



## TextANIMAR: Text-Based 3D Animal Fine-Grained Retrieval

Trung-Nghia Le<sup>a,b</sup>, Tam V. Nguyen<sup>b,c</sup>, Minh-Quan Le<sup>b,a,b</sup>, Trong-Thuan Nguyen<sup>b,a,b</sup>, Viet-Tham Huynh<sup>a,b</sup>, Trong-Le Do<sup>b,a,b</sup>, Khanh-Duy Le<sup>a,b</sup>, Mai-Khiem Tran<sup>b,a,b</sup>, Nhat Hoang-Xuan<sup>b,a,b</sup>, Thang-Long Nguyen-Ho<sup>b,a,b</sup>, Vinh-Tiep Nguyen<sup>b,d,b</sup>, Tuong-Nghiem Diep<sup>a,b</sup>, Khanh-Duy Ho<sup>a,b</sup>, Xuan-Hieu Nguyen<sup>a,b</sup>, Thien-Phuc Tran<sup>a,b</sup>, Tuan-Anh Yang<sup>a,b</sup>, Kim-Phat Tran<sup>a,b</sup>, Nhu-Vinh Hoang<sup>a,b</sup>, Minh-Quang Nguyen<sup>a,b</sup>, E-Ro Nguyen<sup>a,b</sup>, Minh-Khoi Nguyen-Nhat<sup>a,b</sup>, Tuan-An To<sup>a,b</sup>, Trung-Truc Huynh-Le<sup>a,b</sup>, Nham-Tan Nguyen<sup>a,b</sup>, Hoang-Chau Luong<sup>a,b</sup>, Truong Hoai Phong<sup>a,b</sup>, Nhat-Quynh Le-Pham<sup>a,b</sup>, Huu-Phuc Pham<sup>a,b</sup>, Trong-Vu Hoang<sup>a,b</sup>, Quang-Binh Nguyen<sup>a,b</sup>, Hai-Dang Nguyen<sup>b,a,b</sup>, Akihiro Sugimoto<sup>e</sup>, Minh-Triet Tran<sup>b,a,b,\*</sup>

<sup>a</sup>University of Science, VNU-HCM, Ho Chi Minh City, Vietnam

<sup>b</sup>Vietnam National University, Ho Chi Minh City, Vietnam

<sup>c</sup>University of Dayton, Ohio, U.S.

<sup>d</sup>University of Information Technology, VNU-HCM, Ho Chi Minh City, Vietnam

<sup>e</sup>National Institute of Informatics, Tokyo, Japan

### ARTICLE INFO

#### Article history:

Received August 10, 2023

3D object retrieval, fine-grained retrieval, and animal models.

### ABSTRACT

3D object retrieval is an important yet challenging task that has drawn more and more attention in recent years. While existing approaches have made strides in addressing this issue, they are often limited to restricted settings such as image and sketch queries, which are often unfriendly interactions for common users. In order to overcome these limitations, this paper presents a novel SHREC challenge track focusing on text-based fine-grained retrieval of 3D animal models. Unlike previous SHREC challenge tracks, the proposed task is considerably more challenging, requiring participants to develop innovative approaches to tackle the problem of text-based retrieval. Despite the increased difficulty, we believe this task can potentially drive useful applications in practice and facilitate more intuitive interactions with 3D objects. Five groups participated in our competition, submitting a total of 114 runs. While the results obtained in our competition are satisfactory, we note that the challenges presented by this task are far from fully solved. As such, we provide insights into potential areas for future research and improvements. We believe we can help push the boundaries of 3D object retrieval and facilitate more user-friendly interactions via vision-language technologies.

© 2023 Elsevier B.V. All rights reserved.

### 1. Introduction

The rapid growth and advancement of 3D technologies have significantly expanded the availability and abundance of 3D objects. As a result, 3D object retrieval has emerged as a prominent and interesting research area, with substantial practical applications [1, 2, 3, 4, 5] across diverse domains, including video

games, creative arts, motion picture production, and virtual reality.

In real-world scenarios, obtaining a 3D model as a query typically demands significant effort and resources. As a solution, content-based 3D object retrieval techniques [6, 7, 8] have been developed, which provide a more accessible approach to query collection. These techniques aim to retrieve 3D objects from a database based on their visual content, encompassing color, texture, shape, and geometric features. Among the various retrieval methods, image-based and sketch-based

\*Corresponding author

e-mail: [tmtriet@fit.hcmus.edu.vn](mailto:tmtriet@fit.hcmus.edu.vn) (Minh-Triet Tran)

approaches [9, 10, 11, 12] have gained popularity. Image-based retrieval methods leverage RGB images captured from the real world to extract relevant visual features, facilitating the retrieval of similar 3D models. This approach is advantageous as it allows users to access valuable 3D models conveniently through readily available 2D images. In contrast, sketch-based retrieval [13, 14, 15, 16] employs hand-drawn sketches as queries. The intuitive nature of freehand drawings allows for a more effective capture of the essential features of 3D objects while filtering out irrelevant information. Nevertheless, image-based and sketch-based approaches introduce notable challenges in 3D object retrieval research. The substantial disparities between 2D and 3D modalities present a significant obstacle, as 2D images or sketches differ considerably from their corresponding 3D counterparts and perspectives. Moreover, sketches are prone to ambiguity and errors, which can detrimentally affect the accuracy of the retrieval process.

We have introduced a novel challenge track called *Text-based 3D ANIMAL model fine-grained Retrieval (TextANIMAR)*<sup>1</sup> aiming to enhance the effectiveness of content-based 3D object retrieval. The primary objective of this track is to retrieve relevant 3D animal models from a dataset using textual queries. This SHREC challenge track poses significantly greater challenges and provides a more effective simulation of real-life scenarios than previous SHREC challenge tracks. We also note that after the challenge concluded, the dataset has been made publicly available for academic purposes.

Firstly, in conventional 3D object retrieval tasks, the primary focus is typically on the object category. These approaches often involve training and testing with samples from the same category. While this leads to feature extraction methods specialized for known categories, it may limit their effectiveness with unseen categories in practice. Their efficacy should be evaluated only within specific contexts. Nevertheless, alternative approaches, such as open-set 3D object retrieval, have emerged as promising solutions for effectively addressing the retrieval of unseen categories. These approaches involve training models on known-category 3D objects and incorporating unseen-category data, offering potential avenues to overcome the challenges posed by classification-based methods. Regardless, our fine-grained retrieval task requires participants to accurately search 3D animal models whose shapes correspond to a given query, necessitating consideration of unseen categories and poses (*cf.* Table 1). This task poses a more significant challenge than traditional category-based retrieval, which requires handling the substantial discrepancies in animal breeds and poses.

Second, the quality and resolution of the input image can impact the performance of image-based 3D object retrieval. However, controlling these factors requires additional effort. It is also worth noting that image queries may encounter challenges in effectively handling variations in object scale, orientation, and perspective, potentially impacting retrieval performance. Conversely, sketches trained on existing datasets often exhibit semi-photorealistic qualities and are expertly created, posing

Table 1: SHREC challenge tracks for 3D object retrieval.

| SHREC Challenge               | Year | Query Type | Training Category | Testing Category |
|-------------------------------|------|------------|-------------------|------------------|
| Pratikakis <i>et al.</i> [17] | 2016 | 3D Shape   | Seen              | Seen             |
| Sipiran <i>et al.</i> [18]    | 2021 | 3D Shape   | Seen              | Seen             |
| Juefei <i>et al.</i> [19]     | 2018 | Sketch     | Seen              | Seen             |
| Juefei <i>et al.</i> [20]     | 2019 | Sketch     | Seen              | Seen             |
| Qin <i>et al.</i> [13]        | 2022 | Sketch     | Seen              | Seen             |
| Hameed <i>et al.</i> [21]     | 2018 | Image      | Seen              | Seen             |
| Hameed <i>et al.</i> [22]     | 2019 | Image      | Seen              | Seen             |
| Li <i>et al.</i> [23]         | 2019 | Image      | Seen              | Seen             |
| Li <i>et al.</i> [24]         | 2020 | Image      | Seen              | Seen             |
| Feng <i>et al.</i> [25]       | 2022 | Image      | Seen              | Unseen           |
| TextANIMAR                    | 2023 | Text       | Unseen            | Unseen           |

challenges for regular users to reproduce them in real-world scenarios. Last but not least, text-based queries are considerably easier to generate than image capture or sketching, making them a more user-friendly alternative. We anticipate that the text-based 3D animal fine-grained retrieval task will stimulate new research directions and find practical applications.

The structure of this paper is as follows. Section 2 discusses the literature review and previous works relevant to 3D object retrieval. Section 3 presents the ANIMAR dataset and the evaluation metrics used in this SHREC challenge track. The participant statistics are reported in Section 4. In Section 5, we describe the methods employed by the participating teams. Section 6 contains the evaluation results, including a detailed analysis of the performance of the different methods. Finally, in Section 7, we summarize the key points of the paper and discuss the implications for future research in this field.

