TriCoLo: Trimodal Contrastive Loss for Text to Shape Retrieval

Yue Ruan1* Han-Hung Lee1* Yiming Zhang1 Ke Zhang1 Angel X. Chang1,2
1Simon Fraser University 2Alberta Machine Intelligence Institute (Amii)
{yuer, hla300, yza440, ke_zhang4, angelx}@sfu.ca
https://3dlg-hcvc.github.io/tricolo/

Figure 1. We introduce TriCoLo, a trimodal contrastive loss for text to 3D shape retrieval. We take objects represented by 3D colored voxels, text descriptions, and multi-view images and jointly use these three modalities to train a trimodal embedding space. This trimodal embedding allows us to perform fine-grained text to shape retrieval.

Abstract

Text-to-shape retrieval is an increasingly relevant problem with the growth of 3D shape data. Recent work on contrastive losses for learning joint embeddings over multimodal data [45] has been successful at tasks such as retrieval and classification. Thus far, work on joint representation learning for 3D shapes and text has focused on improving embeddings through modeling of complex attention between representations [53], or multi-task learning [25]. We propose a trimodal learning scheme over text, multi-view images and 3D shape voxels, and show that with large batch contrastive learning we achieve good performance on text-to-shape retrieval without complex attention mechanisms or losses. Our experiments serve as a foundation for follow-up work on building trimodal embeddings for text-image-shape.

1. Introduction

There has been a dramatic increase in the availability of 3D content in recent years. Improved scanning hardware and reconstruction algorithms are democratizing 3D content creation. The growth in virtual and augmented reality applications has also driven demand for more synthetic (i.e. human-designed) 3D content. It is no wonder that operating systems now natively support viewing and editing 3D content (e.g., iOS/MacOS and Windows). In addition to curated 3D object datasets for research [5, 14, 20, 46, 60], large repositories of 3D shapes provide both synthetic [49, 55, 56] and scanned objects [16, 43].

As 3D assets become more pervasive, we need techniques that allow users to easily and rapidly search through large 3D collections. In recent years, text to image search has seen renewed interest due to improved architectures [9, 34, 37, 45] and objectives [17, 32, 45, 63] for joint representation learning. In contrast, there has been little research on text-driven 3D content search.

Early work by Min et al. [39] compared the text query with text associated with the shape (essentially text-text retrieval). Chen et al. [7] were the first to jointly embed text and 3D shapes for text-to-shape retrieval. They learned the embedding space using triplet loss combined with learning by association [24]. Leveraging the ‘chairs and tables’ dataset [7], followup work investigated improved methods for text-to-shape retrieval [25, 53].

So far, prior work on text-to-shape retrieval has not provided a systematic investigation of: 1) whether 3D information is necessary for text-to-shape retrieval (or whether
single view images to represent a shape are sufficient; 2) whether there are benefits to incorporating information across three modalities; and 3) what contrastive learning setup and loss should be used for constructing joint text-shape embeddings. In our work, we present a systematic study of what is important for improved text-to-shape retrieval. We conduct experiments to examine the effect of input representation (single-view vs multi-view vs 3D voxels), loss function, batch size, and resolution. We show that recent contrastive learning algorithms [63] are sufficient to achieve good performance while avoiding more complex mechanisms, such as combining metric learning with learning by association [7] or training using part-based segmentation of the 3D shapes [53].

In addition, we propose a joint embedding that leverages the multiple modalities offered by 3D data. Specifically, we learn a joint embedding in a trimal setting: voxel, multi-view images and text. Prior work on text-to-shape retrieval either learns a joint representation with voxels and text, or multi-view images and text, both of which are bimodal settings. We use all three modalities to learn the joint embedding space in an end-to-end fashion and show that trimal works better than bimal embedding for text-to-shape retrieval. In summary, our contributions are:

- We introduce a trimal training scheme with contrastive loss that jointly embeds multi-view images, voxels, and language. We show the trimal embedding is effective for text to 3D shape retrieval.
- We release ShapeNet c13, a dataset of paired shapes and captions for 13 object categories from ShapeNet [5].
- We present extensive experiments and analysis to provide guidelines on effective settings for applying contrastive loss for text-to-shape retrieval.
- We establish a high-performing baseline for text-to-shape retrieval. Our simple but effective approach outperforms more complex techniques from prior work. Since we introduced TriCoLo in 2022, several follow-up works [53, 57] have used it as a comparison baseline.

