TriCoLo: Trimodal Contrastive Loss for Text to Shape Retrieval

by

Yue Ruan

B.Sc., Central China Normal University, 2018

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Spring 2022

Declaration of Committee

Name: **Yue Ruan**

Degree: **Master of Science**

Thesis title: **TriCoLo: Trimodal Contrastive Loss for Text to Shape Retrieval**

Committee: **Chair:** Angelica Lim
Assistant Professor, Computing Science

Angel X. Chang
Supervisor
Assistant Professor, Computing Science

Richard Zhang
Committee Member
Professor, Computing Science

Greg Mori
Examiner
Professor, Computing Science

Abstract

The thesis focuses on applying contrastive loss in learning joint embeddings over multimodal data and proves the effectiveness at a downstream task (retrieval). Previous work on joint representation learning for 3D shapes and text has mostly focused on improving embeddings through modeling of complex attention between representations, or multi-task learning. We show that with large batch contrastive learning we achieve SoTA on text-to-shape retrieval without complex attention mechanisms or losses. Prior work in 3D and text representations has also focused on bimodal representation learning using either voxels or multi-view images with text. We show that a trimodal learning scheme can lead to even higher performance and better representations for all modalities.

Keywords: Vision and language; Contrastive learning

Acknowledgements

First and foremost, I would like to extend my deepest gratitude to my supervisor Dr. Angel X. Chang, who gave me an invaluable opportunity to study and work in this laboratory I aspire to. She gave me excellent guidance and helpful comments when I was working on this project. The completion of my thesis would not have been possible without the support and nurturing of Dr. Chang.

I would like to convey my heartfelt gratitude to my collaborators, Han-Hung Lee and Ke Zhang who discussed with me about this project, helped me review my code and gave me precious advice when I was stuck in difficulties.

In addition, I also want to thank Dave (Zhenyu) Chen who taught me how to write a qualified academic paper, and Sanjay Haresh and Peizhi Yan who spent their time on proofreading Chapter 5 in my thesis.

Special thanks to Hanxiao Jiang, Fenggen Yu, Manyi Li, Zeshi Yang, Zhiqin Chen and all other lab members in GrUVi for their help when I have questions about research. They are outstanding and kind people who motivate me to do better work during these years.

Finally, all my sincere appreciation goes to my parents and my best friend, Xiaotong Yang, for their steadfast love and encouragement that support me through the darkest time during the pandemic.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

There has been a dramatic increase in the availability of 3D content in recent years. Improved scanning hardware and reconstruction algorithms are beginning to democratize 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, 49, 64], large repositories of 3D shapes provide both synthetic [54, 60, 61] and scanned objects [23].

As 3D assets become more and more pervasive, we need to have 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, 39, 41, 48] and objectives [16, 37, 48, 67] for joint representation learning. On the other hand, there has been very little research on text-driven 3D content search.

The little prior work on text-to-shape retrieval has thus far not provided a systematic investigation of: 1) whether 3D information is necessary for text-to-shape retrieval, or whether it is sufficient to leverage existing text-to-image retrieval methods; 2) whether there are benefits to incorporating data with three modalities information; and 3) what kind of loss/contrastive learning setup should be used for constructing joint text-shape embeddings.

Early work by Min et al. [43] compared the text query with text associated with the shape (this is essentially just text-text retrieval). Chen et al. [7] were the first to create a joint embedding of text and 3D shapes for text-to-shape retrieval. Leveraging the ‘chairs and tables’ dataset introduced by Chen et al. [7], followup work investigated improved methods for text-to-shape retrieval [28, 57]. This line of work leveraged a triplet loss for metric learning over two modalities. We show that recent contrastive learning algorithms [67] are sufficient to achieve SoTA performance while avoiding more complex mechanisms, and result in a more flexible representation.

Most contrastive learning algorithms focus on one modality such as images [8, 26, 53, 55], or two modalities [48]. There is far less literature on three or more modalities. 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. Instead, we propose learning in a trimodal setting (with three modalities): voxel, images and text. This does not require extra datasets as the multi-view images can be rendered from 3D objects. We leverage these modalities to learn the joint embedding space for all three modalities in an end-to-end fashion. The resulting retrieval results are better than learning from bimodal settings.

1.1 Thesis contribution

Using our contrastive loss model, we conduct experiments on text-to-shape retrieval to examine the effect of trimodal vs bimodal embeddings, batch size, and input representation (single view vs multi view vs 3D voxels). We show that with careful tuning we can outperform recent methods that rely on part-based segmentation of the 3D shapes. In summary, our main contributions are:

- We introduce a simple trimodal training scheme for text to 3D shape retrieval.
- We present experiments and analysis to provide insights on the effectiveness of using contrastive learning for cross-modal representation learning and downstream tasks.
- We achieve state-of-the-art performance on multiple retrieval metrics, outperforming existing approaches with more complex methods by 2.31% on RR@1 (relative improvement of 29%).