## 2. Related Benchmark

Content-based 3D object retrieval aims to retrieve 3D objects from a database by analyzing the visual contents of the objects, including color, texture, shape, and geometric features. In order to facilitate research in this field, previous SHREC challenges have included various tracks dedicated to related tasks (see Table 1).

Few SHREC tracks focus on retrieving 3D objects from a database similar in shape to a given query 3D objects. Pratikakis *et al.* [17] introduced the concept of partial 3D object retrieval, which addresses scenarios where the available information about the query object is incomplete. These techniques help build digital libraries of cultural heritage objects, which require partial 3D object retrieval capabilities. In a related direction, Sipiran *et al.* [18] also held a competition to evaluate the ability of retrieval methods to discriminate cultural heritage objects by overall shape.

In contrast, the appeal of sketch-based 3D object retrieval was based on the natural and intuitive nature of freehand sketches and has attracted significant attention in recent years. This area of research has been actively promoted through SHREC tracks organized by Juefei *et al.* [19, 20] focused on 2D scene sketch-based 3D scene retrieval. To address the domains shift between the sketch and 3D object, domain adaptation (e.g.,

<sup>1</sup><https://aichallenge.hcmus.edu.vn/textanimar>

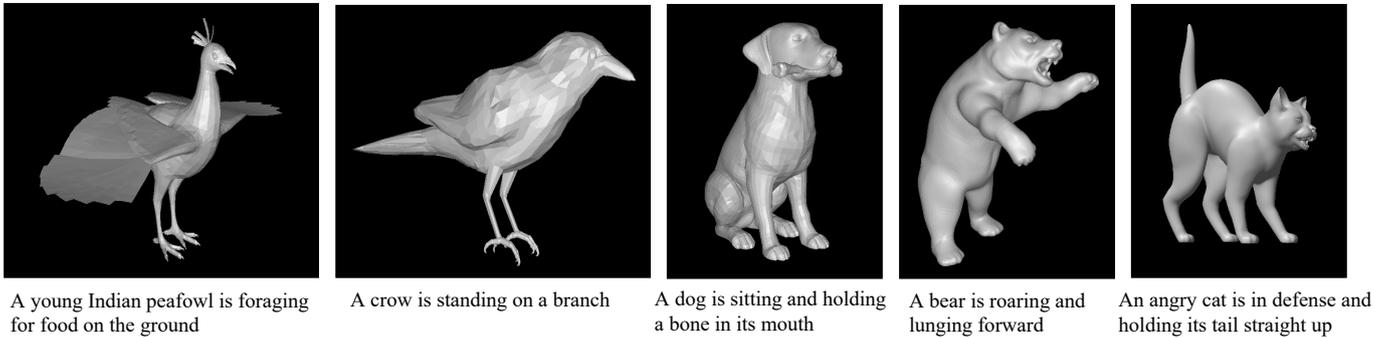


Fig. 1: Text ground-truth in ANIMAR dataset, including the text query and the corresponding 3D animal model.

two-stream CNN with triplet loss, adversarial training, and various data augmentation methods) was employed. In addition, Qin *et al.* [13] further advanced the task by organizing a competition for sketch-based 3D shape retrieval in real-world settings. This competition involved a large-scale collection of sketches drawn by amateurs with varying levels of drawing skills as well as a diverse set of 3D shapes, including models scanned from natural objects. Solutions were developed to simulate realistic retrieval scenarios, incorporating techniques like point cloud and multi-view learning using different deep learning architectures.

Image-based approaches have dominated the field of content-based 3D object retrieval. The SHREC competitions organized by Hameed *et al.* [21, 22] have significantly advanced 3D scene retrieval from 2D scene image queries. These methods capture different views of 3D scenes for feature learning, incorporating saliency algorithms to select the most promising views for each 3D model. Feature extraction techniques such as Bag of Visual Words have been employed for extracting features from 2D images. Furthermore, VGG, ResNet50, Two-stream CNN, and Conditional Variational autoencoder combined data augmentation demonstrate their effectiveness in this task. Li *et al.* [23, 24] organized SHREC tracks focused on searching for relevant 3D everyday objects using monocular images captured in real-world settings. In these competitions, various deep learning architectures were utilized to learn captured 2D views of 3D objects.

In conventional 3D object retrieval tasks, all 3D models are categorized, which may not fully capture the diversity present in real-world objects. To address this limitation, recent SHREC tracks proposed by Feng *et al.* [25] have evaluated the performance of different retrieval algorithms under the open-set setting and modality-missing setting. The submitted methods, such as multi-modal learning, have shown promising results in retrieving 3D objects from unknown categories, where the retrieval sets include categories not seen in the training set. However, the open-set retrieval setting still needs to fully simulate the real world when models are trained on known categories. Different from other 3D object retrieval tasks, our TextANIMAR competition stands out by fully simulating real-world scenarios. This is accomplished by utilizing unseen categories for both the train and test 3D objects, providing a more challenging and realistic evaluation setting.

A **female mandrill** is **climbing out the top of the tree.**  
Description Context

Fig. 2: The text query comprises two main components: a description of the animal and a context.

### 3. Dataset and Evaluation

#### 3.1. Dataset

In this competition, we constructed a new dataset, namely ANIMAR, which encompasses a corpus of 711 distinct 3D animal models along with 150 text queries.

We collected 186 mesh models representing over 50 diverse animal categories from Planet Zoo<sup>2</sup> [26], a publicly available online resource and video games. Our main objective for this competition track is to imitate real-world scenarios where users seek to explore and identify various types of animals. To this end, we intentionally concealed categorical information throughout the training and retrieval stages. Additionally, we created a simplified set of watertight mesh models by reducing the number of faces by 25%, 50%, and 75%, resulting in a total of 525 models. Following the approach of Douze *et al.* [27], our 3D animal model database is utilized for both the training and retrieval phases.

We manually curated 150 English sentences, each incorporating two fundamental constituents: an explicit depiction of the animal's natural shape, focusing on breed-specific attributes, and a context-driven description to match the desired pose for model utilization (see Fig. 2). For instance, when searching for a tiger model suitable for a hunting action, the corresponding description, "a tiger is hunting," ensures the retrieved model possesses the most appropriate pose. Furthermore, we emphasize searching single-animal models where only single-animal descriptions are utilized to optimize the search process. Leveraging these context-aware descriptions enables more accurate and efficient retrieval of 3D animal models, fostering an enhanced user experience in virtual environments. We expect this to facilitate interactive search, whereby users can effortlessly explore and identify 3D animal models based on their species, actions, or even environmental contexts.

<sup>2</sup><https://www.planetzoogame.com>

In our dataset, 100 sentences are aligned with their corresponding models in the database resulting in 382 pairs of query-model for training, while the remaining 50 sentences are utilized as queries, corresponding to 188 pairs of query-model, during the retrieval phase. Figure 1 illustrates examples in our ANIMAR dataset, including the text queries and the corresponding 3D animal models.

### 3.2. Evaluation Metrics

We provide a comprehensive evaluation of the performance of different methods in this track. The following metrics are utilized:

- **Nearest Neighbor (NN)** evaluates top-1 retrieval accuracy.
- **Precision-at-10 (P@10)** is the ratio of relevant items in the top-10 returned results.
- **Normalized Discounted Cumulative Gain (NDCG)** is a measure of ranking quality defined as  $\sum_{i=1}^p \frac{rel_i}{\log_2(i+1)}$ , where  $p$  is the length of the returned rank list, and  $rel_i$  denotes the relevance of the  $i$ -th item.
- **Mean Average Precision (mAP)** is the area under the precision-recall curve that measures the precision of methods at different levels and then takes the average. mAP is calculated as  $\frac{1}{r} \sum_{i=1}^r P(i)(R(i) - R(i-1))$ , where  $r$  is the number of retrieved relevant items,  $P(i)$  and  $R(i)$  are the precision and recall at the position of the  $i^{th}$  relevant item, respectively.
- **First Tier (FT)** indicates the recall of the top  $m$  retrieval results, where  $m$  represents the number of relevant images present in the entire database. It quantifies the accuracy of retrieving the most relevant images among all possible matches. The FT score is calculated by dividing the number of relevant images retrieved in the top  $m$  by  $m$ .
- **Second Tier (ST)** measures the recall of the top  $2m$  retrieval results, where  $m$  represents the number of relevant images in the entire database. It assesses the system's ability to retrieve relevant images within a broader set of results. The ST score is calculated by dividing the number of relevant images retrieved in the top  $2m$  by  $m$ .
- **Fallout Rate (FR)** reflects the ratio of non-relevant retrieved items to the total number of non-relevant items available. It evaluates the system's effectiveness in avoiding the retrieval of non-relevant items. The FR score is calculated using the formula: dividing the number of non-relevant items retrieved by the total number of non-relevant items.