2. Related work

There has been growing interest in connecting language to 3D representations for several tasks: identifying 3D objects in scenes [2, 6, 28, 47, 62, 64], describing 3D objects [10, 26], using 3D augmentation in caption-driven image retrieval [59], generating [7] and disambiguating [1, 54] 3D shapes using natural language.

3D shape retrieval. Min et al. [39] was one of the first to address text to 3D shape retrieval by comparing the text query with textual information associated with the shape. Their approach was based purely on text, and relied on each shape having an associated description. Chen et al. [7] was the first work to create a joint embedding of text and 3D shapes and use that for text-to-shape retrieval. The joint embedding was constructed using a CNN encoder on voxels and GRU encoders on text, with a combined triplet loss [51] and learning by association [24] to align the embedded representations. To improve retrieval, Han et al. [25] used a GRU to encode image features from multiple views to represent the shape, and use reconstruction losses (both intra and inter modalities) in addition to triplet loss and classification loss to train the joint embedding. In contrast, we use multi-view and voxel representation for the shape and do not rely on reconstruction losses. Tang et al. [53] incorporated part-level information, and used point cloud representations for the shapes. In their work, semantic part data was used to compute attention with words to model 3D part relationship with the descriptions. However, obtaining semantic part information can be difficult. Following our work, Wang et al. [57] shows improved performance for text-to-shape retrieval by better selection of positive and negative pairs for contrastive learning, even with just bimal embeddings of text and multi-image views.

3D object disambiguation through language. The task of object disambiguation through language (also known as a reference game) is related to our text-to-shape retrieval. The main difference between the two tasks is a matter of scale. In shape retrieval, we retrieve all objects that match a textual query from a large set of candidate objects. In contrast, in 3D object disambiguation, there is a small set of objects (typically just two or three) from which we select the one that best matches the description. Reference games involving images and language have a long history [13, 18, 21, 30, 40], but there is significantly less work that takes advantage of the 3D nature of objects. Achlioptas et al. [1] used a speaker-listener model for selecting the correct object based on the text description from among three objects. They showed that combining 3D features (from point clouds) with 2D features (from images) is better than just using 3D or 2D features. More recently, Thomason et al. [54] showed that using multi-view images can improve the disambiguation power of a model. Unlike this line of prior work, we focus on text-to-3D shape retrieval and examine the benefit of combining multi-view images and colored 3D voxel representations. Note that the text-to-shape retrieval problem has different characteristics and challenges as it requires selecting from a large number of instances at inference time (vs disambiguating two or three).

Joint embedding. Joint embedding spaces for text and images [17, 19, 32, 45, 58, 63] have enabled retrieval and generation between text and 2D images. Most joint embedding approaches use contrastive learning, focusing on one modality such as images [8, 23, 48, 50], or two modalities [45]. Recently, an increasing number of works explore combinations of more modalities [3, 4, 36, 38], with prior work (typically in combining vision, audio, and lan-
guage) showing that multiple modalities can improve performance [3, 4, 38]. Liu et al. [36] introduce a data augmentation technique where modalities are disturbed to generate negative samples. These lines of prior work are orthogonal to our work as we investigate the use of trimodal contrastive loss on a joint embedding with 3D shape, language, and multiview images for text-to-shape retrieval. Since our work was introduced in 2022, other works started to investigate trimodal embeddings of text-image-shape [35, 61, 65], showing it is useful to train 3D encoders to align 3D embeddings against frozen CLIP text and vision embeddings. ULIP [61] and OpenShape [35] demonstrated that such aligned point cloud embeddings are useful by quantitatively evaluating on classification, and Zhao et al. [65] showed that using an aligned space results in more faithful text-to-shape generation. These works did not focus on evaluating how well the aligned space works for more fine-grained text-to-shape retrieval or comparing different encoders.

3. Problem statement

We tackle the problem of 3D shape retrieval given an input query sentence \( x_t \). We use the Text2Shape [7] dataset which contains tables and chairs from ShapeNet [5] paired with several text descriptions for each object. The text descriptions provide fine-grained information about the appearance of the objects such as color, texture, shape, and whether the object has a certain part (e.g. armrest or a circular base). Accurate retrieval requires learning a good similarity measure between text description and 3D shape. To this end, we learn a shared latent space to facilitate the process of text-shape alignment.