This thesis is a joint work [51] with Han-Hung Lee, Ke Zhang and Dr. Angel X. Chang. My responsibility includes: starting the Text2Shape project, building the training pipeline and bi-modal loss, implementing the scripts for computing the evaluation metrics, and conducting experiments on the Text2Shape and SNARE datasets. In addition, I also helped to collate part of the qualitative examples and wrote the initial draft. My collaborator Han-Hung Lee not only proposed the tri-modal loss, but also conducted experiments on the Text2Shape and ShapeNet C13 datasets and helped to write the conference paper. Ke Zhang collated part of the qualitative examples, provided the T-SNE visualization and conducted the manual analysis. As our mentor and supervisor for the whole project, Dr. Chang prepared important data for us and assisted with paper writing. The descriptions for the ShapeNet C13 dataset was provided by Dave Zhenyu Chen.

1.2 Thesis organization

This thesis is structured as follows: Chapter 2 discusses related work. Chapter 3 defines the text to shape retrieval problem and explains our approach. Chapter 4 describes our text-to-shape retrieval experiments on the Text2Shape dataset [7]. We describe the dataset, experiment settings, metrics as well as provide quantitative and qualitative evaluations, and conduct error analysis. In Chapter 5, we investigate how well our method can work on the reference game task and apply it to the SNARE dataset [58]. Chapter 6 discusses the limitations of our method and concludes the thesis. Chapter 1, most of Chapter 2, 3, 4, 6 are directly reproduced from the TriCoLo paper [51].

Chapter 2

Related Work

There has been growing interest in connecting language to 3D representations for several tasks: identifying 3D objects in scenes [2, 6, 33, 50, 66, 68], describing 3D objects [10, 29], using 3D scene geometry augmentation in caption-driven image retrieval [63], generating [7] and disambiguating [1, 58] 3D shapes using natural language.

3D shape retrieval. Min et al. [43] were one of the first to address the problem of 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, and using a combined [53]triplet loss and learning by association [27] to align the embedded representations. To improve retrieval performance, Han et al. [28] 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 constrast, our work considers multi-view and voxel representation for the shape and does not rely on any reconstruction losses. Tang et al. [57], the current state-of-the-art approach, proposed to incorporate 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, and because attention is used both data modalities are required to compute the final representation which can limit uses for other downstream tasks such as generation.

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 are interested in retrieving all objects that match a textual query from a large set of candidate objects. In contrast, in 3D object disambiguation, there is a smaller set of objects (typically three) from which we want to select the one that best matches the description. Reference games involving images and language have a long history [13, 18, 22, 35, 44], 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 a model 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. [58] showed that using multi-view images can improve the disambiguation power of a model. Unlike this line of prior work, we focus on the problem of text to 3D shape retrieval and examine the benefit of combining multi-view images and colored 3D voxel representations.

Joint embedding. Joint embedding spaces for text and images [16, 19, 37, 48, 62, 67] have enabled retrieval and generation between text and 2D images. Most joint embedding approaches use contrastive learning. With the success of joint embeddings, researchers have also started to explore combining more modalities [3, 4, 40, 42]. Work involving vision, audio, and language shows that having multiple modalities can improve performance [3, 4, 42]. Liu et al. [40] introduce a general 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 creating a joint embedding with 3D shape, language, and multiview images for text to shape retrieval.

Contrastive learning. Wu et al. [65] proposed to learn representation using a instance discrimination task which does not rely on annotated class labels. They used noise contrastive estimation (NCE) and memory bank to work around with the computational challenge brought by a huge number of instance classes. Oord et al. [45] proposed infoNCE loss which becomes a standard loss function in the following contrastive learning work. Their method can learn meaningful representations for not only images domain but also audio, video, texts and in reinforcement learning domains. Tian et al. [59] extended contrastive learning from two image views to multiple views. He et al. [31] changed memory bank in [65] to a dynamic queue and proposed momentum encoder. Their unsupervised method can outperform supervised couterpart on universally used datasets including PASCAL VOC and COCO. Chen et al. [8] investigated the impact of different techniques on contrastive learning including different losses, larger batch size, various data augmentation, use of a projection head and training time. Grill et al. [25] proposed a method which did not use negatives and learned only from positives. It also did not use infoNCE but use MSE loss instead. He et al. [32] summarized prior works and proposed a simple Siamese network that did not need a large batch size, momentum encoder or negatives. They showed that using a critical stop-gradient operation can prevent the network from model collapsing. He et al. [32] also showed the potential ability of masked learning. Zhang et al. [67] showed that the NT-XEnt loss can be used to learn joint embeddings of images and text in the biomedical domain. By training on large amounts of image and caption data, Radford et al. [48] demonstrated that contrastive learning can be very effective at learning good joint multi-modal embeddings that can be used in a zero-shot manner.