## 4. Participants

Five groups participated in the TextANIMAR challenge track, collectively submitting 114 runs. The contest had a three-week duration for participants to complete their submissions. To participate, each group was required to register and submit

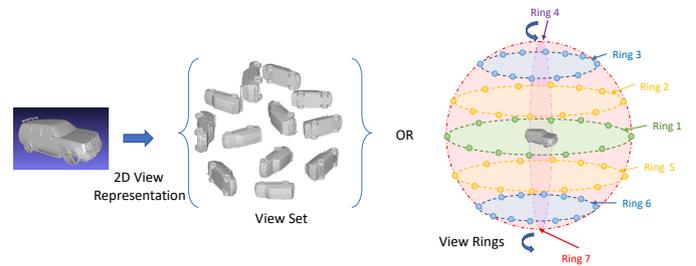


Fig. 3: For the 3D object representation, the set of images generated is  $R = 7$  rings with  $V = 12$  views on each ring. The chosen latitudes were 0 (the equator),  $\pm 90$  (the poles), and  $\pm 30, \pm 60$ .

their results, including a description of the methods employed. It is worth noting that the organizers did not participate in this challenge. Below are the details of the participating groups (team members will be added upon acceptance):

- Polars team submitted by Minh-Khoi Nguyen-Nhat, Tuan-An To, Trung-Truc Huynh-Le, Nham-Tan Nguyen, and Hoang-Chau Luong (see Section 5.2).
- TikTorch team submitted by Nhat-Quynh Le-Pham, Huu-Phuc Pham, Trong-Vu Hoang, Quang-Binh Nguyen, and Hai-Dang Nguyen (see Section 5.3).
- Etinifni team submitted by Tuong-Nghiem Diep, Khanh-Duy Ho, Xuan-Hieu Nguyen, Thien-Phuc Tran, Tuan-Anh Yang, Kim-Phat Tran, Nhu-Vinh Hoang, and Minh-Quang Nguyen (see Section 5.4).
- THP team submitted by Truong Hoai Phong (see Section 5.5).
- Nero team submitted by E-Ro Nguyen (see Section 5.6).

## 5. Methods

### 5.1. Overview of Submitted Solutions

The solutions submitted to our track can be categorized into two distinct groups; each uses different techniques for representing 3D objects. The former group (*i.e.*, model-based learning approach) directly learns point cloud, while the latter group (*i.e.*, view-based learning) captures the 3D object as a set of random images.

The model-based learning approach is exemplified by the Polars team. This approach directly learns point clouds via PointNet [28] and PointMLP [29] to facilitate the representation of 3D animal objects (as shown in Section 5.2).

The view-based learning approach involves the collaboration of multiple teams, namely TikTorch, Etinifni, THP, and Nero. This approach represents each 3D object using a series of ring images, as illustrated in Fig. 3. These images are captured by moving a camera around the object along a predefined path, with each ring including a collection of images. The effectiveness of the multi-view technique is particularly notable when the camera follows a trajectory parallel to the ground plane about the object. This approach provides valuable images for

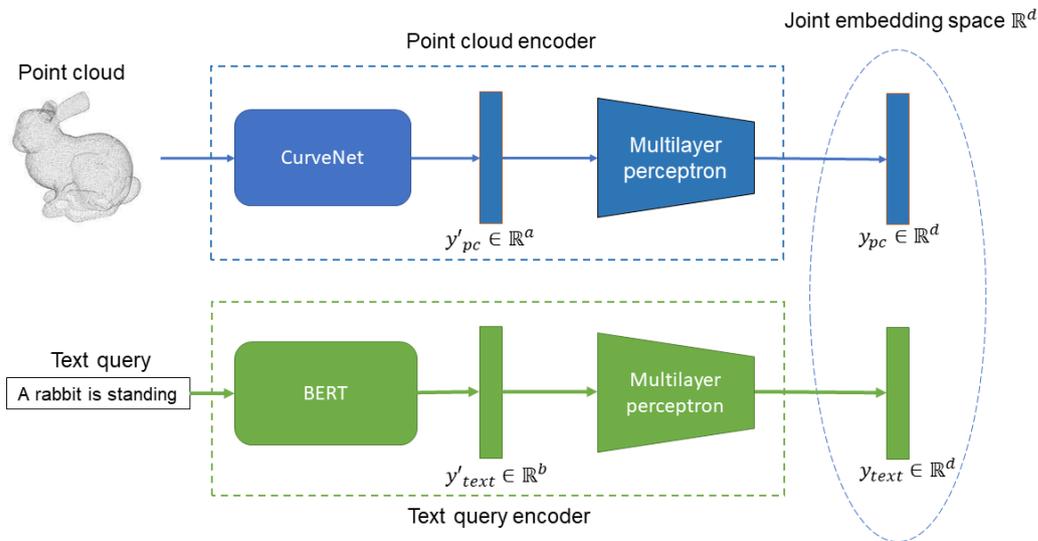


Fig. 4: Overview of proposed text-cloud contrastive learning framework of Polars team.

learning the distinctive features of 3D objects. Although sharing similar concepts, the TikTorch team has developed novel encoders (as depicted in Section 5.3). In contrast, the other teams rely on CLIP models [30] (as shown in Sections 5.4, 5.5, and 5.6) to support their research efforts.

## 5.2. Polars Team

### 5.2.1. Proposed Framework

As illustrated in Fig. 4, their proposed framework consists of a query encode branch and a point cloud encode branch. The former encodes the query in natural language into a vector in a joint embedding space, while the latter does the same work for point cloud input.

The query encoder uses pre-trained BERT [31] to extract the text query's raw embedding feature. The point cloud encoder employs PointNet [28] and CurveNet [32] to extract the raw embedding feature. Each raw embedding feature is then forwarded into a projection module (*i.e.*, multi-layer perceptron) in order to map the feature into a joint latent space  $\mathcal{R}^d$ .

In addition, they employ the InfoNCE loss [33] to enhance the learning representation. Specifically, given a pair of embeddings,  $(y_{text})$  for the text query and  $(y_{pc})$  for the point cloud, they convert the text query index into an integer ( $l$ ). This allows them to generate two positive pairs for optimization:  $(y_{text}, l)$  and  $(y_{pc}, l)$ . By utilizing these positive pairs, they aim to optimize the representation learning process and improve the overall performance of the system.

### 5.2.2. Training Details

The based learning rate was 0.001 and was scheduled by a MultiStepLR at steps 120, 250, 350, and 500, respectively. A target embedding space dimension  $d$  was 128. In training, they froze the BERT parameters and trained on the remaining part of the model.

## 5.3. TikTorch Team

### 5.3.1. Contrastive Learning Solution

Figure 5 illustrates their proposed contrastive learning framework for text-based 3D animal fine-grained retrieval. From two different domains (3D objects and sentences), they try to learn embedding vectors for objects and texts in a common vector space, in which the embedding vectors of similar objects and texts will be closer to each other and vice versa.

To achieve this goal, the team constructs two feature extractors: a 3D Object Feature Extractor and a Text Feature Extractor. These extractors generate two feature vectors, one with  $U$  dimensions and the other with  $V$  dimensions. Subsequently, these feature vectors are embedded in a shared vector space with  $P$  dimensions using two Multi-layer Perceptron (MLP) networks. The contrastive loss [34] is applied to facilitate the simultaneous learning of parameters for both models.

**3D object feature extractor.** Each 3D object is represented by a collection of seven rings, with each ring consisting of 12 images (*cf.* Fig. 3). The team focuses on optimizing the features extracted from the equator ring, as it offers the most significant potential for distinguishing between objects. The extractor module operates a ring extractor extracting the features of the images within each ring and then combining these image features to obtain the overall features of the object.

