval, self-supervised learning, and mesh analysis. This discussion delves into recent efforts contributing to the field of self-supervised 3D object retrieval.

2.1. Traditional methods

Within this section, we delve into the categorization of previous research, particularly emphasizing traditional methodologies utilized in the analysis of 3D objects.

2.1.1. Classification

Prior to the deep learning revolution, 3D object classification relied on various techniques to automatically understand object categories based on their 3D structure. Methods like spin images (Johnson, 1997) and quasi-spin images (van Blokland and Theoharis, 2020) captured the object's overall shape, while 3D SIFT (Scovanner et al., 2007) (Scale-Invariant Feature Transform) and 3D SURF (Knopp et al., 2010) (Speeded Up Robust Features) focused on identifying and describing distinctive points (keypoints) and their surrounding areas (descriptors) on the object. By analyzing these features and comparing them to a database of known objects with pre-assigned categories, the system could assign the most likely category label to the unknown object. However, these traditional methods often required expert knowledge for feature design and were sensitive to variations in object pose, viewpoint, and data noise. The emergence of deep learning has transformed this field by enabling the automatic learning of robust features directly from data, leading to significant advancements in the accuracy and robustness of 3D object classification tasks.

2.1.2. Retrieval

Searching and retrieving relevant information from massive data collections has always been crucial. Traditionally, text retrieval relied on techniques like probabilistic models (Amati and Van Rijsbergen, 2002), latent semantic indexing (LSI) (Dumais et al., 2004), the Okapi BM25 (Robertson et al., 2009) algorithm, and the PageRank (Brin and Page, 1998) algorithm to identify the most relevant documents. With the rise of machine learning, approaches like ranking documents by relevance (Burges et al., 2005) and utilizing models such as BERT (Yates et al., 2021) have emerged. Similarly, image retrieval traditionally employed methods like bag-of-visual-words (BoVW) (Shekhar and Jawahar, 2012), vector of locally aggregated descriptors (VLAD) (Jégou et al., 2010), and fisher vectors (Perronnin et al., 2010) to analyze and find similar images within large datasets.

2.2. Neural network-based methods

In this section, we explore the landscape of research focusing on neural network-based methods, highlighting the transformative impact of deep learning on the classification and retrieval tasks associated with 3D objects.

2.2.1. Supervised

The rise of machine learning and neural networks has revolutionized 3D object classification. These methods differ from traditional approaches by learning discriminative features directly from raw data, eliminating the need for hand-crafted features and achieving superior performance. Popular examples of deep learning methods for 3D object classification include MeshNet (Feng et al., 2019), MeshCNN (Hanocka et al., 2019), MeshNet++ (Singh et al., 2021), and MeshMAE (Liang et al., 2022). These techniques unlock the potential to automatically analyze and understand the intricate shapes and structures of 3D objects, leading to significant advancements in various fields.

While deep learning has led to remarkable advancements in image retrieval with techniques like text-image search (Wang et al., 2016) and fine-tuned CNNs (Radenović et al., 2018), extending these methods directly to the 3D domain presents unique challenges. This is because existing approaches like ULIP (Xue et al., 2023a), OpenShape (Liu et al., 2023), and ULIP2 (Xue et al., 2023b) primarily focus on retrieving 3D models using text or image embeddings, not directly retrieving 3D objects themselves. The complex nature of 3D data necessitates the development of specialized techniques specifically tailored for 3D object retrieval. This presents an exciting research opportunity to explore and develop novel methods that can effectively identify and retrieve relevant 3D objects from large datasets.

2.2.2. Self-supervised

Self-supervised learning has emerged as a powerful technique in various domains, allowing algorithms to learn directly from unlabeled data. Unlike traditional methods requiring human-annotated labels, self-supervised algorithms create their own learning tasks. These tasks often involve learning useful representations of the data, achieved through techniques like autoencoders. This approach has proven successful in fields like image retrieval (Chen et al., 2020; He et al., 2020b; Grill et al., 2020), signal processing (Baevski et al., 2020; Oord et al., 2018), and document retrieval (Adhikari et al., 2019; Clark et al., 2020; Abolghasemi et al., 2022), where it learns valuable features without the need for explicit supervision. In the realm of 3D point cloud processing, self-supervised and unsupervised learning methods are also gaining traction. Some examples include Deep Spatiality (Han et al., 2018), L2G Auto-encoder (Liu et al., 2019b), and 3D local features (Thabet et al., 2020), which specifically focus on generating local features for 3D objects. Additionally, methods like Deep Shape descriptor (Shi et al., 2020), few-shot learning (Sharma and Kaul, 2020; Bharti et al., 2020), sampling invariance (Chen et al., 2021), pretraining of 3D features (Zhang et al., 2021), and graph topology (Chen et al., 2019) aim to learn representations for the entire 3D point cloud, tackling different aspects like shape description, few-shot learning, and capturing the overall structure of the point cloud. These various approaches and their successes highlight the potential of self-supervised learning for tackling complex tasks in 3D point cloud processing.

2.3. Retrieval

This section examines retrieval methods, focusing on distributing the methods based on the query of the retrieval or the results generated by the methods. This section focuses on the input and output of the retrieval process instead of the method as done in the previous sections.

2.3.1. Based on query

In the captivating domain of 3D object retrieval, the search strategy can be guided by various query types. Users can leverage view-based methods (Gao et al., 2020; He et al., 2019; Bai et al., 2016) like those explored in which utilize one or more images of the desired object to find similar ones within a database. Alternatively, sketch-based methods (Eitz et al., 2012; Wang et al., 2015; Li et al., 2015; Qi et al., 2018; van Blokland and Theoharis, 2018) enable users to search using 2D sketches or drawings, allowing for a freehand approach to representing the desired object. Finally, object-based methods (Bronstein et al., 2011; Xie et al., 2016; Feng et al., 2019; Singh et al., 2021) empower users to search using existing 3D models or even parts of objects as queries, providing a more detailed representation for the search. Our work contributes to this field by proposing a model-based retrieval system, which falls under the umbrella of object-based methods, offering an additional avenue for users to efficiently find the 3D objects they seek.

2.3.2. Based on results

Delving deeper into the world of 3D object retrieval, methods go beyond just the type of query (view-based, sketch-based, etc.). They also consider various object properties to refine the search. One key distinction lies in the object's rigidity: rigid object retrieval methods (Bai et al., 2016; Feng et al., 2019; Singh et al., 2021; van Blokland and Theoharis, 2018, 2020) specialize in searching for objects with fixed shapes, while non-rigid object retrieval methods (Bronstein et al., 2011; Xie et al., 2016; Sharma et al., 2011) handle objects that can deform, like clothing or human bodies. Additionally, methods can be categorized based on the completeness of the object being searched for: complete object retrieval methods (Bai et al., 2016; Feng et al., 2019; Singh et al., 2021; Hamdi et al., 2021) involve searching for entire objects, while part-based retrieval methods (Bai et al., 2016; van Blokland and Theoharis, 2018, 2020) focus on finding objects with similar components. For complete object retrieval, methods like Shape Google (Bronstein et al., 2011) and Deep Shape (Xie et al., 2016) utilize Heat Kernel Signature (HKS) (Bronstein and Kokkinos, 2010) as features for comparison. Part-based retrieval, on the other hand, decomposes an object into its parts and uses those parts as queries, with examples like Microshapes (van Blokland and Theoharis, 2018) and QUICCI (van Blokland and Theoharis, 2020) employing Hamming distance for feature matching. Additionally, methods like GIFT use inverted files for matching and re-ranking retrieved objects. This diverse landscape of approaches and considerations highlights the flexibility and adaptability of 3D object retrieval methods, allowing users to tailor their search strategies based on the specific characteristics of the objects they seek.