4. Approach

Inspired by recent developments in multimodal contrastive learning [3, 4, 36, 38], we use 3D voxels, multi-view images and language to learn a shared embedding using contrastive learning. Fig. 2 shows how we encode different modalities with per-modality architectures. Embeddings of the same object are pulled closer, while those of different objects are pushed apart using contrastive loss.

**Encoder models.** We represent the input 3D voxels, text description and multi-view images as \( x_v, x_t \) and \( x_i \), respectively. For each modality \( m \in \{v, i, t\} \), an encoder \( f_m \) takes the input \( x_m \) and outputs an encoding \( u_m \in \mathbb{R}^d \). The text encoder \( f_t \) is a Bi-directional Gate Recurrent Unit (Bi-GRU) [12] which takes a text description \( x_t \) in \( \mathbb{R}^{L \times e_t} \) and outputs the embedding \( u_t \in \mathbb{R}^d \), where \( L \) and \( e_t \) are the sentence and word embedding lengths respectively. For voxels, a 3D CNN model \( f_v \) takes a 3D input of \( x_v \in \mathbb{R}^{r_v \times r_v \times 4} \) and outputs \( u_v \in \mathbb{R}^d \) where \( r_v \) is the voxel resolution. Finally, the image encoder takes \( M \) views of the object \( x_i \in \mathbb{R}^{M \times r_i \times r_i \times 3} \) through an MVCNN [52] architecture with pretrained ResNet18 [27] backbone \( f_i \) to obtain the representation \( u_i \in \mathbb{R}^d \) where \( r_i \) is the image resolution.

**Loss function.** We adopt the bimodal loss from ConVIRT [63]. Specifically for two modalities \( m_1, m_2 \in \{v, i, t\} \) so that \( m_1 \neq m_2 \) and a batch size of \( N \) we construct \( N \) positive pairs \((u_{m_1, j}, u_{m_2, j})\) for embeddings of the same object and \( N^2 - N \) negative pairs \((u_{m_1, j}, u_{m_2, k})\) for different objects. We then apply the symmetric NT-Xent contrastive loss from ConVIRT [63] and popularized by CLIP [45]:

\[
L_j^{\rightarrow i} = -\log \frac{\exp(\langle u_v, u_t \rangle / \tau)}{\sum_{k=1}^{N} \exp(\langle u_v, u_{t_k} \rangle / \tau)},
\]

\[
L_j^{\rightarrow v} = -\log \frac{\exp(\langle u_i, u_t \rangle / \tau)}{\sum_{k=1}^{N} \exp(\langle u_i, u_{v_k} \rangle / \tau)}
\]

where \( \tau \in \mathbb{R}^+ \) is a temperature parameter that controls the concentration of the distribution and smoothness of softmax, and \( \langle \cdot, \cdot \rangle \) is the cosine similarity. Finally we calculate a weighted sum of \( L_j^{\rightarrow i} \) and \( L_j^{\rightarrow v} \) and average over the mini-batch: \( L(v, t) = \frac{1}{N} \sum_{j=1}^{N} (\alpha L_j^{\rightarrow v} + (1 - \alpha) L_j^{\rightarrow i}) \),

where \( \alpha \in [0, 1] \) | **Trimodal loss:** To extend the loss to three modalities we simply calculate the contrastive loss over all pair possibilities for the text, voxel and image representations. This gives the final loss: \( L_{tri} = L(v, i) + L(v, t) + L(i, t) \).

**Retrieval.** For the retrieval task, we are given an input text description and we have to return the matching object. To do this, we leverage the shared embedding space we have built between three modalities. We consider three strategies for matching the text and shape by calculating the cosine similarity between: 1) text and voxel embeddings, 2) text and image embeddings, and 3) text and sum of image and voxel embeddings.
5. Experiments

In the main paper, we present experiments on text-to-shape retrieval on the ‘chair and tables’ dataset from Text2Shape [7]. The ‘chairs and tables’ dataset consists of solid colored voxels of 6521 chairs and 8378 tables from ShapeNet [5], and text descriptions collected from humans (≈ 5 per shape). We follow the train/val/test split by Chen et al. [7] which ensures that shapes do not occur in the same split. We present additional results on shape-to-text retrieval, and results on the primitives dataset in the supplement. To show our method works beyond ‘chairs and tables’, we conduct text-to-shape retrieval on an extended set of 13 object categories from ShapeNet [5] (see supplement). Results across experiments consistently show that our trimodal model outperforms bimodal models.