In the ring extractor, the team performs fine-tuning of EfficientNetV2-Small [35] to extract features from the 12 images within each ring. These extracted feature vectors undergo encoding using the T-Encoder, a part of the Transformer's encoder [36]. The T-Encoder captures the relationship among the images within the same ring, determining their relative significance. Subsequently, the feature vectors are combined by taking their average, resulting in a single feature vector for each ring. After obtaining the feature vectors for the three rings, they pass through two T-encoder blocks. Finally, these feature

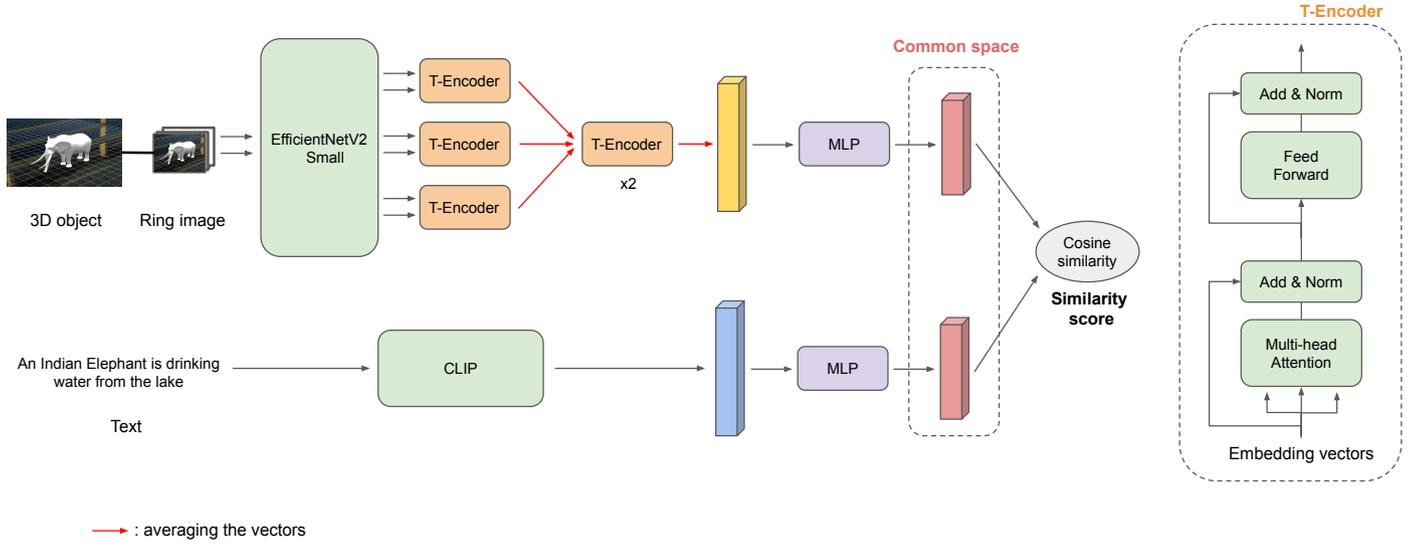


Fig. 5: Proposed contrastive learning solution of TikTorch team.

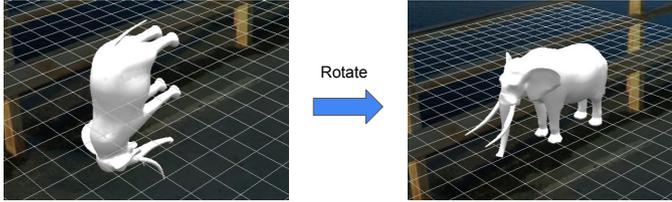


Fig. 6: Example of rotating a 3D object whose axis is not aligned with the majority of objects.

vectors are averaged to derive the overall feature vector representing the 3D object.

#### Data pre-processing.

In order to maintain consistency in the resulting multi-view images and their corresponding camera angles, it is crucial to synchronize the axial orientation of the 3D objects before generating batch images. To accomplish this, the team comprehensively examined the available dataset, identifying several objects rotated at a 90-degree angle along the  $Ox$  axis. Subsequently, these objects are consistently rotated to align with the majority of the dataset, as illustrated in Fig. 6. They capture images of the objects using a camera angle set to capture images from bottom to top (ring 0 to ring 6), as shown in Fig. 3. They find that the most informative views are captured from ring 3, which provides a direct side perspective from the object. Hence, they focus on processing the images from ring 3 to extract the relevant features and information.

**Text feature extractor.** To extract the features of the prompt, they fine-tune the CLIP text encoder [30], a masked self-attention Transformer. This encoder was pre-trained to maximize the similarity of pairs of image and text via a contrastive loss. The models in the CLIP family reduce the parameter size significantly while maintaining competitive accuracy, especially in optimizing text-image similarity tasks.

**Common space embedding.** To calculate the similarity between 3D objects and prompts, the team employs feature vector embedding in a shared space. Since the feature vectors of

3D objects and prompts have different dimensions, they utilize two MLP networks with two layers, where the output layer has the same number of units, to align them in the common vector space. They incorporate a Dropout layer [37] to mitigate overfitting in each network. In this shared space, the similarity between two embedding vectors, denoted as  $\mathbf{u}$  and  $\mathbf{v}$ , is computed using the cosine similarity metric:

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \quad (1)$$

**Loss function.** The team employs the Normalized Temperature-scaled Cross Entropy Loss (NT-Xent) [34] as the contrastive loss function. For a mini-batch of  $2N$  samples  $\{\mathbf{x}_i\}$ , consisting of  $N$  objects and  $N$  queries, they denote  $\mathbf{z}_i$  as the embedding vector of sample  $\mathbf{x}_i$  in the common vector space. Let  $P_i$  be the set of indices of samples that are similar to  $\mathbf{x}_i$  in the current mini-batch, excluding  $i$ , i.e.,  $(\mathbf{x}_i, \mathbf{x}_j)$  forms a positive pair for  $j \in P_i$ . It should be noted that  $\mathbf{x}_i$  can belong to multiple positive pairs, such as when two 3D objects are similar to the same query. The loss function for a positive pair  $(\mathbf{x}_i, \mathbf{x}_j)$  is defined as:

$$l_{i,j} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j) / \tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i, k \notin P_i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k) / \tau)}. \quad (2)$$

**Training phase.** To train the proposed network, they used AdamW [38] optimizer, along with the StepLR, to reduce the learning during the training process. They also applied the  $k$ -fold cross-validation technique with  $k = 5$ .

**Retrieval phase.** They ensemble the results of models trained on  $k$ -fold by majority vote. The similarity between a 3D object and a prompt is the max value of the similarity score computed by the five models.

#### 5.3.2. Data Augmentation

They find that there are animal models in the ANIMAR dataset with similar appearance (e.g., animals in near families, different granularity versions of a model). Therefore, they aim

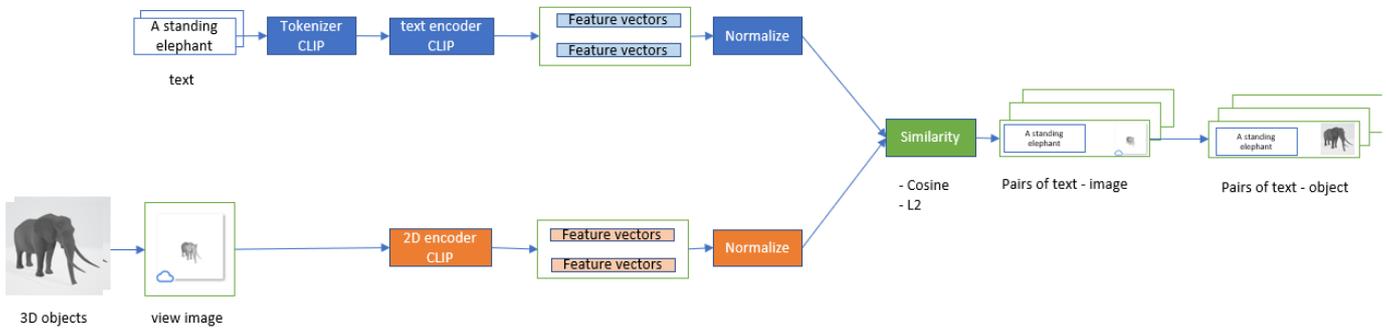


Fig. 7: Proposed framework of Etinifni team.

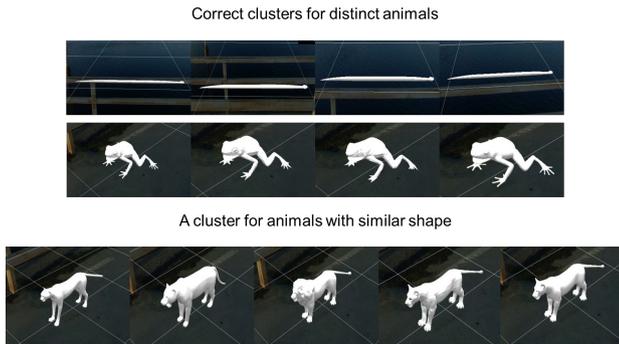


Fig. 8: Clustering results are divided into two groups. The first group is the correct clusters of a specific animal (e.g., the first-row cluster is for a snake, and the second-row cluster is for a frog), while the second one is clusters of animals with similar shapes (the third-row cluster for both lions and leopards).

to group and describe these models by corresponding descriptions to increase the training data.

**Query clustering.** They find that the ANIMAR dataset has the following characteristics: (1) most animals in near families have a similar appearance, (2) an animal can have many 3D models with different densities of point clouds corresponding to the granularity. From these remarks, they cluster 3D models using KMeans [39] for further processing. Since there is nothing specific information about the objects yet, they take a naive approach that each object is represented by statistics on the set of points such as mean, variance, percentile, min, and max.