2.4. Mesh processing

While 3D shapes in computer graphics and vision are traditionally analyzed and processed using geometric methods, recent advancements in deep learning offer exciting new possibilities. Methods like MeshCNN (Hanocka et al., 2019), ExMeshCNN (Kim and Chae, 2022), Picasso (Lei et al., 2021), MeshMAE (Liang et al., 2022), and Markov Random Fields (Lavoué and Wolf, 2008) represent a wave of deep learning approaches pushing the boundaries of mesh analysis. MeshCNN (Hanocka et al., 2019) uses mesh-specific operations to directly classify 3D shapes, while ExMeshCNN (Kim and Chae, 2022) delves deeper, learning both the surface and geometric characteristics of a mesh for improved analysis. Picasso leverages deep learning to achieve faster and higher-quality mesh simplification, and MeshMAE (Hanocka et al., 2019) utilizes powerful transformer architectures to tackle mesh classification and segmentation tasks effectively. Additionally, Markov Random Fields offer a probabilistic approach to understanding mesh structure and relationships between its components, further enhancing the analysis capabilities. These deep learning methods, though requiring careful mesh representation for optimal results, demonstrate the potential for significant advancements in processing and analyzing 3D shapes, impacting various applications in computer graphics and vision. Compared to data on a grid, meshes require specific attention in their representation before being used by neural networks to extract maximum information from the 3D object.

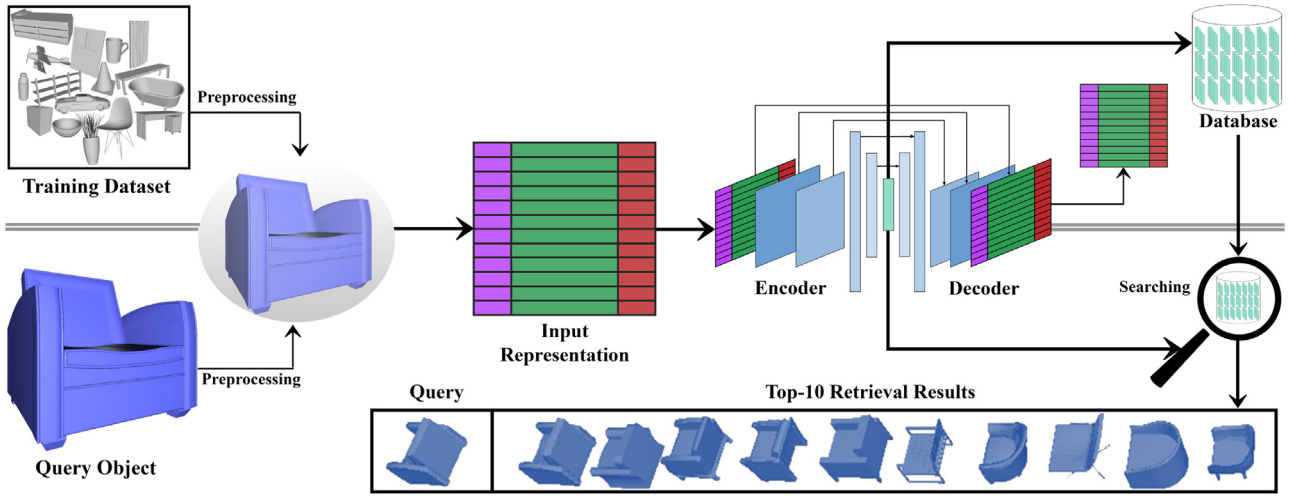


Fig. 2. Our system operates in two distinct phases: an offline training and database building stage, and an online querying stage. Both leverage the same self-supervised trained encoder to generate representations and embeddings. The offline phase heavily relies on preprocessing for normalization and defining the input representation (detailed in Fig. 3). This meticulous preparation ensures efficient representation generation for online querying, enabling lightning-fast retrieval when a user submits a search.

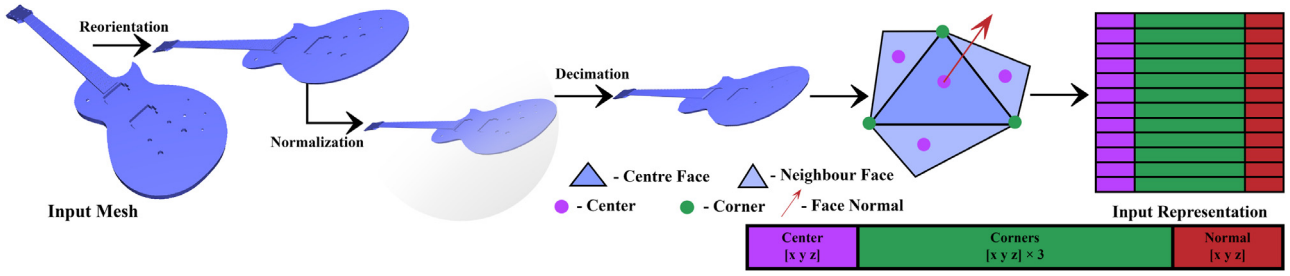


Fig. 3. To streamline processing, 3D objects undergo preprocessing: reorientation for consistency, normalization for uniform scale, and decimation to simplify the mesh. This results in a compact representation for each object, enabling swift retrieval.

3. Method

Our proposed method for 3D object retrieval is outlined in Fig. 2. To ensure consistent processing, all mesh objects undergo pre-processing. This involves scaling them to a unit sphere and reducing their triangle count to a uniform 512. Inspired by MeshNet++ (Singh et al., 2021), we represent each face of the triangular mesh using its center, corner, and normal vector coordinates. This serves as a sufficient and efficient representation for our purposes. A fully-connected encoder–decoder network is employed to learn an embedding for each object. This embedding captures the object's essential geometric features and is used to create a database of object representations. During retrieval, the encoder is used to embed query objects. By comparing the embedded query object to the database of embeddings, similar objects can be ranked and retrieved solely based on their geometric similarity. To ensure the learned embedding effectively captures the relationships within the object collection, we leverage a self-supervised learning approach during the training process. This approach allows the model to learn meaningful representations even without the need for explicit object category labels. The details of the self-supervised learning objective and the network architecture are provided in the following sections.

3.1. Preprocessing

We perform some essential preprocessing steps as shown in Fig. 3, to prepare the 3D mesh objects to extract the essential properties of the mesh. This ensures the data is consistent and suitable for analysis, ultimately improving the model's performance. Following are the required steps:

3.1.1. Reorientation

To achieve rotation invariance and facilitate efficient object comparison, our method incorporates a novel pre-processing step specifically designed for meshes. We employ Poisson Disk Sampling to select a representative set of 5000 points, sampled uniformly from the surface of the object i.e. mesh, ensuring a more uniform distribution compared to using all vertices. By applying Principal Component Analysis (PCA) to these sampled points as shown in Fig. 4, we identify the object's principal axes, representing the directions of greatest variance. Finally, we reorient the original mesh by aligning its most prominent axis with the X-axis based on the top three PCA vectors. This approach effectively addresses non-uniform vertex distribution and achieves rotation invariance, allowing for accurate object comparison and retrieval solely based on their intrinsic geometric features, regardless of their initial orientation in the database.

3.1.2. Normalization

The pre-processing stage incorporates normalization to ensure consistent object representation and facilitate efficient processing. Each triangle mesh undergoes two normalization steps:

1. Size normalization: the mesh is uniformly scaled to fit within a unit sphere, ensuring all objects have comparable size and preventing larger objects from biasing distance calculations during retrieval.
2. Origin centering: the object's center of mass is shifted to the origin, removing any bias due to the object's original position and allowing the retrieval process to focus solely on its intrinsic geometric features.

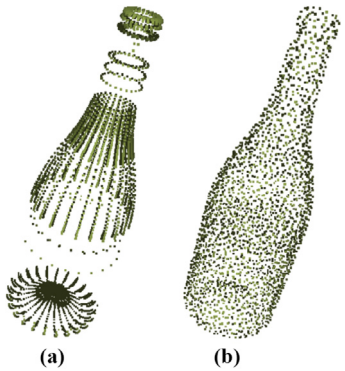


Fig. 4. Why Point Cloud Sampling for Reorientation? The figure illustrates the importance of uniform point cloud sampling for mesh reorientation. Here, (a) depicts the vertex distribution of a mesh object where the PCA-based dominant angles can introduce bias due to non-uniform vertex distribution. In contrast, (b) showcases uniform random sampling across the mesh surface, leading to more reliable principal component analysis (PCA) for determining the dominant axes and achieving accurate mesh reorientation.