5.1. Metrics

We follow prior work on text-to-shape retrieval [7, 25, 53] and use standard metrics of Recall Rate (RR@k) and Normalized Discounted Cumulative Gain (NDCG) [29]. RR@k deems a retrieval successful if the ground truth (GT) appears in the top k candidates (we set k to 1 and 5). NDCG compares the ranked retrieval results with optimal ranking. However, since we assume there is only one relevant shape for each query, we do not take full advantage of the NDCG metric. We also evaluate using Mean Reciprocal Rank (MRR), the average of the inverse of the rank of the GT.

We note that there are often multiple shapes that can match the text description. Since the text description can be underspecified, we also measure the similarity of the top k retrieved shapes to the GT shape. Following work in shape retrieval [33], we use a point-wise $F^1_\tau$ with $\tau = 0.1$ to calculate shape similarity. $F^1_\tau$ is the harmonic mean of the fraction of points from retrieved shapes within $\tau$ of a point from GT (point-wise precision), and the fraction of points from GT within $\tau$ of a point from retrieved shapes (point-wise recall). To compute $F^1_\tau$, we sample 10K points uniformly on the mesh surface of GT and retrieved shapes. See the supplement for more details.

5.2. Implementation details

We use a one-layer bi-directional GRU [12] for the text encoder, and a 3D CNN architecture for the voxel encoder. We use the pretokenized and lemmatized text from Chen et al. [7], with a vocabulary consisting of 3587 unique words and 1 pad token. For the Bi-GRU, we use word embedding size of 256, and a hidden state size of 128. Word embeddings are randomly initialized from a standard Normal distribution. For the 3D CNN, we use 5 Conv3D layers of sparse convolutions from the spconv library.1 For multi-view images we use the MVCNN [52] architecture with pre-trained ResNet18 [27] backbone. A fully-connected layer is added to ensure the output dimension for all encoders is 512. Unless otherwise specified, training uses batch size 128, voxel resolution 64$^3$, image resolution 128$^2$ and 6 images for the MVCNN. In preprocessing, we normalize image and voxel values from 0-255 to 0-1. We implement our models using PyTorch [42] and train with the Adam optimizer [31]. We use a learning rate of 0.00035 for batch size 128, and adopt the linear scaling rule [22] to scale the learning rate for other batch sizes. We train for up to 20 epochs until convergence, and select the checkpoint with the best performance on the val set. All models are trained on an A40 GPU with each experiment taking about 1 hour. Our models are memory efficient with most models requiring less than 12GB (see supplement). We render the multiview images with Pyrender2 from 12 camera positions elevated slightly above the object, pointing towards the object, and separated by 30 degrees. For multiview experiments using fewer images, we subsample so images are evenly spaced.

5.3. Models

Baselines. We compare to Text2Shape [7], Y2Seq2Seq [25] and Parts2Words [53]. Text2Shape [7] uses a triplet loss [51] combined with learning by association [24]. Y2Seq2Seq [25] uses a view-based model and a triplet constraint. For Parts2Words, we report results for a global model (no part modeling) and their full model that uses part information. Parts2Words [53] uses point clouds as input instead of voxels. The global model uses PointNet [44] as the feature encoder and a Bi-GRU as the text encoder and aggregates the point and text features. Parts2Words jointly embeds point clouds and text by aligning parts from shapes and words from sentences. Both the global and the part-based models use a semi-hard negative mining triplet ranking loss. We note that these methods use either more complex losses or require additional labelled data compared to our model. In addition to baselines from prior work, we use two random baselines: one computes the expected metric mathematically, and the other uses our architecture with random weights.

Our models. We train variants of our model with just two modalities (Bi) or all three modalities (Tri). For the bimodal models, we only consider text and image (I), or text and voxels (V). During retrieval, we compute the similarity of the text with image (I), or text with voxels (V). In the case of trimodal embedding, we also use a combination of the image and voxel when computing the similarity, with (I+V) denoting that the retrieval was done by calculating similarity with text and the sum of image and voxel representations.