Chapter 3

Text to Shape Retrieval

3.1 Problem statement

We tackle the problem of object retrieval given an input query sentence x_t. Specifically we use the Text2Shape [7] dataset which contains tables and chairs from ShapeNet [5] and provides several text descriptions for each object. The text descriptions provide fine-grained information about the appearance of the objects, for example whether a chair has armrests, whether tables have a rectangular or round base, and texture appearance like color. However, it is also worth noting that some sentences may be ambiguous in that there could be multiple objects that satisfy the description. Accurate retrieval requires that we learn 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.

3.2 Approach

Inspired by recent developments in multimodal contrastive learning [3, 4, 40, 42], we leverage 3D voxels and multi-view images with language to learn a shared embedding space using contrastive learning. As illustrated in Fig. 3.1, we encode the different modalities with per-modality architectures. Embeddings for the same object are then pulled closer, while those belonging to different objects are pushed apart using contrastive loss.

3.2.1 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 $m \in (v, i, t)$, we define an encoder f_m that 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 $x_t \in R^{L \times e_t}$ and outputs the embedding $u_t \in R^d$, where L and e_t are the sentence and word embedding lengths respectively. For voxels we use a 3D CNN model f_v that takes a 3D input of $x_v \in \mathbb{R}^{r_v \times r_v \times r_v \times 4}$ 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 $x_i \in R^{M \times r_i \times r_i \times 3}$ through an MVCNN [56]

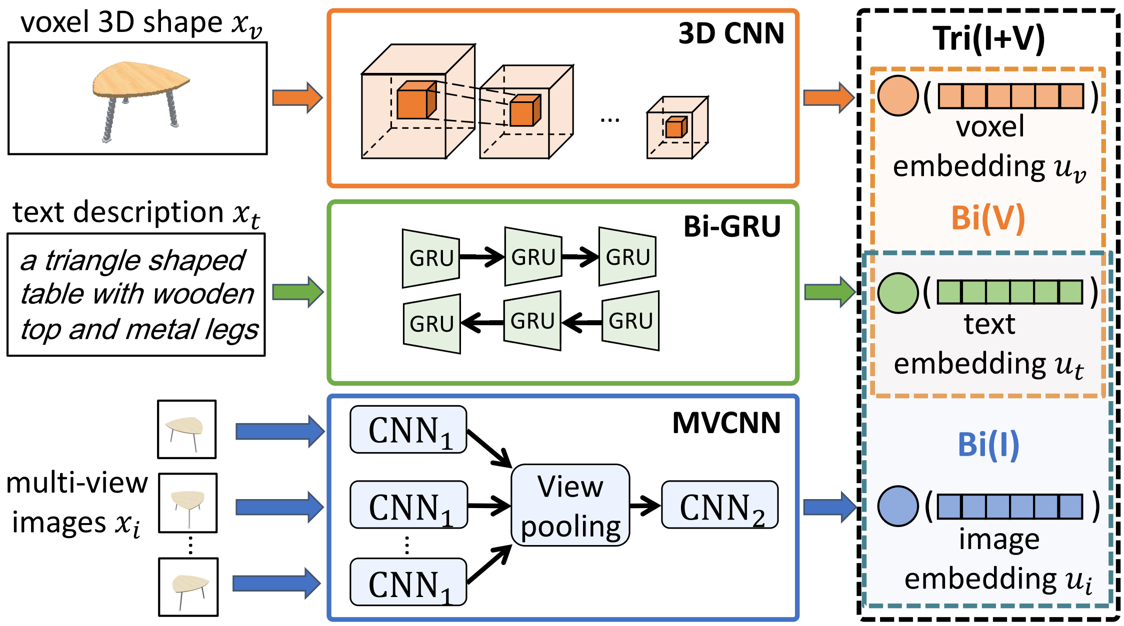


Figure 3.1: Trimodal pipeline. Given the voxel shapes x_v, input text description x_t and rendered images x_i, 3D CNN, Bi-GRU and MVCNN transform them to feature vector u_v, u_t and u_i which are aligned in the hidden space. We then minimize a bidirectional contrastive loss to learn effective shape representations, text representations and image representations that are close to each other if they are from the same object.