These normalization techniques establish a standardized representation for all objects, enabling effective comparison and retrieval based on their inherent geometric characteristics.

3.1.3. Decimation

In our approach, we address the computational challenges associated with processing high-resolution meshes. While our previous work employed a decimation strategy, "simplify" method from Manifold (Huang et al., 2018) to reduce the number of triangles to a uniform size (n = 512), this approach may not be universally successful across diverse datasets and object types. To overcome this limitation, we propose a two-pronged approach:

1. Initial Decimation: We begin by applying the "manifold" method from Manifold to attempt the decimation process. This method aims to simplify the mesh while preserving its structural integrity.
2. Iterative Refinement: If the initial attempt using "manifold" does not achieve the desired n = 512 faces, we employ the "simplify" method iteratively. This iterative process continues until the target number of faces is successfully attained.

This combined approach ensures consistent mesh decimation across various datasets, regardless of the object's initial complexity. By reducing the mesh size, we significantly reduce the computational cost associated with subsequent processing and analysis steps within our method.

3.1.4. Input representation

Following extensive experimentation, we choose a simple and reduced representation for each triangle within the mesh. This representation captures the following:

- Center: The (x, y, z) coordinates of the triangle's center, providing information about the object's overall structure in 3D space.
- Corners: The (x, y, z) coordinates of each of the three corners of the triangle, capturing the specific shape and boundaries of the triangle.
- Normal: The (x, y, z) coordinates of the unit normal vector of the triangle, indicating its outward direction and contributing to the overall surface orientation of the object.

This combination of features serves as a foundation for effectively capturing the essential geometric aspects of 3D objects represented as triangle meshes, enabling our proposed method to perform efficient and accurate retrieval tasks.

3.2. Architecture

Our method adopts a Siamese network architecture to learn embedding vectors that represent the entirety of each mesh object in an embedding space. This space allows for efficient comparison and retrieval based on the learned features. While typical encoder–decoder networks in the image domain leverage convolutional layers for feature extraction, these layers are not directly applicable to mesh data. To address this, we employ multilayer perceptrons (MLPs) as fully connected layers within both the encoder and decoder (Fig. 2). Each component has five MLP layers, and we incorporate skip connections to facilitate the flow of gradients and mitigate the vanishing gradient problem. This architecture ultimately produces a 256-dimensional embedding vector from the encoder's bottleneck layer, capturing the essential features of the input mesh object for effective retrieval. The code snippet of our model architecture is given below:

```
# encoder
enc1=nn.Linear(inp=len*n_face,out=12*n_face)
enc2=nn.Linear(inp=12*n_face,out=9*n_face)
enc3=nn.Linear(inp=9*n_face,out=3072)
enc4=nn.Linear(inp=3072,out=1024)
enc5= nn.Linear(inp=1024,out=256)
# decoder
dec5=nn.Linear(inp=256,out=1024)
dec4=nn.Linear(inp=2*1024,out=3072)
dec3=nn.Linear(inp=2*3072,out=9*n_face)
dec2=nn.Linear(inp=2*9*n_face,out=12*n_face)
dec1=nn.Linear(inp=2*12*n_face,out=len*n_face)
# main
def forward(x):
    x1 = relu(enc1(x))
    x2 = relu(enc2(x1))
    x3 = relu(enc3(x2))
    x4 = relu(enc4(x3))
    fea= x5 = relu(enc5(x4))
    x6 = relu(dec5(x5))
    x6 = torch.cat((x6, x4), dim=2)
    x7 = relu(dec4(x6))
    x7 = torch.cat((x7, x3), dim=2)
    x8 = relu(dec3(x7))
    x8 = torch.cat((x8, x2), dim=2)
    x9 = relu(dec2(x8))
    x9 = torch.cat((x9, x1), dim=2)
    x10 = relu(dec1(x9))
    return x10, fea
```

where n_face is the number of faces in each mesh and len is the length of the representation for each face as given in Section 3.1.4. enc_i denote the encoder layers and dec_i denote the decoder layers for the encoder decoder architecture. The framework used by the method is PyTorch. The model is trained using the Mean Squared Error (MSE) loss function, optimizing performance through the Stochastic Gradient Descent (SGD) optimizer. A MultiStep Learning Rate scheduler is employed to adjust the learning rate, which is initially set to 0.05, ensuring stable convergence. Training is conducted with a batch size of 8 over 300 epochs, utilizing an NVIDIA RTX 3090 GPU to efficiently handle computations and optimize performance.

3.3. Object retrieval

Our approach capitalizes on the efficiency of nearest neighbor search techniques within the learned embedding space. This enables efficient retrieval from large datasets due to the inherent scalability of these techniques to dataset size. Following the training of the encoder–decoder model, all objects in the database are processed through the encoder, generating their corresponding embedding representations within the embedding space. This established database serves as the foundation for the subsequent object retrieval process, which consists of two key stages: database generation and retrieval.

3.3.1. Database generation

To facilitate efficient retrieval using the learned embedding space, we generate a database that maps input mesh objects to their corresponding feature vectors. This database serves as the foundation for subsequent retrieval tasks. The generation process involves passing each mesh object through the trained encoder model, which outputs a 256-dimensional embedding vector capturing its key features. These embedding vectors, collectively forming the database, serve as the basis for comparing query objects and retrieving similar ones, as explained in the following section.

3.3.2. Retrieval

The retrieval stage focuses on processing user queries. We first pre-process the query mesh and encode it using the trained encoder, generating a 256-dimensional embedding vector that encapsulates its intrinsic characteristics. This vector is then used to search the established database. Objects within the database are ranked based on their L2 distance to the query object's embedding, with smaller distances indicating greater similarity. The top-ranked objects, representing the closest matches to the query based on their features captured in the embedding space, are presented as the final retrieval results. Similar objects reside close together in the embedding space, while dissimilar ones are positioned farther apart, reflecting the effectiveness of the learned space for accurate retrieval based on overall similarity. This approach facilitates efficient retrieval of objects with similar structural properties from large datasets.

4. Datasets

While trained on the train set of ModelNet40, we evaluate our model's performance on a broader range of data, including ShapeNet (Chang et al., 2015), Objaverse (Xiang et al., 2016), and the ModelNet40 (Wu et al., 2015) test set. This ensures the model generalizes well beyond its training data.

4.1. ModelNet

This research leverages the ModelNet40 dataset for 3D object retrieval. The dataset has 12,311 labeled objects from 40 categories, but to reduce computational complexity, the meshes are preprocessed to have n = 512 triangles. This process discards some objects, resulting in a final dataset of 11,920 meshes split 83%–17% for training and testing.

4.2. Shapenet

To thoroughly evaluate our model's ability to generalize beyond its training data, we employ the ShapeNet (Chang et al., 2015) dataset for testing. This comprehensive dataset boasts 52,190 labeled objects across 55 distinct categories. However, to ensure successful preprocessing and compatibility with our model, we implemented an iterative Manifold method, as a single pass proved insufficient.

4.3. Objaverse LVIS

While the Objaverse (Xiang et al., 2016) dataset boasts over 800,000 objects, we focus on the annotated subset (potentially referred to as Objaverse LVIS) for evaluation purposes. This choice is not driven by needing labeled data for our method, but rather by the need for calculable metrics to assess its performance. To ensure sufficient retrieval matches, we further filter the categories, selecting only those with over 100 objects from the initial 1230 categories. This meticulous selection process results in a final set of 76 usable classes containing 9,546 objects, which then undergo the preprocessing steps as mentioned in Section 4.2.

5. Experimentation and results

5.1. Implementation details

Our model leverages the ModelNet40 dataset (detailed in Section 4.1) for training. The training process involves generating embeddings for all objects within the database and takes approximately 15 h to complete on an Nvidia RTX 3090 GPU. The model itself has a footprint of 1.1 GB. During training, we employ stochastic gradient descent with a multi-step learning rate scheduler to optimize the model. The learning rate is set to 0.05, and the training utilizes a batch size of 8 objects. The L2 loss function with ReLU activation on each layer guides the training process.