1https://github.com/traveller59/spconv

2https://github.com/mmatl/pyrender
Table 1. Retrieval results on the test set with Tri(I+V). For each description, we use Tri(I+V) to retrieve the top-5 shapes. We show the F1 score (as a percentage) for each retrieved shape and mark the ground-truth shape (indicated by green GT). The expected F1 score for GT is 100. Shapes that are not a perfect match to the description are marked in dark orange (color mismatch), and gold (shape detail mismatch). Results show that our network has good language grounding ability overall. It can retrieve shapes that match L-shaped (row 1), stainless steel frame feet (row 2), circular table (row 3), no leg (row 3), circular base (row 3), greenish top (row 4), wooden (row 4), boxy look (row 5), gray (row 5), armless (row 6) and nine-square back (row 6). Though we assume one ground-truth shape, multiple shapes can match the query description (row 1, 3-5).

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Top1</th>
<th>Top2</th>
<th>Top3</th>
<th>Top4</th>
<th>Top5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>an L-shaped dark brown colored wooden table.</td>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
<td><img src="image3" alt="Image" /></td>
<td><img src="image4" alt="Image" /></td>
<td><img src="image5" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>17.31</td>
<td>6.60</td>
<td>20.34</td>
<td>GT</td>
<td>2.19</td>
</tr>
<tr>
<td>2</td>
<td>a luxurious gray leather modern concept plush chair with stainless steel frame feet</td>
<td><img src="image6" alt="Image" /></td>
<td><img src="image7" alt="Image" /></td>
<td><img src="image8" alt="Image" /></td>
<td><img src="image9" alt="Image" /></td>
<td><img src="image10" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GT</td>
<td>2.89</td>
<td>9.50</td>
<td>1.78</td>
<td>3.42</td>
</tr>
<tr>
<td>3</td>
<td>simple circular table with no leg and only one circular base.</td>
<td><img src="image11" alt="Image" /></td>
<td><img src="image12" alt="Image" /></td>
<td><img src="image13" alt="Image" /></td>
<td><img src="image14" alt="Image" /></td>
<td><img src="image15" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.79</td>
<td>5.77</td>
<td>0.59</td>
<td>GT</td>
<td>6.32</td>
</tr>
<tr>
<td>4</td>
<td>This is greenish top wooden billiards table.</td>
<td><img src="image16" alt="Image" /></td>
<td><img src="image17" alt="Image" /></td>
<td><img src="image18" alt="Image" /></td>
<td><img src="image19" alt="Image" /></td>
<td><img src="image20" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.79</td>
<td>GT</td>
<td>8.04</td>
<td>19.59</td>
<td>3.18</td>
</tr>
<tr>
<td>5</td>
<td>this is a boxy look gray chair. It appears to be made out of granite and is gray with 4 short legs and a high, arched back.</td>
<td><img src="image21" alt="Image" /></td>
<td><img src="image22" alt="Image" /></td>
<td><img src="image23" alt="Image" /></td>
<td><img src="image24" alt="Image" /></td>
<td><img src="image25" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GT</td>
<td>4.64</td>
<td>22.32</td>
<td>12.97</td>
<td>11.42</td>
</tr>
<tr>
<td>6</td>
<td>wooden armless dining room chair with open nine-square back.</td>
<td><img src="image26" alt="Image" /></td>
<td><img src="image27" alt="Image" /></td>
<td><img src="image28" alt="Image" /></td>
<td><img src="image29" alt="Image" /></td>
<td><img src="image30" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GT</td>
<td>19.36</td>
<td>13.31</td>
<td>18.78</td>
<td>12.38</td>
</tr>
</tbody>
</table>

Figure 3. Retrieval results on the test set with Tri(I+V). For each description, we use Tri(I+V) to retrieve the top-5 shapes. We show the F1 score (as a percentage) for each retrieved shape and mark the ground-truth shape (indicated by green GT). The expected F1 score for GT is 100. Shapes that are not a perfect match to the description are marked in dark orange (color mismatch), and gold (shape detail mismatch). Results show that our network has good language grounding ability overall. It can retrieve shapes that match L-shaped (row 1), stainless steel frame feet (row 2), circular table (row 3), no leg (row 3), circular base (row 3), greenish top (row 4), wooden (row 4), boxy look (row 5), gray (row 5), armless (row 6) and nine-square back (row 6). Though we assume one ground-truth shape, multiple shapes can match the query description (row 1, 3-5).