The team also observed that specific local regions on the animal could aid in distinguishing it from other objects. Therefore, instead of computing statistics globally on the entire point cloud, they divided it into smaller clouds along each dimension and performed local statistics on each cloud. Specifically, they divided the height into five parts, the length into six parts, and the width into two parts. Through experimentation, they found that using 150 clusters yielded satisfactory results. The final clustering result was divided into two groups. The first group clustered distinct animals with their different versions. In contrast, the second group grouped animals with similar shapes (e.g., a group of rhinos and hippos or a group of lions and leopards), as shown in Fig. 8. This step allowed them to manually separate these classes for further data augmentation purposes to increase the training data.

**Descriptive query generation.** They observe that 3D animal models are mostly context-independent, except for a few insignificant cases like an arched angry cat. Based on this observation, they *extract descriptive phrases* that contain the characteristics and species names of the corresponding animals. Notably, this information is present in the subject of the sentence (refer to Fig. 2). By doing so, they can convert animal descriptions into new text queries, which directs the model's attention toward these specific features.

Directly using the initial training set is insufficient to train the model due to three main reasons: (1) the number of text queries for model training is too small, about 350 prompts over 711 models, (2) adding context makes the training process more difficult, (3) the number of given text queries is insufficiently distributed for different versions of an identical animal. Therefore, through the clustering step, they utilize queries from a 3D model to assign them to its other versions. As a result, after this step, they are able to increase the original dataset by nearly three times (1100 prompts).

#### 5.4. Etinifni Team

##### 5.4.1. Text-Image Learning Framework

They develop a text-image learning framework for the text-based 3D animal fine-grained retrieval (see Fig. 7), including an image feature extractor and a text feature extractor.

**Image feature extraction.** For each 3D object, they extract two views from different angles and convert the text-object retrieval task into a text-image retrieval task (i.e., retrieving each 3D object by retrieving its corresponding two images). They also find that using many views extracted from 3D objects (e.g., 12 views as in the work of Su *et al.* [40]) is inefficient in retrieval tasks as various views from an object may cause noise and harm the model.

Blender<sup>3</sup> is used to set up a camera at an appropriate distance, height, and orientation to ensure comprehensive coverage of the object's surface. Additionally, light is positioned at the camera position to provide suitable illumination. They use the horizontal view and oblique angle view of the object (depicted in Fig. 10). CLIP image encoder [30] is used to extract features from view images. They are normalized and then used for the training stage.

<sup>3</sup><https://www.blender.org>

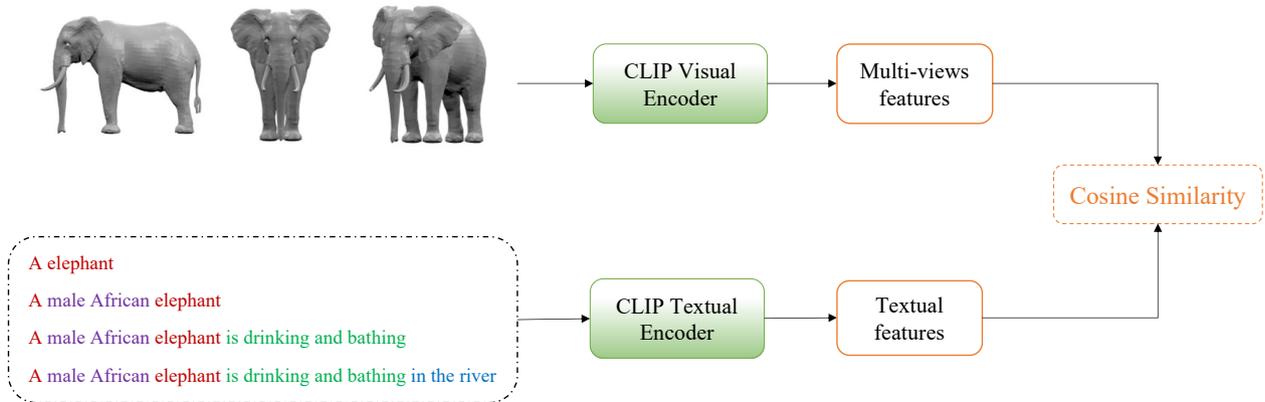


Fig. 9: Proposed framework of THP team.

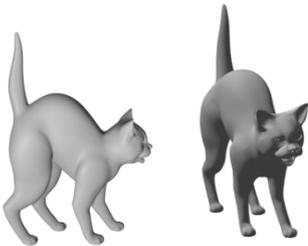


Fig. 10: Two views of a cat are used in the proposed framework of the Etinifni Team.

**Text feature extraction.** They first pre-process text by cleaning and fixing error prompts manually. The prompts are then fed into the text tokenizer and encoder of the CLIP model [30] with backbone ViT-B16 to produce feature vectors. After that, these feature vectors are normalized and then used for the training stage.

**Loss function.** Since each text in the dataset can be paired with multiple view images, inspired by the work of Tran *et al.* [41], they readjust the CLIP model's original loss function as suitable to the newly generated dataset, as follows:

$$S_T = \frac{T \cdot T^r}{\|T\| \|T^r\|}; S_I = \frac{I \cdot I^r}{\|I\| \|I^r\|}; \quad (3)$$

$$\text{Logits} = \frac{T \cdot I^r}{\|T\| \|I^r\|}; \text{Target} = \sigma(c \cdot \frac{S_T + S_I}{2}), \quad (4)$$

where  $T$  is the text embedding matrix, and  $I$  is the image embedding matrix. From  $T$  and  $I$ , they calculate text similarity  $S_T$  and image similarity  $S_I$  using the cosine similarity function. The pairwise similarity between text and image, which are *Logits*, are aimed to match the mean self-similarity (of text and image) *Target* using cross-entropy loss.  $\sigma(x)$  is the softmax function and  $c$  is the logit scale.

**Training phase.** They split the dataset into two subsets, the training and validation datasets, with the ratio 80% and 20%, respectively. They trained the model for 100 epochs with a batch size of 48. They also used the early stopping technique, in which after 10 epochs, the training process is terminated if the best loss on the validation dataset does not update. Corresponding to each best loss, they obtained the weight of the

model. They applied the AdamW optimization algorithm with  $\beta = (0.9, 0.98)$ ,  $\epsilon = 1e-6$ , and a learning rate of  $1e-6$ . They also applied the learning rate decay technique when training.

**Retrieval phase.** After obtaining the similarity score of each text's correspondence to all views retrieved from corresponding 3D objects, they calculate the similarity score of each piece of text's correspondence to all 3D objects by summing the similarity score of each piece of text to two views of each 3D object as follows:

$$\cos(T_i, O_j) = \cos(T_i, I_{j1}) + \cos(T_i, I_{j2}), \quad (5)$$

where  $\cos(T_i, O_j)$  is the similarity score between text  $T_i$  and 3D object  $O_j$  which is calculated by the cosine similarity function.  $\cos(T_i, I_{j1})$  is the similarity score between text  $T_i$  and the first image  $I_{j1}$  of object  $O_j$  which is calculated by cosine similarity function. From the similarity score, they rank them in descending order and extract the corresponding IDs of 3D objects.

## 5.5. THP Team

### 5.5.1. Proposed Solution

Figure 9 depicts their proposed solution, in which the pre-trained CLIP model [30] is used to extract visual and textual features for each query. CLIP visual feature vectors and textual query vectors are calculated and matched using the cosine similarity function. For each view of the 3D object, they sum the six highest similarity scores. Then the object score is calculated as the maximum score of four views; each object score corresponds to four sentences. They take the average of these scores to get the final result.

**2D Projection.** They use four camera setups to take multiple views of 3D objects:

- For the first camera setup, assuming the 3D object is initially solid along the  $z$ -axis, the camera is solid on the  $Oxy$  plane and looks at the center of the object. The camera is moved around the subject to create 12 views from a distance of 30 degrees each time.
- For the second camera setup, the camera is raised to 30 degrees above the  $Oxy$  plane and moved around to create the next 12 views.