Once trained, the generated object embeddings are stored for efficient retrieval. When a user submits a query object, it undergoes the same pre-processing steps and is then fed through the trained encoder network. This process generates an embedding vector for the query object, which is then compared to the stored database embeddings. Objects within the database are ranked based on their L2 distance to the query embedding, with smaller distances indicating greater similarity. To evaluate the effectiveness of our approach, we test the model on separate test sets from ModelNet40, ShapeNet, and Objaverse LVIS (details provided in Section 4). The testing times per step are presented in Table 3. We report top-k mean Average Precision (mAP) scores (typically k = 5 or 10) to quantify the retrieval accuracy based on the ranked results.

5.2. Results

Our proposed method achieves impressive results on both quantitative and qualitative fronts, demonstrating its effectiveness in retrieving 3D objects based on geometric similarity. While quantitative metrics provide an objective measure of retrieval accuracy, achieving visually compelling results is equally crucial. Given that visual similarity is our primary focus, this section delves into the qualitative aspects of our retrieval results, showcasing how they compare favorably to those produced by competing methods. As mentioned in Section 4, our model has been rigorously tested on various datasets. As mentioned in Table 4, our ablation study justifies the design decisions taken.

5.2.1. Quantitative validation

Our method prioritizes capturing the intrinsic geometric properties of objects for retrieval. While class labels are not used during retrieval itself, they are employed for quantitative evaluation using mean Average Precision (mAP). As shown in Table 1, our method achieves state-of-the-art mAP scores compared to existing methods. This effectiveness extends to unseen data, with outstanding mAP scores on various datasets beyond the training set (Table 2).

Our method prioritizes computational efficiency for practical applications. As detailed in Section 3, Table 3 presents the execution times for various processing steps. Notably, the method achieves retrieval in a mere 201 ms for pre-processed manifold meshes. For non-manifold meshes, an off-the-shelf method is employed for efficient conversion.

5.2.2. Qualitative validation

The qualitative retrieval results, showcased in Figs. 9, 10, and 11, for ModelNet40, Shapenet, and Objaverse LVIS, respectively, illustrate the effectiveness of our method in capturing the geometric shape of objects. These figures reveal that retrieved objects exhibit a high degree of visual similarity to the query object, even when belonging to different categories. This emphasizes the model's ability to identify objects with similar shapes, independent of their classification labels.

Figs. 5 and 6 exemplify some of the most intriguing retrieval capabilities of our method for Shapenet and Objaverse LVIS respectively. Here, we observe how the model effectively captures the geometric

Table 1
Comparison of the mAP scores achieved by our self-supervised method with those of state-of-the-art retrieval methods on the ModelNet40 dataset. Our method demonstrates competitive performance, achieving results that are on par with or superior to existing approaches.

Method	Modality	mAP (%)
Methods that need supervision		
MVCNN (Su et al., 2015)	Multiview(MV)	80.2
3D ShapeNets (Wu et al., 2015)	Voxels	49.2
GIFT (Bai et al., 2016)	Multiview	81.9
PVNet (You et al., 2018)	Points & MV	88.5
PVRNet (You et al., 2019)	Points & MV	90.5
MLVCNN (Jiang et al., 2019)	Multiview	92.2
DensePoint (Liu et al., 2019a)	Points	88.5
MeshNet (Feng et al., 2019)	Mesh	81.9
GSL (ATCL) (He et al., 2020a)	Multiview	90.1
MMFN (Nie et al., 2020)	Multimodal	90.3
Attention-Guided Fusion Network (Peng et al., 2020)	Points & MV	91.4
MVTN (Hamdi et al., 2021)	Multiview	92.9
DAGL (Shi et al., 2022)	Multiview	91.1
SCA-PVNet (Lin et al., 2024)	Points & MV	92.5
Methods that need no supervision		
LFD (Chen et al., 2003)	Voxels	40.9
SPH (Kazhdan et al., 2003)	Voxel	33.9
DAN (Nie et al., 2021)	Multiview	90.4
VSP-GAN (Zhou et al., 2024)	Multiview	78.2
Ours Top-10	Mesh	87.88
Ours Top-5	**Mesh**	**94.87**

Table 2
Summary of the mAP scores achieved by our method for top-k retrieval tasks (k = 5 and 10) on different datasets (refer to Section 4 for details).

Dataset	Top-5	Top-10
ModelNet40 (Wu et al., 2015)	94.87	87.88
Shapenet (Chang et al., 2015)	92.92	84.51
Objaverse LVIS (Deitke et al., 2023)	96.11	89.61

Table 3
The execution times associated with different processing steps within our method (refer to Section 3 for details).

Step	Time taken
Off-the shelf method	
Manifold	2.375 s
Our method	
Decimation and Reorientation	203.44 ms
Normalization	4.2905 ms
Embedding extraction	43.728 ms
Retrieval	0.310 ms
Total Retrieval time	**251.76 ms**

shape of the query object. The retrieved objects exhibit fascinating behavior, demonstrating the model's ability to identify objects that share similar shapes despite potentially belonging to different categories. This highlights the strength of our approach in focusing on the underlying geometric characteristics that define object similarity, going beyond classification labels.

Fig. 7 presents the confusion matrix for object retrieval on the ShapeNet dataset. The prominent diagonal elements indicate that our method successfully retrieves objects from the same class as the query object, even without relying on class labels. This achievement underscores the model's ability to identify geometric shape similarities as the primary driver for retrieval. However, the matrix also reveals interesting off-diagonal elements, particularly those highlighting retrieval pairs like "rifle" and "pistol" or "radiotelephone" and "telephone". These pairings further emphasize the model's strength in capturing underlying geometric shape similarities, even when the objects belong to distinct categories. Similarly, Fig. 8, presents the confusion matrix for object retrieval on the Objaverse LVIS dataset. This behavior reinforces the effectiveness of our method in focusing on intrinsic geometric properties for accurate retrieval.

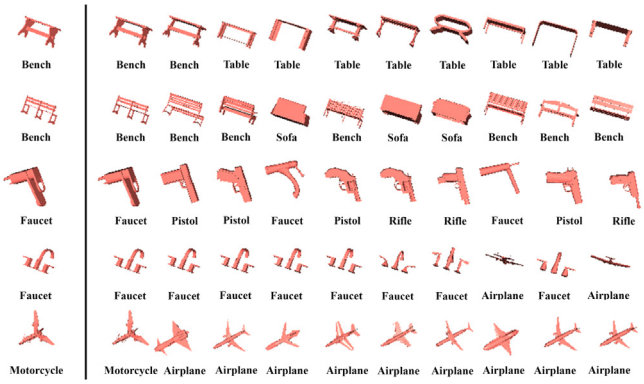


Fig. 5. ShapeNet Retrieval Highlights. This figure showcases select retrieval results from the ShapeNet dataset, emphasizing our model's ability to identify objects from different classes that share remarkable geometric similarities.

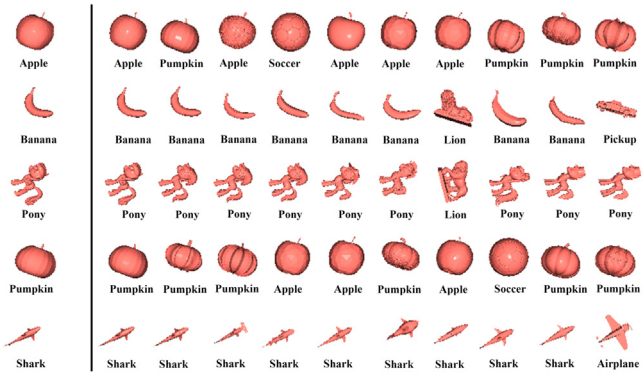


Fig. 6. Objaverse LVIS Retrieval Highlights. This figure showcases select retrieval results from the Objaverse LVIS (Deitke et al., 2023) dataset, emphasizing our model's ability to identify objects from different classes that share remarkable geometric similarities.

5.3. Limitation and future work

Our current approach effectively captures the overall structure of 3D meshes but sacrifices finer details due to decimation, a necessary step given our system's memory constraints. To address this, we plan to explore larger systems and optimize MLP layers for reduced memory usage.

Moreover, the mismatch between MLPs' sequential input expectation and meshes' inherently unordered nature presents a challenge. We will investigate methods to establish meaningful vertex or face orderings or develop permutation-invariant MLPs.