5.4. Results

We present qualitative retrieval examples and quantitative evaluations comparing our method to prior work. We examine the choice of different loss functions and hyperparameters (see supplement for experiments on image and voxel resolution, and experiments with different backbones). We train models with different seeds and report mean and standard error across 3 runs.

Example retrievals Fig. 3 shows successful retrievals of shapes using Tri(I+V), our best-performing model (see supplement for more examples). Our model successfully grounds language describing shape (L-shaped, boxy), color (brown, greenish), and texture (wooden). It can also handle negation (armless). Note that many shapes match the description despite not being the ground-truth shape, indicating that there are indeed many matching shapes for a given description. For example, in row 3 the retrieved shapes all match the description, but the four of the five shapes would be negatives in our training and retrieval metrics.

Comparison with prior work. We report the text-to-shape retrieval results in Tab. 1. The full Parts2Words [53] model assumes prior part segmentation knowledge to compute attention with the word embeddings and trains using the triplet loss with negative sampling. In contrast, we do not leverage any part prior knowledge, or attention mechanisms. For comparison, we include Parts2Words (global) with average pooling, which does not use any part information. Tab. 1 shows that our method performs better on all retrieval metrics, and we can achieve slightly better per-
 performance than even the full Parts2Words model. Note that there are several differences in the prior work compared to our own: the network architectures and specifics of the loss functions, as well as different input representations. Chen et al. [7] used 32\(^3\) colored voxels, while Y2Seq2Seq [25] used multi-view images, and Parts2Words [53] used colored point clouds.

### Bimodal vs Trimodal
We compare the trimodal joint embedding with bimodal ones (see Table 2). The modalities in the parentheses indicate which representation was used to retrieve the 3D shapes with respect to the text embeddings. We see that the trimodal embedding improves retrieval performance across all metrics when retrieving by both images and voxels. We obtain the best result when we sum the image and voxel embeddings, indicating that the information in the voxels is complementary to the multi-view images. Fig. 4 shows a comparison of retrieved shapes for an example description from the validation set. The retrieved shapes using Tri(I+V) conform to the description more closely than Bi(I) or Bi(V) shapes. See supplement for more examples.

### Can pretrained CLIP encoders help?
We investigate whether using pretrained vision-language encoders can help improve performance. Specifically, we experiment with CLIP [45], a popular text-image embedding that was trained using contrastive learning with the NT-Xent loss on a large corpus of 400M image-text pairs. We took the pretrained CLIP with ViT-L/14 [15] backbone, and aggregated the embeddings from 6 multi-view images.

We compared the performance of using CLIP in a zero-shot (ZS) manner (without training any weights) and using the CLIP image and text encoders in our models. For zero-shot retrieval, we use the average of the multi-view image embeddings as our overall image embedding, and match that against the text embedding by taking the dot product. To incorporate CLIP into our models, we project frozen CLIP embeddings using a two-layer MLP (see supplement). We present results for zero-shot CLIP and variations of using the text or image CLIP encoders in Table 3. We find that zero-shot CLIP does not perform well, likely due to the domain gap between the rendered images and the CLIP training data. Nevertheless, it can beat the baseline method from the original Text2Shape [7] without being trained on the dataset. Incorporating CLIP into our models and training the MLP results in higher performance, showing the value of training a task-specific MLP on the Text2Shape dataset. We find that our models, which are trained from scratch, are able to outperform the CLIP variants. We also experiment on the ShapeNet c13 data (see supplement), where the CLIP-based models perform much better. We believe that is likely due to less training data for the other categories.

### NT-Xent vs triplet loss.
To validate the choice of NT-Xent
Table 5. Comparison of number of images on shape retrieval for Bi(I) on the val set. We find that having multiple views is important for improved performance, but increasing the number of images beyond 6 causes a slight decrease in RR@1. We believe that 6 views is sufficient to capture the necessary information, and increasing it further increases the number of parameters and requires more compute.