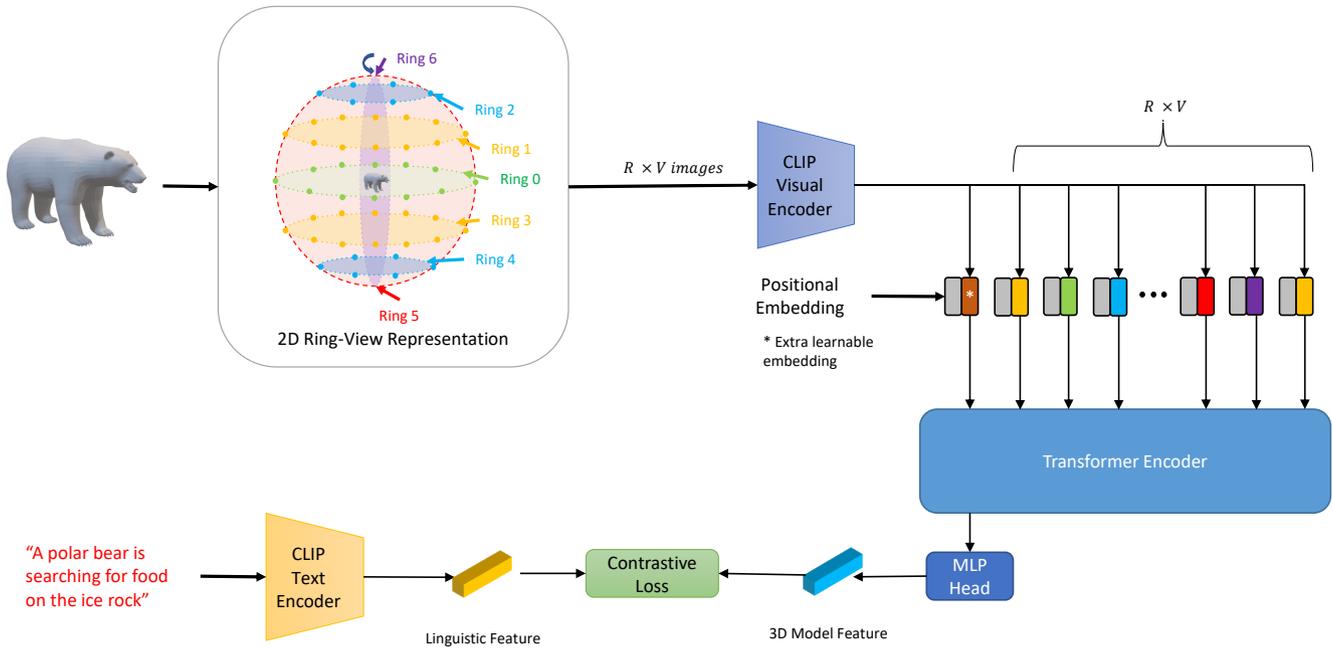


Fig. 11: Proposed framework of Nero team.

- For the third camera setup, the camera is placed on the  $Oyz$  plane and looks at the object's center. The camera moves like the first setup to create the next 12 views.
- Camera for the last setup is raised from the third setup to 30 degrees to create the next 12 views.

A total of 48 views are captured for each object. Despite generating images from other directions, their tests have shown that these 48 views provide sufficient information to observe the object's characteristics. By using Max Pooling to select the final features, additional views do not reduce the model's accuracy.

### 5.5.2. Data Pre-processing

Images are cropped and resized to 224x224 with padding of 5 pixels. For each text query, they split the sentence into small components, such as article, adjective, noun, verb, and object, and then recombine them to make different sentences (see Fig. 9). This helps the model increase the ability to recognize detailed descriptions in sentences. Nouns act as global features and detailed features are shown through adjectives and verbs. Contextual information seems to have little impact because the image has no background.

## 5.6. Nero Team

### 5.6.1. Proposed Network

Figure 11 shows the proposed network, containing four significant components: 2D ring-view representation, CLIP visual encoder, CLIP text encoder, and ring-view transformer encoder.

**Ring-view encoder for 3D models.** To utilize the given 3D object, they extract 2D snapshots from cameras orbiting around

it. They first determine the smallest spherical hull of the object and divide it into a fixed set of  $R$  latitudes, called *rings*. The camera is then positioned at  $V$  evenly spaced positions, facing the center of the object, which they refer to as *views*. This results in  $R \times V$  2D images called ring-view images. For this work, they set  $V = 12$  and  $R$  ranges from 0 to 6, representing the cameras on the equator and the 30/60/90 latitudes from both hemispheres.

**CLIP visual encoder.** To encode the  $R \times V$  2D images generated by the ring-view method, they leverage the CLIP Visual Encoder to obtain  $R \times V$  ring-view features of the 3D animal model views. The CLIP Visual Encoder is a pre-trained deep neural network that encodes natural images into high-dimensional feature vectors based on training on a large corpus of image-text pairs. Using this pre-trained visual encoder, they can effectively capture the relevant visual features of the 3D animal model without requiring additional training or fine-tuning. These encoded visual features are then aggregated for use in conjunction with the textual embeddings to perform fine-grained retrieval tasks. The use of the CLIP Visual Encoder enables us to leverage pre-existing, state-of-the-art visual representations to achieve high accuracy in 3D animal model retrieval.

**CLIP text encoder.** In their method for 3D animal model retrieval, they adopt the CLIP Text Encoder model [30] to generate textual embeddings for the descriptions of the 3D models. These embeddings are then combined with visual embeddings for fine-grained retrieval tasks. The powerful semantic representation capabilities of the CLIP Text Encoder enable us to achieve better performance in distinguishing between visually

Table 2: Leaderboard results of TextANIMAR competition. Best run results on the public test.

| Team     | NN               | P@10             | NDCG             | mAP              | FT               | ST               | FR                |
|----------|------------------|------------------|------------------|------------------|------------------|------------------|-------------------|
| TikTorch | <b>0.520 (1)</b> | 0.220 (2)        | <b>0.651 (1)</b> | <b>0.527 (1)</b> | <b>0.450 (1)</b> | <b>0.570 (1)</b> | 0.0110 (2)        |
| Etinifni | 0.400 (2)        | <b>0.236 (1)</b> | 0.628 (2)        | 0.482 (2)        | 0.370 (2)        | 0.500 (2)        | <b>0.0109 (1)</b> |
| THP      | 0.280 (3)        | 0.192 (3)        | 0.541 (3)        | 0.380 (3)        | 0.300 (3)        | 0.430 (3)        | 0.0114 (3)        |
| Nero     | 0.080 (4)        | 0.084 (4)        | 0.383 (4)        | 0.168 (4)        | 0.140 (4)        | 0.180 (4)        | 0.0130 (4)        |
| Polars   | 0.040 (5)        | 0.032 (5)        | 0.255 (5)        | 0.070 (5)        | 0.000 (5)        | 0.010 (5)        | 0.0141 (5)        |

Table 3: Leaderboard results of TextANIMAR competition. Best run results on the private test.

| Team     | NN               | P@10             | NDCG             | mAP              | FT               | ST               | FR                |
|----------|------------------|------------------|------------------|------------------|------------------|------------------|-------------------|
| TikTorch | <b>0.460 (1)</b> | <b>0.238 (1)</b> | <b>0.647 (1)</b> | <b>0.525 (1)</b> | <b>0.440 (1)</b> | <b>0.585 (1)</b> | <b>0.0108 (1)</b> |
| Etinifni | 0.360 (2)        | 0.200 (2)        | 0.612 (2)        | 0.460 (2)        | 0.300 (3)        | 0.425 (3)        | 0.0114 (2)        |
| THP      | 0.280 (3)        | 0.182 (3)        | 0.549 (3)        | 0.386 (3)        | 0.305 (2)        | 0.430 (2)        | 0.0116 (3)        |
| Nero     | 0.100 (4)        | 0.098 (4)        | 0.398 (4)        | 0.183 (4)        | 0.145 (4)        | 0.210 (4)        | 0.0128 (4)        |
| Polars   | 0.040 (5)        | 0.018 (5)        | 0.252 (5)        | 0.060 (5)        | 0.015 (5)        | 0.030 (5)        | 0.0139 (5)        |

similar but semantically distinct classes of 3D animal models without requiring additional training.

**Vision transformer for ring-view features.** To capture the global information of 3D animal models, they need to gather the ring-view visual features effectively. Each ring-view feature contains the 3D model at a different angle, so they leverage the robust Transformer architecture, which is highly effective for a wide range of natural language processing and computer vision tasks. Specifically, they inherit from the Vision Transformer and consider the Ring-View extraction a patch embedding, where each ring-view is treated as a separate patch. First, the position embeddings are added to the ring-view embeddings to retain positional information. They then use the Transformer encoder to process these patches and generate a final pooled embedding for the 3D animal model. The Transformer encoder allows the model to capture long-range dependencies between the ring views and extract high-level features relevant for fine-grained retrieval.

**Loss function.** During the training process, they employ a variant of contrastive loss called InfoNCE [33]. They compute the InfoNCE loss function not only for the prompt and its corresponding 3D animal model but also for the prompt and other prompts, as well as for pairs of 3D animal models. For each training data batch, they randomly sample a set of prompts along with their corresponding 3D animal models. Subsequently, they compute the InfoNCE loss for the following pairs: prompt and corresponding 3D animal model, prompt and randomly sampled prompt, 3D animal model and randomly sampled 3D animal model. By considering all of these pairs, they encourage the model to learn meaningful representations that capture both the similarity and dissimilarity relationships between prompts and 3D animal models, as well as among different 3D animal models. This approach yields improved retrieval performance and more robust representations of the 3D animal models.