While our method excels at retrieving rigid objects, it struggles with non-rigid shapes. Future work will focus on incorporating geodesic distances to better handle the varying poses and forms of non-rigid objects.

Our scale-invariant retrieval, achieved through normalization, is beneficial in many cases like those showcased in Fig. 7 with the "cap"

Table 4
Results of various ablation studies. Based on these results, we settle on 512 faced meshes, 5 layer network, 256 dimensions for embedding and no neighbors to represent each triangle.

Experiment		mAP	
		Top-5	Top-10
Number of faces	**512**	**0.9487**	**0.8788**
	768	0.9484	0.8752
	1024	0.9466	0.8734
Number of network layers	4	0.9480	0.8700
	5	**0.9487**	**0.8788**
	6	0.9453	0.8693
Embedding Vector Dimension	128	0.9463	0.8591
	256	**0.9487**	**0.8788**
	512	0.9476	0.8635
	1024	0.9471	0.8641
Input Parameters	No centers	0.9472	0.8641
	No corners	0.9457	0.8591
	No normals	0.9461	0.8598
	No neighbors	**0.9487**	**0.8788**
	1-ring Neighbors	0.9469	0.8688
	2-ring Neighbors	0.9471	0.8641

Table 5
Results on Modelnet40 circular rotation and patch-wise shuffle of the input vector for limited permutation invariance.

Experiment		mAP	
		Top-5	Top-10
Circular Rotation by n rows	0	0.9488	0.8792
	2	0.9511	0.8769
	4	0.9462	0.8746
	6	0.9283	0.8562
	8	0.8665	0.8070
Patch-wise shuffle of n rows	1	0.9488	0.8792
	2	0.9484	0.8799
	4	0.9485	0.8781
	8	0.9480	0.8776
	16	0.9499	0.8781
	32	0.9433	0.8714
	64	0.7139	0.6648
	128	0.5383	0.5177

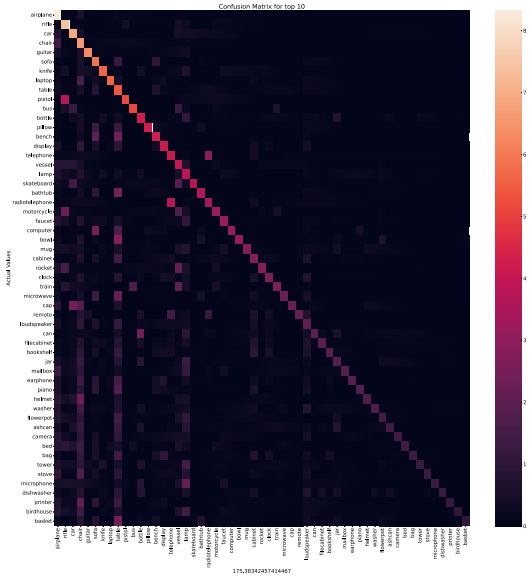


Fig. 7. Confusion Matrix for top-10 retrieval of the 55 classes of Shapenet.

- “car” and “bathtub” - “bowl” matches. This may be limiting when preserving object scale is essential. We plan to explore incorporating

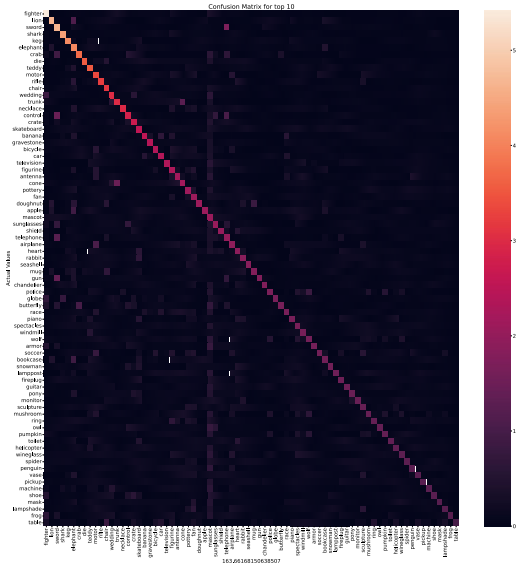


Fig. 8. Confusion Matrix for top-10 retrieval of the 75 classes of Objaverse LVIS as described in Section 4.3.

scale information as a selective criterion during retrieval to enable more context-aware matching.

Potential Solutions:

- **Finer Details**: Employ a two-level network architecture to initially capture the coarse shape and then delve into finer details.
- **Permutation Invariance**: Develop permutation-invariant MLPs or establish a meaningful uniform ordering for meshes.
- **Non-Rigid Objects**: Incorporate geodesic distances to handle varying poses and forms.
- **Scale-Aware Retrieval**: Introduce scale information as a selective criterion to enable context-based matching.

Our method operates directly on triangle meshes, the dominant 3D object representation. While we currently decimate objects to 512 triangles due to GPU memory limitations. This approach effectively captures the overall shape crucial for retrieval tasks but might miss finer details. Additionally, generating water-tight meshes with exactly 512 triangles can be challenging. Another limitation lies in the method’s sensitivity to triangle permutations as shown in Table 5. Although mesh representations are theoretically permutation invariant, object creation tools often employ specific triangle ordering schemes that can impact our method’s performance. To address this, future work will investigate preprocessing techniques that ensure a canonical triangle ordering across all meshes, enhancing the robustness of our method.

Our method leverages normalization to achieve scale-invariant retrieval based on geometric shape, offering advantages like those showcased in Fig. 7 with the “cap”-“car” and “bathtub”-“bowl” matches. However, this normalization introduces a limitation in scenarios where preserving the object’s scale is critical. Future work will explore incorporating scale information as a selective criterion during retrieval.

6. Conclusion

In conclusion, we presented a novel self-supervised approach for efficient 3D mesh object retrieval. By leveraging a simple mesh representation and an encoder–decoder architecture, our method successfully generates informative embeddings that capture the intrinsic geometric characteristics of objects. This approach overcomes the limitations of traditional supervised methods, particularly the need for vast amounts of labeled data. Our method achieves state-of-the-art performance on benchmark datasets and exhibits excellent transferability across various