<table>
<thead>
<tr>
<th># of images</th>
<th>RR@1(↑)</th>
<th>RR@5(↑)</th>
<th>NDCG@5(↑)</th>
<th>MRR(↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.15 ± 0.11</td>
<td>26.34 ± 0.32</td>
<td>17.84 ± 0.20</td>
<td>18.36 ± 0.19</td>
</tr>
<tr>
<td>3</td>
<td>11.19 ± 0.19</td>
<td>30.22 ± 0.25</td>
<td>20.97 ± 0.05</td>
<td>21.25 ± 0.12</td>
</tr>
<tr>
<td>6</td>
<td>11.61 ± 0.20</td>
<td>30.65 ± 0.19</td>
<td>21.36 ± 0.23</td>
<td>21.46 ± 0.25</td>
</tr>
<tr>
<td>12</td>
<td>11.23 ± 0.20</td>
<td>31.13 ± 0.10</td>
<td>21.43 ± 0.09</td>
<td>21.50 ± 0.15</td>
</tr>
</tbody>
</table>

Table 6. Comparison of batch-size on shape retrieval for Tri(I+V) on the val set. We find that increasing the batch size increases the performance. However, the performance decreased for the largest batch size we tried (256). This could be due to overfitting on the limited amount of negatives, or the presence of more noisy negatives in the large batch.

<table>
<thead>
<tr>
<th>batch size</th>
<th>RR@1(↑)</th>
<th>RR@5(↑)</th>
<th>NDCG@5(↑)</th>
<th>MRR(↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>10.62 ± 0.27</td>
<td>30.19 ± 0.61</td>
<td>20.60 ± 0.16</td>
<td>20.86 ± 0.11</td>
</tr>
<tr>
<td>64</td>
<td>11.48 ± 0.34</td>
<td>31.40 ± 0.55</td>
<td>21.67 ± 0.37</td>
<td>21.80 ± 0.32</td>
</tr>
<tr>
<td>128</td>
<td>12.52 ± 0.28</td>
<td>32.67 ± 0.61</td>
<td>22.87 ± 0.46</td>
<td>22.68 ± 0.32</td>
</tr>
<tr>
<td>256</td>
<td>12.43 ± 0.35</td>
<td>32.25 ± 0.58</td>
<td>22.53 ± 0.49</td>
<td>22.62 ± 0.40</td>
</tr>
</tbody>
</table>

5.5. Hyper-parameter analysis

We compare the performance of bimodal models on the validation set with different numbers of input images and batch sizes. We use the bimodal models as they are faster to train and require less memory than the trimodal model.

Do we need multi-view images? For Bi(I), we experiment with number of images ranging from 1 to 12 and find that performance increases as we increase the number of images to 6, after which there are diminishing returns and even a small drop in performance (see Tab. 5). The results indicate that multi-view images provide a benefit over a single view.

Does larger batch size always help? We also compare batch sizes of 32, 64, 128 for Tri(I+V) and find that performance increases with increasing batch size from 32 to 128 (see Tab. 6). This is consistent with findings from prior work on contrastive learning [8, 41]. However, the performance drops when the batch size increases to 256 for Bi(I). For Bi(V), increasing the batch size to 256 makes little difference. We hypothesize this is due to more false negatives in the batch since the text description may apply to multiple shapes. Another reason may be that since our dataset size is small compared to image datasets used in prior work [8, 45], having a big batch size might overfit our model. We also note that variance is quite high between runs, which we again attribute to false negatives in the batch and randomness introduced when sampling batches. However, more investigation is warranted.

Impact of other parameters. We also conduct additional experiments (see supplement) examining the effect of different resolutions for image and voxels (higher resolution is better), sparse convolutions (similar performance with less memory, but more time to train), image backbone (smaller backbone works better), zero-shot performance of CLIP (works but not as good as model trained on the data) as well as CLIP embeddings projected using a trained MLP (similar performance as Bi(I) on ‘chairs and tables’ but slightly better for ShapeNet c13).

5.6. Error analysis

Manual analysis. We conduct a manual analysis of the top 5 results returned for 50 text queries from the validation set for Bi(I), Bi(V), and Tri(I+V). We count the number of query results (shapes) that match the description exactly, and categorize the error into color mismatch, large shape mismatch, shape detail mismatch, and missing part (see supplement for examples and Tab. 7 for analysis summary).

As expected from the quantitative results, Tri(I+V) has the most shapes that match the text. With the limited number of queries we examined, all models have similar performance on color and missing parts. The Bi(I) model had difficulty getting small shape details correct, and Tri(I+V) obtained the best performance on matching the overall shape.