### 5.6.2. Retrieval Phase

To retrieve 3D animal models using text descriptions, they first encode the textual descriptions using the CLIP Text En-

coder, resulting in a high-dimensional textual embedding. They then calculate the similarity scores between the textual embedding of the query and each of the final features of available 3D models using a similarity metric such as cosine similarity. The 3D models are then sorted by their similarity scores in descending order, and all the models are returned in this sorted order. This approach allows us to retrieve all relevant 3D animal models based on natural language descriptions, ranked by their similarity to the query. By leveraging the powerful semantic representation capabilities of the CLIP Model, they can achieve high accuracy in the text-based retrieval of 3D animal models.

### 5.6.3. Training Details

They used the PyTorch framework to train the model. Specifically, they used the AdamW optimizer [38] with a learning rate of 0.0001 to optimize the model parameters. During training, they randomly sampled a set of prompts and their corresponding 3D animal models for each batch of training data. They trained the model for 100 epochs, using a batch size of 16, with the learning rate scheduled to decrease by a factor of 0.1 at epochs 50 and 75.

## 6. Results and Discussions

The TextANIMAR track evaluates submissions on two subsets: the public and private tests. The private test comprises 50 text queries, leading to 188 query-model pairs. Besides, half of the private tests (25 text queries) are randomly selected and assigned to the public test subset to ensure fairness and prevent cheating. The leaderboard for the private test is unveiled only after the challenge’s conclusion.

The leaderboard results for the public and private tests are presented in Tables 2 and 3, respectively. It is important to note that only the top-performing runs submitted by each team are displayed. However, for a fair comparison, we focus on analyzing the results from the private test, which assessed all submitted text queries.

Table 3 illustrates the performance of the submitted methods, with the TikTorch team consistently emerging as the top-performing approach. They achieved a significant lead over

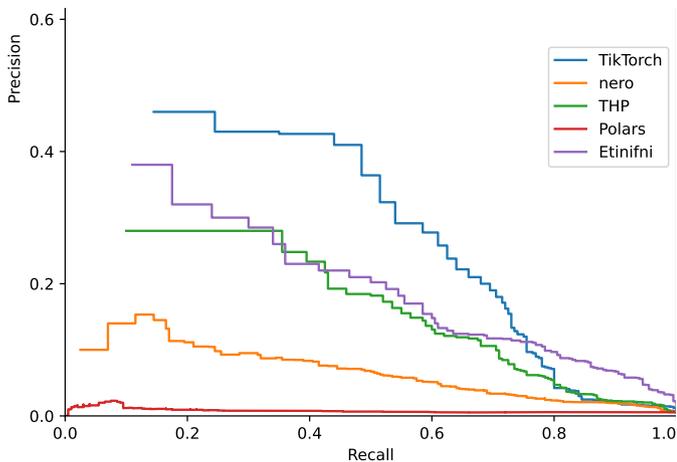


Fig. 12: The visualization of precision-recall curves of submissions on the private test of teams. It can be seen that the TikTorch team achieves the best average performance with the highest area under the curve, while Etinifni obtains the highest precision at a high recall (recall  $\geq 0.8$ ) among the five teams.

other teams in all performance metrics, including the public test results (*cf.* Table 2). Following closely, Etinifni secured the second position, with THP closely behind. Both of these teams utilized CLIP encoders for image and text representation. CLIP has shown promising performance in open-world tasks for 2D images, and its transferability to 3D point clouds is evident in their results. It is interesting that these two teams swap positions on FT and ST metrics, although the results are approximate. In the public test, TikTorch and Etinifni also obtained the top two rankings, with Etinifni surpassing TikTorch in terms of  $P@10$  and FR. The remaining three teams, THP, Nero, and Polars, consistently performed in private and public test sets. Hence, these results illustrate the challenges of our ANIMAR dataset when participants did not perform excellently (the best NN result is less than 0.5, and  $P@10$  results do not surpass 0.25). It also suggests that there is still room for improvement in addressing this research problem.

Figure 12 displays the precision-recall curves of the submissions from five teams, namely TikTorch, Etinifni, THP, Nero, and Polars, on the private test. The graph clearly shows that TikTorch achieved the highest average performance, as indicated by the largest area under the curve. When examining the precision at a low recall threshold (recall  $\leq 0.4$ ), the three teams, TikTorch, Etinifni, and THP, demonstrate relatively acceptable precision levels. However, at a high recall threshold (recall  $\geq 0.8$ ), Etinifni stands out as the most effective team among the five, exhibiting the highest precision.

To sum up, the view-based learning technique emerged as a successful approach for achieving high performance. It can be explained that 3D objects in the ANIMAR dataset have high-density point clouds leading to difficulty for feature extraction models [28, 32]. It is worth noting that these models generally randomly sample pointclouds (*e.g.*, 1024). In contrast, the utilization of view images captured by moving the trajectory camera, as shown in Fig. 3, facilitates feature learning by leveraging the semantic information and representation of 3D objects. This further confirms the efficacy of the view-based learning approach in 3D object retrieval.

## 7. Conclusion

This paper introduces a novel track for text-based retrieval of fine-grained 3D animal models along with a newly constructed ANIMAR dataset to complement existing content-based 3D object retrieval tasks. Our SHREC 2023 challenge track is designed to simulate real-life scenarios and has the potential to become a significant research direction in the field of 3D object retrieval. Despite being more challenging than previous iterations, five groups successfully participated in the track, submitting 114 runs of their proposed methods. The evaluated results of this track were satisfactory, but they also revealed the difficulties of the task at hand.

In the future, we will expand the dataset by collecting a more diverse set of 3D animal models that cover a more comprehensive range of species, postures, and environmental contexts. This expansion will enhance the generalization capability of potential solutions and improve performance on unseen 3D animal models. We also intend to generate synthetic data and texture maps to augment the existing 3D animal models with different postures, backgrounds, and patterns, enabling the training of more robust and effective representation models. Another focus of our future research is investigating language models for effective text query analysis. This analysis can lead to improved retrieval performance and enable useful applications for users who are unable to draw sketches. By pursuing these avenues of research, we aim to enhance the state-of-the-art in 3D object retrieval and facilitate more intuitive and user-friendly interactions with these technologies.

## CRedit authorship contribution statement

**Trung-Nghia Le:** Conceptualization, Writing – review & editing, Project administration, Supervision. **Tam V. Nguyen:** Conceptualization, Writing – review & editing. **Minh-Quan Le:** Software, Writing – review & editing. **Trong-Thuan Nguyen:** Software, Writing – review & editing. **Viet-Tham Huynh:** Data curation. **Trong-Le Do:** Software, Investigation. **Khanh-Duy Le:** Visualization. **Mai-Khiem Tran:** Data curation. **Nhat Hoang-Xuan:** Software. **Thang-Long Nguyen-Ho:** Software. **Vinh-Tiep Nguyen:** Conceptualization. **Tuong-Nghiem Diep:** Methodology, Writing – original draft. **Khanh-Duy Ho:** Methodology, Writing – original draft. **Xuan-Hieu Nguyen:** Methodology, Writing – original draft. **Thien-Phuc Tran:** Methodology, Writing – original draft. **Tuan-Anh Yang:** Methodology, Writing – original draft. **Kim-Phat Tran:** Methodology, Writing – original draft. **Nhu-Vinh Hoang:** Methodology, Writing – original draft. **Minh-Quang Nguyen:** Methodology, Writing – original draft. **E-Ro Nguyen:** Methodology, Writing – original draft. **Minh-Khoi Nguyen-Nhat:** Methodology, Writing – original draft. **Tuan-An To:** Methodology, Writing – original draft. **Trung-Truc Huynh-Le:** Methodology, Writing – original draft. **Nham-Tan Nguyen:** Methodology, Writing – original draft. **Hoang-Chau Luong:** Methodology, Writing – original draft. **Truong Hoai Phong:** Methodology, Writing – original draft. **Nhat-Quynh Le-Pham:** Methodology, Writing – original draft. **Huu-Phuc Pham:** Methodology, Writing

– original draft. **Trong-Vu Hoang**: Methodology, Writing – original draft. **Quang-Binh Nguyen**: Methodology, Writing – original draft. **Hai-Dang Nguyen**: Methodology, Writing – original draft. **Akihiro Sugimoto**: Conceptualization. **Minh-Triet Tran**: Conceptualization, Supervision, Funding acquisition, Writing - Review & Editing.

### Data availability

After the challenge concluded, the dataset has been made publicly available for academic purposes.

### Declaration of competing interest

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

### Acknowledgments

This work was funded by the Vingroup Innovation Foundation (VINIF.2019.DA19) and National Science Foundation Grant (NSF#2025234).