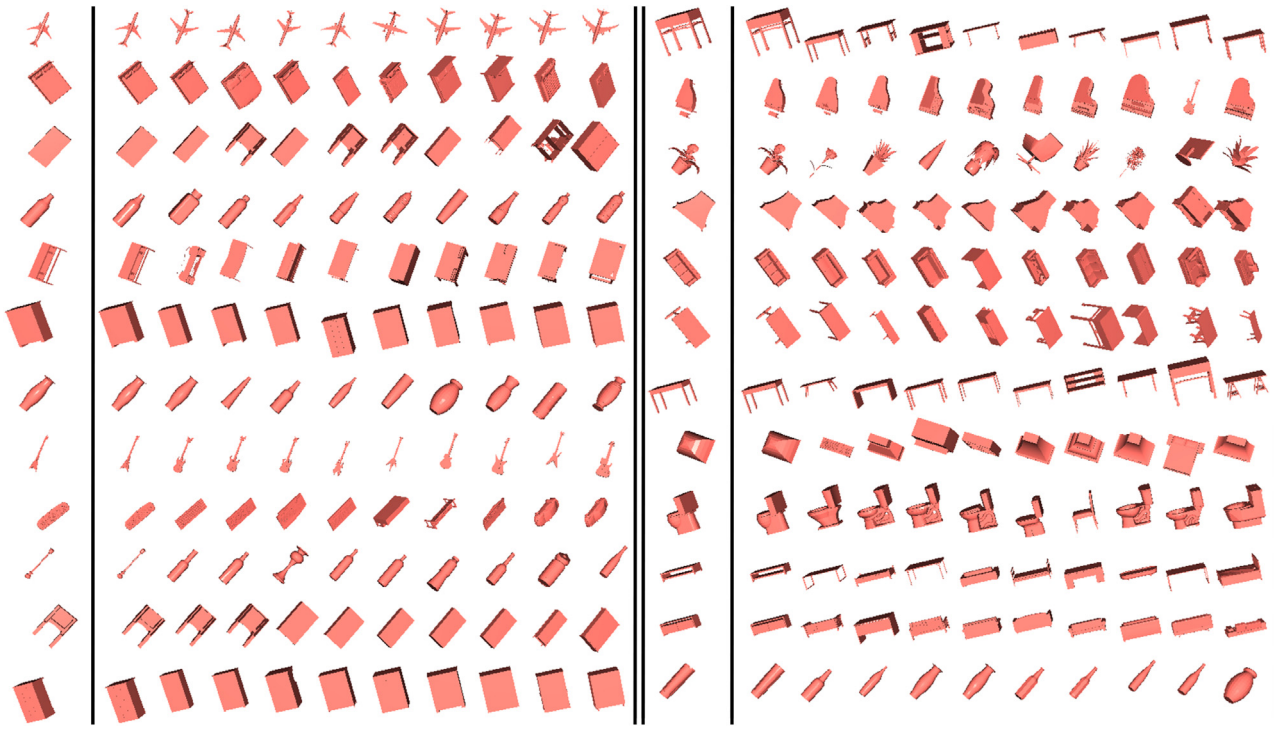


Fig. 9. Retrieval Results on ModelNet40: This figure showcases the retrieval effectiveness of our method on the ModelNet40 dataset. Each column displays the results for a single query object (leftmost column). The top 10 most similar objects from the database are presented to the right of the query, ranked from closest (left) to least close (right). Both halves depict 12 different queries. The models are visualized in random orientations chosen to best highlight their features based on lighting and object structure. These results are specifically selected to summarize the overall performance of our method on ModelNet40. Additional retrieval examples can be found in the supplementary material.

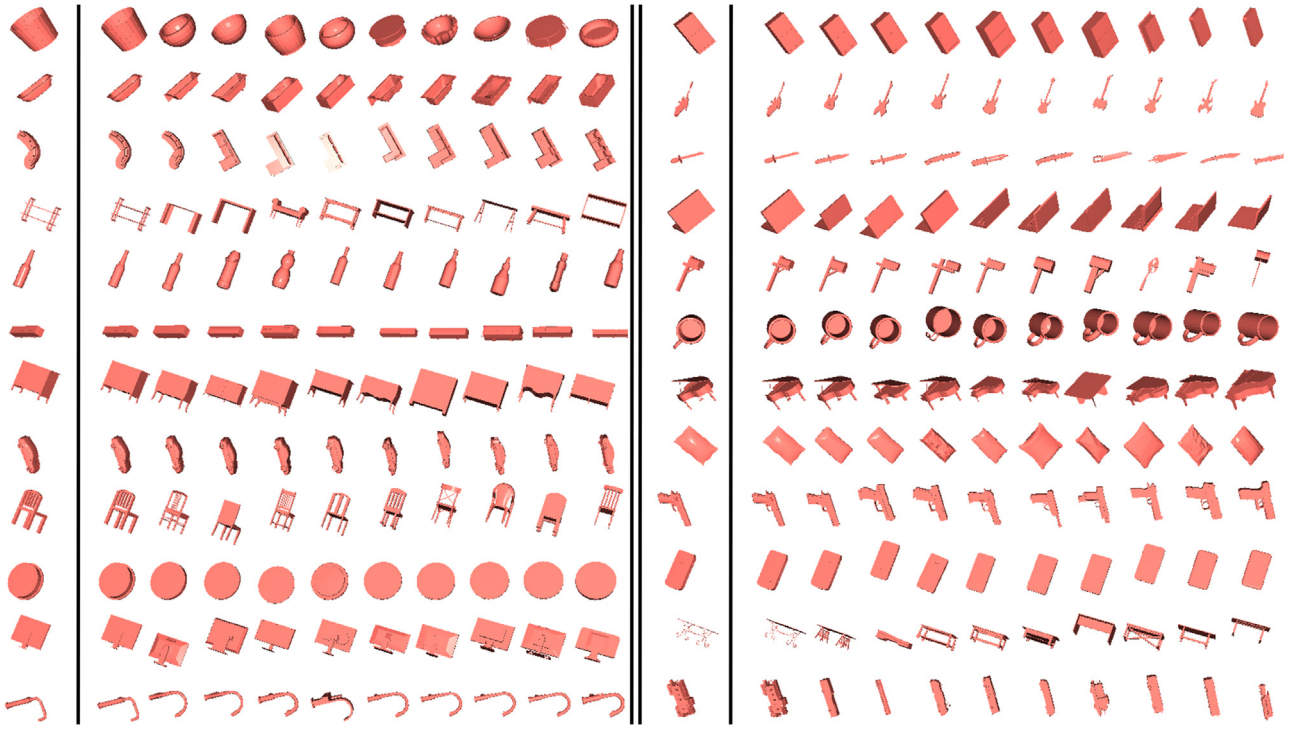


Fig. 10. Retrieval Results on Shapenet: This figure showcases the retrieval effectiveness of our method on the Shapenet dataset similar to Fig. 9.

datasets, including those significantly larger than the training set, such as ShapeNet and Objaverse LVIS. This demonstrates the effectiveness in capturing generalizable features and its potential for real-world applications in 3D model search and retrieval tasks.

Transformers have gained popularity in computer vision, including 3D learning, particularly for generative tasks where they operate on sequences of tokens. However, their application to discriminative tasks – such as those requiring compact and global shape embeddings

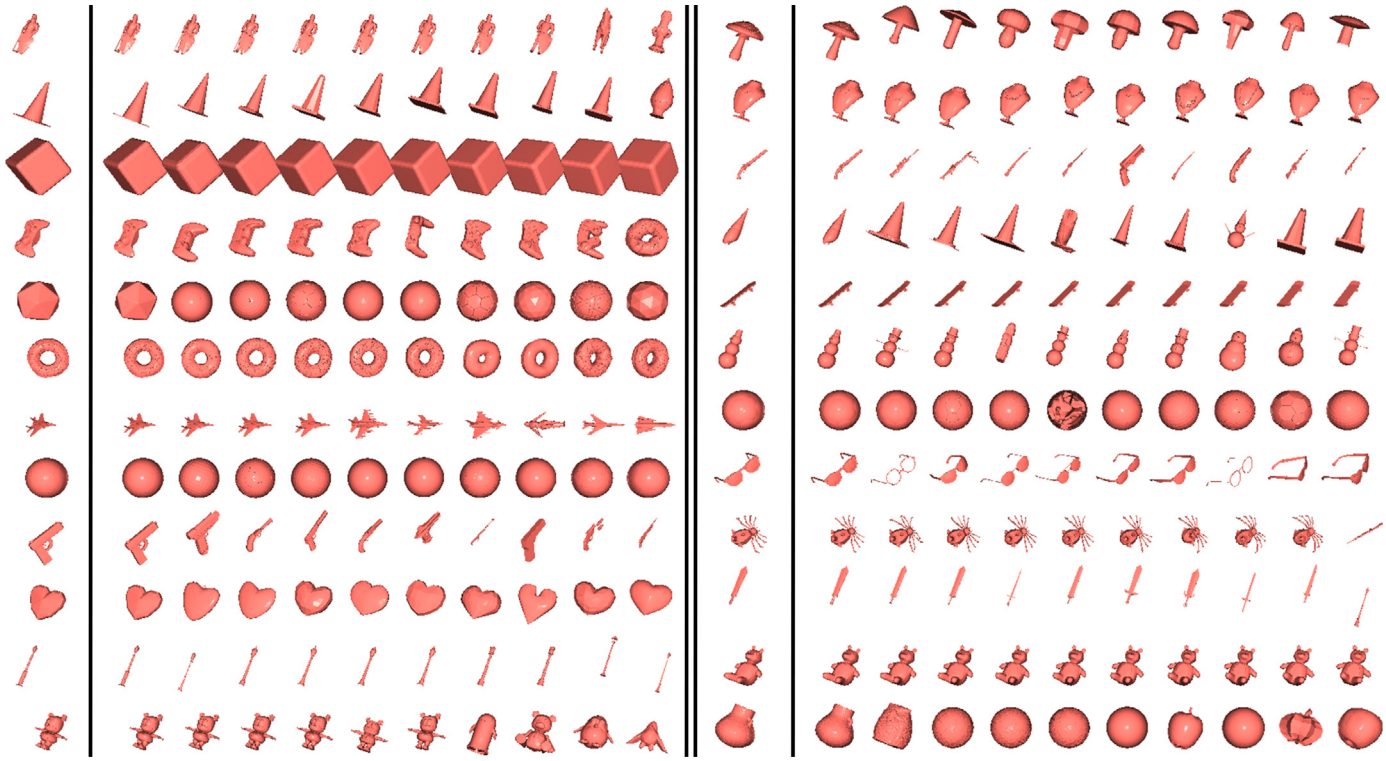


Fig. 11. Retrieval Results on Objaverse LVIS (Deitke et al., 2023): This figure showcases the retrieval effectiveness of our method on the Objaverse LVIS dataset similar to Fig. 9.