Failure cases. We conduct a manual analysis of the top 5 results returned for 50 text queries from the validation set for Bi(I), Bi(V), and Tri(I+V). We count the number of query results (shapes) that match the description exactly, and categorize the error into color mismatch, large shape mismatch, shape detail mismatch, and missing part (see supplement for examples and Tab. 7 for analysis summary).

As expected from the quantitative results, Tri(I+V) has the most shapes that match the text. With the limited number of queries we examined, all models have similar performance on color and missing parts. The Bi(I) model had difficulty getting small shape details correct, and Tri(I+V) obtained the best performance on matching the overall shape.
<table>
<thead>
<tr>
<th>match</th>
<th>color mismatch</th>
<th>big shape error</th>
<th>small shape error</th>
<th>missing part</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bi(I)</td>
<td>65</td>
<td>22</td>
<td>85</td>
<td>5</td>
</tr>
<tr>
<td>Bi(V)</td>
<td>67</td>
<td>26</td>
<td>76</td>
<td>5</td>
</tr>
<tr>
<td>Tri(I+V)</td>
<td>64</td>
<td>17</td>
<td>74</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 7. Manual analysis of the top 5 results returned for 50 text queries. We group the results into whether they perfectly match the description, or whether there is a mismatch in color or shape. We confirm that Tri(I+V) has the best overall performance with the most perfect matches and the least number of shape mismatches.

shapes match the input description (28/100 samples that we manually inspected belonged to this error case). This illustrates that the evaluation data and protocol should be improved beyond what was established in Text2Shape [7]. Despite this, our manual analysis indicates the metrics do a good job of ranking the models. The second row shows the challenge of retrieving shapes with fine detail. While our model can retrieve *taupe* colored armchairs, the retrieved objects did not have *wide armrest*. Our model is good at the overall shape and color, but can miss fine details and infrequent terms such as *foldable*. More data and modeling part-to-text correspondence (as in Parts2Words [53]) can help to reduce these types of failures.

5.7. Limitations

We investigated a trimodal loss for text-to-shape retrieval and found that with careful tuning we outperformed the SoTA as of early 2022, when this work was initially performed. We restricted our study to voxel-based 3D representations which often do not capture geometric details and fine-grained surface textures. It would be interesting to consider other modalities such as point clouds, depth images, and textured 3D polygonal meshes which may alleviate these limitations. One big challenge of incorporating additional modalities is the memory cost. In addition, we focused on a specific type of contrastive loss. Other contrastive losses, data augmentation, as well as other loss terms such as captioning loss and reconstruction loss are promising directions for further improvement. Our dataset is limited in the style of the text and the coverage of shapes. The evaluation also assumes that there is only one correct shape but as we have noted, multiple shapes can match a description. Thus, a significant challenge is to handle false negative pairs in a mini-batch due to the descriptions being ambiguous. These limitations suggest opportunities for future work. We believe our work can serve as a good foundation for follow-up work in text-to-shape retrieval.

6. Conclusion

We carry out a systematic study of contrastive losses for text-to-shape retrieval. We show that using simple contrastive losses can achieve comparable results to text-to-shape retrieval methods relying on extra annotation and complex losses. Our experiments show that incorporating 3D information either via voxels or multi-view images is helpful for the task. We identify important challenges to solve for the development of useful text-to-retrieval models. In addition, we propose a trimodal contrastive loss which further improves performance by considering both 2D and 3D representations. We hope our systematic study will serve as a foundation encouraging more work on text-to-shape retrieval, which is an increasingly important task as there are more and more 3D data repositories.

Acknowledgements. This work is funded by the Canada CIFAR AI Chair program, an NSERC Discovery Grant, and a TUM-IAS Hans Fischer Fellowship (Focus Group Visual Computing). This research was enabled in part by support provided by WestGrid and Compute Canada. We thank Dave Zhenyu Chen for collecting the text descriptions for ShapeNet c13. We also thank the anonymous reviewers for their feedback, and Manolis Savva for proofreading and editing suggestions.
References


[25] Zhizhong Han, Mingyang Shang, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Y2Seq2Seq: Cross-modal representation learning for 3D shape and text by joint reconstruction and prediction of view and word sequences. In Proceedings of the AAAI Conference on Artificial Intelligence, 2019. 1, 2, 4, 6, 7

[26] Zhizhong Han, Chao Chen, Yu-Shen Liu, and Matthias