### References

- [1] Stotko, P, Krumpfen, S, Hullin, MB, Weinmann, M, Klein, R. Slamcast: Large-scale, real-time 3d reconstruction and streaming for immersive multi-client live telepresence. *IEEE Transactions on Visualization and Computer Graphics* 2019;25(5):2102–2112.
- [2] Liu, X, Kofman, J. Real-time 3d surface-shape measurement using background-modulated modified fourier transform profilometry with geometry-constraint. *Optics and Lasers in Engineering* 2019;115:217–224.
- [3] Wang, J, Mueller, F, Bernard, F, Sorli, S, Sotnychenko, O, Qian, N, et al. Rgb2hands: real-time tracking of 3d hand interactions from monocular rgb video. *ACM Transactions on Graphics (ToG)* 2020;39(6):1–16.
- [4] Guo, C, Jiang, T, Chen, X, Song, J, Hilliges, O. Vid2avatar: 3d avatar reconstruction from videos in the wild via self-supervised scene decomposition. *arXiv preprint arXiv:230211566* 2023;.
- [5] Koca, BA, Çubukçu, B, Yüzgeç, U. Augmented reality application for preschool children with unity 3d platform. In: *International Symposium on Multidisciplinary Studies and Innovative Technologies (ISM-SIT)*. 2019, p. 1–4.
- [6] He, X, Zhou, Y, Zhou, Z, Bai, S, Bai, X. Triplet-center loss for multi-view 3d object retrieval. In: *Conference on Computer Vision and Pattern Recognition*. 2018, p. 1945–1954.
- [7] Li, Z, Xu, J, Zhao, Y, Li, W, Nie, W. Mpan: Multi-part attention network for point cloud based 3d shape retrieval. *IEEE Access* 2020;8:157322–157332.
- [8] Kim, H, Yeo, C, Cha, M, Mun, D. A method of generating depth images for view-based shape retrieval of 3d cad models from partial point clouds. *Multimedia Tools and Applications* 2021;80:10859–10880.
- [9] Han, XF, Laga, H, Bennamoun, M. Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2019;43(5):1578–1604.
- [10] Song, D, Ling, Y, Li, T, Zhang, T, Jin, G, Guo, J, et al. Gradual adaption with memory mechanism for image-based 3d model retrieval. *Image and Vision Computing* 2022;123:104482.
- [11] Song, D, Zhang, CM, Zhao, XQ, Wang, T, Nie, WZ, Li, XY, et al. Self-supervised image-based 3d model retrieval. *ACM Transactions on Multimedia Computing, Communications and Applications* 2023;.
- [12] Nie, J, Zhang, T, Li, T, Yu, S, Li, X, Wei, Z. Image-based 3d model retrieval via disentangled feature learning and enhanced semantic alignment. *Information Processing & Management* 2023;60(2):103159.
- [13] Qin, J, Yuan, S, Chen, J, Amor, BB, Fang, Y, Hoang-Xuan, N, et al. Shrec'22 track: Sketch-based 3d shape retrieval in the wild. *Computers & Graphics* 2022;107:104–115.
- [14] Shi, X, Chen, H, Zhao, X. Rebor: A new sketch-based 3d object retrieval framework using retina inspired features. *Multimedia Tools and Applications* 2021;80:23297–23311.
- [15] Yang, H, Tian, Y, Yang, C, Wang, Z, Wang, L, Li, H. Sequential learning for sketch-based 3d model retrieval. *Multimedia Systems* 2022;:1–18.
- [16] Bai, S, Bai, J. Hda2l: Hierarchical domain-augmented adaptive learning for sketch-based 3d shape retrieval. *Knowledge-Based Systems* 2023;:110302.
- [17] Pratikakis, I, Savelonas, M, Arnaoutoglou, F, Ioannakis, G, Koutsoudis, A, Theoharis, T, et al. Shrec'16 track: Partial shape queries for 3d object retrieval. *3DOR* 2016;1(8).
- [18] Sipiran, I, Lazo, P, Lopez, C, Jimenez, M, Bagewadi, N, Bustos, B, et al. Shrec 2021: Retrieval of cultural heritage objects. *Computers & Graphics* 2021;100:1–20.
- [19] Yuan, J, Li, B, Lu, Y, Bai, S, Bai, X, Bui, NM, et al. Shrec'18 track: 2d scene sketch-based 3d scene retrieval. *Eurographics Workshop on 3D Object Retrieval* 2018;18:70.
- [20] Yuan, J, Abdul-Rashid, H, Li, B, Lu, Y, Schreck, T, Bui, NM, et al. Shrec'19 track: Extended 2d scene sketch-based 3d scene retrieval. *Eurographics Workshop on 3D Object Retrieval* 2019;18:70.
- [21] Abdul-Rashid, H, Yuan, J, Li, B, Lu, Y, Bai, S, Bai, X, et al. 2D Image-Based 3D Scene Retrieval. In: *Telea, A, Theoharis, T, Veltkamp, R, editors. Eurographics Workshop on 3D Object Retrieval*. 2018;.
- [22] Abdul-Rashid, H, Yuan, J, Li, B, Lu, Y, Schreck, T, Bui, NM, et al. Shrec'19 track: Extended 2d scene image-based 3d scene retrieval. *Eurographics Workshop on 3D Object Retrieval* 2019;700:70.
- [23] Li, W, Liu, A, Nie, W, Song, D, Li, Y, Wang, W, et al. Shrec 2019-monocular image based 3d model retrieval. In: *Eurographics Workshop 3D Object Retrieval*. 2019, p. 1–8.
- [24] Li, W, Song, D, Liu, A, Nie, W, Zhang, T, Zhao, X, et al. SHREC 2020 Track: Extended Monocular Image Based 3D Model Retrieval. In: *Schreck, T, Theoharis, T, Pratikakis, I, Spagnuolo, M, Veltkamp, RC, editors. Eurographics Workshop on 3D Object Retrieval*. 2020;.
- [25] Feng, Y, Gao, Y, Zhao, X, Guo, Y, Bagewadi, N, Bui, NT, et al. Shrec'22 track: Open-set 3d object retrieval. *Computers & Graphics* 2022;107:231–240.
- [26] Wu\*, Y, Chen\*, Z, Liu, S, Ren, Z, Wang, S. CASA: Category-agnostic skeletal animal reconstruction. In: *Neural Information Processing Systems*. 2022;.
- [27] Douze, M, Tolias, G, Pizzi, E, Papakipos, Z, Chaussonot, L, Radenovic, F, et al. The 2021 image similarity dataset and challenge. *arXiv preprint arXiv:210609672* 2021;.
- [28] Qi, CR, Su, H, Mo, K, Guibas, LJ. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: *Conference on Computer Vision and Pattern Recognition*. 2017, p. 652–660.
- [29] Ma, X, Qin, C, You, H, Ran, H, Fu, Y. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:220207123* 2022;.
- [30] Radford, A, Kim, JW, Hallacy, C, Ramesh, A, Goh, G, Agarwal, S, et al. Learning transferable visual models from natural language supervision. In: *International Conference on Machine Learning*. 2021, p. 8748–8763.
- [31] Devlin, J, Chang, MW, Lee, K, Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:181004805* 2018;.
- [32] Muzahid, A, Wan, W, Sohel, F, Wu, L, Hou, L. Curvenet: Curvature-based multitask learning deep networks for 3d object recognition. *IEEE/CAA Journal of Automatica Sinica* 2020;8(6):1177–1187.
- [33] Oord, Avd, Li, Y, Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:180703748* 2018;.
- [34] Chen, T, Kornblith, S, Norouzi, M, Hinton, G. A simple framework for contrastive learning of visual representations. In: *International Conference on Machine Learning*. 2020, p. 1597–1607.
- [35] Tan, M, Le, Q. Efficientnetv2: Smaller models and faster training. In: *International Conference on Machine Learning*. 2021, p. 10096–10106.
- [36] Vaswani, A, Shazeer, N, Parmar, N, Uszkoreit, J, Jones, L, Gomez,

- AN, et al. Attention is all you need. *Advances in neural information processing systems* 2017;30.
- [37] Hinton, GE, Srivastava, N, Krizhevsky, A, Sutskever, I, Salakhutdinov, RR. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:12070580* 2012;.
- [38] Loshchilov, I, Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:171105101* 2017;.
- [39] Lloyd, S. Least squares quantization in pcm. *IEEE Transactions on Information Theory* 1982;28(2):129–137.
- [40] Su, H, Maji, S, Kalogerakis, E, Learned-Miller, EG. Multi-view convolutional neural networks for 3d shape recognition. In: *ICCV*. 2015;.
- [41] Tran, LD, Alam, N, Graham, Y, Vo, LK, Diep, NT, Nguyen, B, et al. An exploration into the benefits of the clip model for lifelog retrieval. In: *International Conference on Content-based Multimedia Indexing*. 2022, p. 15–22.