– is less direct, often resulting in high-dimensional representations and significant computational overhead. A key contribution of our work is demonstrating that a lightweight MLP-based encoder–decoder architecture can achieve effective performance for 3D shape retrieval.

Future work will focus on incorporating scale information into the retrieval process to address specific use cases where scale is critical. Additionally, we will explore techniques to enhance robustness against triangle permutations within mesh representations, further improving the overall flexibility and generalizability of the approach.

CRediT authorship contribution statement

Kajal Sanklecha: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Prayushi Mathur:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Formal analysis, Data curation. **P.J. Narayanan:** Validation, Supervision, Project administration.

Declaration of competing interest

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

Acknowledgments

We would like to extend our gratitude to Siddharth Mangipudi, Bhoomi Sharma, Sahishnavi Bhartipudi, Shambhavi Ojha, Aditya Sharma, Harshit Karwal, Adnan Qidwai, Akshit Kumar and Anjali Sinha for their invaluable assistance in selecting the insightful results presented in this paper given the scale of the results generated. Their contributions significantly enhanced the clarity and effectiveness of the presented findings.

Supplementary material

Please refer our supplementary materials for a more result's illustration and deeper exploration of the findings' implications.

Supplementary Material : Link

Data availability

Data will be made available on request.

References

Abolghasemi, A., Verberne, S., Azzopardi, L., 2022. Improving BERT-based query-by-document retrieval with multi-task optimization. In: Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part II. Springer, pp. 3–12.

Adhikari, A., Ram, A., Tang, R., Lin, J., 2019. Docbert: Bert for document classification. arXiv preprint arXiv:1904.08398.

Amati, G., Van Rijsbergen, C.J., 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Trans. Inf. Syst. (TOIS) 20 (4), 357–389.

Baevski, A., Zhou, Y., Mohamed, A., Auli, M., 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. Adv. Neural Inf. Process. Syst. 33, 12449–12460.

Bai, S., Bai, X., Zhou, Z., Zhang, Z., Jan Latecki, L., 2016. Gift: A real-time and scalable 3d shape search engine. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5023–5032.

Bharti, A., Vineeth, N., Jawahar, C., 2020. Few shot learning with no labels. arXiv preprint arXiv:2012.13751.

Brin, S., Page, L., 1998. The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst. 30 (1–7), 107–117.

Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M., 2011. Shape google: Geometric words and expressions for invariant shape retrieval. ACM Trans. Graph. 30 (1), 1–20.

Bronstein, M.M., Kokkinos, I., 2010. Scale-invariant heat kernel signatures for non-rigid shape recognition. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, pp. 1704–1711.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G., 2005. Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine Learning. pp. 89–96.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A., 2020. Unsupervised learning of visual features by contrasting cluster assignments. Adv. Neural Inf. Process. Syst. 33, 9912–9924.

Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F., 2015. ShapeNet: An Information-Rich 3D Model Repository. Technical Report, Stanford University — Princeton University — Toyota Technological Institute at Chicago, arXiv:1512.03012 [cs.GR].

Chen, S., Duan, C., Yang, Y., Li, D., Feng, C., Tian, D., 2019. Deep unsupervised learning of 3D point clouds via graph topology inference and filtering. IEEE Trans. Image Process. 29, 3183–3198.

Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020. A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning. PMLR, pp. 1597–1607.

Chen, H., Luo, S., Gao, X., Hu, W., 2021. Unsupervised learning of geometric sampling invariant representations for 3d point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 893–903.

Chen, D.Y., Tian, X.P., Shen, Y.T., Ouhyoung, M., 2003. On visual similarity based 3D model retrieval. In: Computer Graphics Forum, vol. 22, (no. 3), Wiley Online Library, pp. 223–232.

Clark, K., Luong, M.T., Le, Q.V., Manning, C.D., 2020. Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555.

Dai, A.M., Olah, C., Le, Q.V., 2015. Document embedding with paragraph vectors. arXiv preprint arXiv:1507.07998.

Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A., 2023. Objaverse: A universe of annotated 3d objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13142–13153.

Dumais, S.T., et al., 2004. Latent semantic analysis. Annu. Rev. Inf. Sci. Technol. 38 (1), 188–230.

Eitz, M., Richter, R., Boubekeur, T., Hildebrand, K., Alexa, M., 2012. Sketch-based shape retrieval. ACM Trans. Graph. 31 (4), 1–10.

Feng, Y., Feng, Y., You, H., Zhao, X., Gao, Y., 2019. Meshnet: Mesh neural network for 3d shape representation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, (no. 01), pp. 8279–8286.

Gao, Z., Xue, H., Wan, S., 2020. Multiple discrimination and pairwise CNN for view-based 3D object retrieval. Neural Netw. 125, 290–302.

Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al., 2020. Bootstrap your own latent-a new approach to self-supervised learning. Adv. Neural Inf. Process. Syst. 33, 21271–21284.

Hamdi, A., Giancola, S., Ghanem, B., 2021. Mvtn: Multi-view transformation network for 3d shape recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1–11.

Han, Z., Liu, Z., Vong, C.M., Liu, Y.S., Bu, S., Han, J., Chen, C.P., 2018. Deep spatiality: Unsupervised learning of spatially-enhanced global and local 3D features by deep neural network with coupled softmax. IEEE Trans. Image Process. 27 (6), 3049–3063.

Hanocka, R., Hertz, A., Fish, N., Giryes, R., Fleishman, S., Cohen-Or, D., 2019. Meshcnn: a network with an edge. ACM Trans. Graph. 38 (4), 1–12.

He, X., Bai, S., Chu, J., Bai, X., 2020a. An improved multi-view convolutional neural network for 3D object retrieval. IEEE Trans. Image Process. 29, 7917–7930.

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., 2020b. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738.

He, X., Huang, T., Bai, S., Bai, X., 2019. View n-gram network for 3d object retrieval. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7515–7524.

Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y., 2018. Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670.

Huang, J., Su, H., Guibas, L., 2018. Robust watertight manifold surface generation method for shapenet models. arXiv preprint arXiv:1802.01698.

Jégou, H., Douze, M., Schmid, C., Pérez, P., 2010. Aggregating local descriptors into a compact image representation. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, pp. 3304–3311.

Jiang, J., Bao, D., Chen, Z., Zhao, X., Gao, Y., 2019. MLVCNN: Multi-loop-view convolutional neural network for 3D shape retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, (no. 01), pp. 8513–8520.

Johnson, A.E., 1997. Spin-Images: A Representation for 3-D Surface Matching. Citeseer.

Kazhdan, M., Funkhouser, T., Rusinkiewicz, S., 2003. Rotation invariant spherical harmonic representation of 3 D shape descriptors. In: Symposium on Geometry Processing, vol. 6, pp. 156–164.

Kim, S., Chae, D.K., 2022. ExMeshCNN: An explainable convolutional neural network architecture for 3D shape analysis. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 795–803.

Knopp, J., Prasad, M., Willems, G., Timofte, R., Van Gool, L., 2010. Hough transform and 3D SURF for robust three dimensional classification. In: Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part VI 11. Springer, pp. 589–602.

Lavoué, G., Wolf, C., 2008. Markov random fields for improving 3D mesh analysis and segmentation. In: 3DOR@ Eurographics. pp. 25–32.

Lei, H., Akhtar, N., Mian, A., 2021. Picasso: A CUDA-based library for deep learning over 3d meshes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13854–13864.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018. Pointcnn: Convolution on x-transformed points. Adv. Neural Inf. Process. Syst. 31.

Li, B., Lu, Y., Ghumman, A., Strylowski, B., Gutierrez, M., Sadiq, S., Forster, S., Feola, N., Bugerin, T., 2015. 3D sketch-based 3D model retrieval. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. pp. 555–558.

Liang, Y., Zhao, S., Yu, B., Zhang, J., He, F., 2022. Meshmae: Masked autoencoders for 3D mesh data analysis. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III. Springer, pp. 37–54.

Lin, D., Cheng, Y., Guo, A., Mao, S., Li, Y., 2024. SCA-PVNet: Self-and-cross attention based aggregation of point cloud and multi-view for 3D object retrieval. Knowl.-Based Syst. 296, 111920.

Liu, Y., Fan, B., Meng, G., Lu, J., Xiang, S., Pan, C., 2019a. Densepoint: Learning densely contextual representation for efficient point cloud processing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5239–5248.

Liu, X., Han, Z., Wen, X., Liu, Y.S., Zwicker, M., 2019b. L2g auto-encoder: Understanding point clouds by local-to-global reconstruction with hierarchical self-attention. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 989–997.

Liu, M., Shi, R., Kuang, K., Zhu, Y., Li, X., Han, S., Cai, H., Porikli, F., Su, H., 2023. OpenShape: Scaling up 3D shape representation towards open-world understanding.

Manda, B., Dhayarkar, S., Mitheran, S., Viekash, V., Muthuganapathy, R., 2021. 'CADSketchNet'-An annotated sketch dataset for 3D CAD model retrieval with deep neural networks. Comput. Graph..

Nie, W., Liang, Q., Wang, Y., Wei, X., Su, Y., 2020. MMFN: Multimodal information fusion networks for 3D model classification and retrieval. ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) 16 (4), 1–22.

Nie, W., Zhao, Y., Song, D., Gao, Y., 2021. DAN: Deep-attention network for 3D shape recognition. IEEE Trans. Image Process. 30, 4371–4383.

Oord, A.v.d., Li, Y., Vinyals, O., 2018. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

Peng, B., Yu, Z., Lei, J., Song, J., 2020. Attention-guided fusion network of point cloud and multiple views for 3D shape recognition. In: 2020 IEEE International Conference on Visual Communications and Image Processing. VCIP, IEEE, pp. 185–188.

Perronnin, F., Sánchez, J., Mensink, T., 2010. Improving the fisher kernel for large-scale image classification. In: Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part IV 11. Springer, pp. 143–156.

Qi, A., Song, Y.Z., Xiang, T., 2018. Semantic embedding for sketch-based 3D shape retrieval. In: BMVC, vol. 3, pp. 11–12.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Adv. Neural Inf. Process. Syst. 30.

Radenović, F., Tolias, G., Chum, O., 2018. Fine-tuning CNN image retrieval with no human annotation. IEEE Trans. Pattern Anal. Mach. Intell. 41 (7), 1655–1668.

Robertson, S., Zaragoza, H., et al., 2009. The probabilistic relevance framework: BM25 and beyond. Found. Trends® Inf. Retr. 3 (4), 333–389.

Sanklecha, K., Mathur, P., Narayanan, P., 2023. Self-supervised 3D mesh object retrieval. In: Proceedings of the Fourteenth Indian Conference on Computer Vision, Graphics and Image Processing. pp. 1–10.

Scovanner, P., Ali, S., Shah, M., 2007. A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of the 15th ACM International Conference on Multimedia. pp. 357–360.

Sharma, A., Horaud, R., Cech, J., Boyer, E., 2011. Topologically-robust 3D shape matching based on diffusion geometry and seed growing. In: CVPR 2011. pp. 2481–2488. http://dx.doi.org/10.1109/CVPR.2011.5995455.

Sharma, C., Kaul, M., 2020. Self-supervised few-shot learning on point clouds. Adv. Neural Inf. Process. Syst. 33, 7212–7221.

Shekhar, R., Jawahar, C., 2012. Word image retrieval using bag of visual words. In: 2012 10th IAPR International Workshop on Document Analysis Systems. IEEE, pp. 297–301.

Shi, Z., Quan, Z., Shi, J., Guo, Z., Zhang, M., Xiao, Z., 2022. 3D model retrieval algorithm based on attention and multi-view fusion. In: Proceedings of the 5th International Conference on Computer Science and Software Engineering. pp. 474–480.

Shi, Y., Xu, M., Yuan, S., Fang, Y., 2020. Unsupervised deep shape descriptor with point distribution learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9353–9362.

Singh, V.V., Sheshappanavar, S.V., Kambhamettu, C., 2021. MeshNet++: A network with a face. In: ACM Multimedia. pp. 4883–4891.

Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E., 2015. Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 945–953.

Sun, J., Ovsjanikov, M., Guibas, L., 2009. A concise and provably informative multi-scale signature based on heat diffusion. In: Computer Graphics Forum, vol. 28, (no. 5), Wiley Online Library, pp. 1383–1392.

Taigman, Y., Yang, M., Ranzato, M., Wolf, L., 2014. DeepFace: Closing the gap to human-level performance in face verification. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1701–1708. http://dx.doi.org/10.1109/CVPR.2014.220.

Thabet, A., Alwassel, H., Ghanem, B., 2020. Self-supervised learning of local features in 3d point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 938–939.

van Blokland, B.I., Theoharis, T., 2018. Microshapes: efficient querying of 3D object collections based on local shape. In: Proceedings of the 11th Eurographics Workshop on 3D Object Retrieval. pp. 9–16.

van Blokland, B.I., Theoharis, T., 2020. An indexing scheme and descriptor for 3D object retrieval based on local shape querying. Comput. Graph. 92, 55–66.

Wang, F., Kang, L., Li, Y., 2015. Sketch-based 3d shape retrieval using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1875–1883.

Wang, L., Li, Y., Lazebnik, S., 2016. Learning deep structure-preserving image-text embeddings. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5005–5013.

Wang, H., Wang, X., Xiong, W., Yu, M., Guo, X., Chang, S., Wang, W.Y., 2019. Self-supervised learning for contextualized extractive summarization. arXiv preprint arXiv:1906.04466.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1912–1920.

Wu, Z., Xiong, Y., Yu, S.X., Lin, D., 2018. Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3733–3742.

Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C., Su, H., Mottaghi, R., Guibas, L., Savarese, S., 2016. Objectnet3d: A large scale database for 3d object recognition. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII 14. Springer, pp. 160–176.

Xie, J., Dai, G., Zhu, F., Wong, E.K., Fang, Y., 2016. Deepshape: Deep-learned shape descriptor for 3d shape retrieval. IEEE Trans. Pattern Anal. Mach. Intell. 39 (7), 1335–1345.

Xue, L., Gao, M., Xing, C., Martín-Martín, R., Wu, J., Xiong, C., Xu, R., Niebles, J.C., Savarese, S., 2023a. ULIP: Learning a unified representation of language, images, and point clouds for 3D understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR.

Xue, L., Yu, N., Zhang, S., Li, J., Martín-Martín, R., Wu, J., Xiong, C., Xu, R., Niebles, J.C., Savarese, S., 2023b. ULIP-2: Towards scalable multimodal pre-training for 3D understanding.

Yates, A., Nogueira, R., Lin, J., 2021. Pretrained transformers for text ranking: BERT and beyond. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. pp. 1154–1156.

You, H., Feng, Y., Ji, R., Gao, Y., 2018. Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition. In: Proceedings of the 26th ACM International Conference on Multimedia. pp. 1310–1318.

You, H., Feng, Y., Zhao, X., Zou, C., Ji, R., Gao, Y., 2019. PVRNet: Point-view relation neural network for 3D shape recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, (no. 01), pp. 9119–9126.

Zhang, Z., Girdhar, R., Joulin, A., Misra, I., 2021. Self-supervised pretraining of 3d features on any point-cloud. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10252–10263.

Zhou, H., Li, J., Liu, X., Lyu, Y., Chen, H., Liu, A.A., 2024. View sequence prediction GAN: unsupervised representation learning for 3D shapes by decomposing view content and viewpoint variance. Multimedia Syst. 30 (4), 1–12.