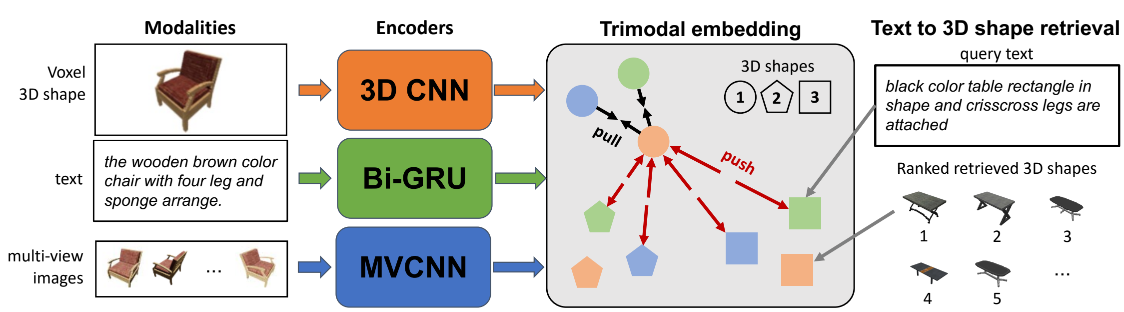


Figure 3.2: We introduce **TriCoLo**, a **tri**modal **co**ntrastive **lo**ss for text to 3D shape retrieval. We take objects represented by 3D colored voxels, text descriptions, and multi-view images and jointly use these three modalities to train a trimodal embedding space. This trimodal embedding allows us to perform fine-grained text to shape retrieval.

architecture with pretrained ResNet18 [30] backbone f_i to obtain the image representation $u_i \in \mathbb{R}^d$ where r_i is the image resolution.

3.2.2 Loss function

We adopt the bimodality loss from ConVIRT [67] for our approach. 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 belonging to the same object and $N^2 - N$ negative pairs $(u_{m_1 j}, u_{m_2 k})_{j \neq k}$ for different objects. The contrastive loss is then applied symmetrically as shown below.

$$l_j^{m_1 \rightarrow m_2} = -\log \frac{\exp(\langle u_{m_{1j}}, u_{m_{2j}} \rangle / \tau)}{\sum_{k=1}^{N} \exp(\langle u_{m_{1j}}, u_{m_{2k}} \rangle / \tau)} \tag{3.1}$$

$$l_j^{m_2 \rightarrow m_1} = -\log \frac{\exp(\langle u_{m_{2j}}, u_{m_{1j}} \rangle / \tau)}{\sum_{k=1}^{N} \exp(\langle u_{m_{2j}}, u_{m_{1k}} \rangle / \tau)} \tag{3.2}$$

where $\tau \in \mathbb{R}^+$ is a temperature parameter that controls the concentration of the distribution and smoothness of softmax, and $\langle , \rangle$ is the cosine similarity. This particular form of contrastive loss is the NT-Xent (normalized temperature-scaled cross entropy loss, as named in Chen et al. [8] and used by other work [45, 65]). Finally we calculate a weighted sum of $l_j^{m_1 \rightarrow m_2}$ and $l_j^{m_2 \rightarrow m_1}$ and average over the minibatch.

$$L(m_1, m_2) = \frac{1}{N} \sum_{j=1}^{N} (\alpha l_j^{m_1 \rightarrow m_2} + (1 - \alpha) l_j^{m_2 \rightarrow m_1}) \tag{3.3}$$

where $\alpha \in [0, 1]$

Trimodal loss To extend the loss to three modalities we simply calculate the ConVIRT [67] loss over all pair possibilities for the text, voxel and image representations. This gives the final loss:

$$L_{\text{tri}} = L(v, i) + L(v, t) + L(i, t) \tag{3.4}$$

3.2.3 Retrieval

For the retrieval task we are given an input text description and we have to return the matching object. To do this we can either calculate similarity between text and voxel representations or text and image representations. Leveraging the fact that the joint embedding space is shared between all three modalities in the trimodal model we can also retrieve objects by calculating similarity between text and the sum of voxel and image representations. Our whole pipeline is shown in Fig. 3.2

Chapter 4

Experiments on the Text2Shape Dataset

4.1 Dataset

We evaluate on the 'chairs and tables' dataset introduced in Text2shape [7]. This dataset contains solid colored voxels from ShapeNet 3D shapes [5] and diverse, fine-grained descriptions from humans. The shapes are 6521 unique chairs and 8378 unique tables. Each 3D shape has an average of 5 captions. We follow the train/val/test split by Chen et al. [7] (see Table 4.1 for statistics).

4.2 Metrics

We follow prior work on text to shape retrieval [7, 28, 57] and use the standard metrics of Recall Rate (RR@k) and Normalized Discounted Cumulative Gain (NDCG) [34] for quantitative comparisons. 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 considers retrieval results with their relevance. We also evaluate using Mean Reciprocal Rank (MRR). MRR is the average of reciprocal ranks which are the multiplicative 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 [38], we use a point-wise $F1^{\tau}$ with $\tau = 0.1, 0.3, 0.5$,

Modality	Category	Train	Validation	Test
Text	Chair	26257	3313	3206
	Table	33520	4122	4246
	Total	59777	7435	7452
Shape	Chair	5221	659	641
	Table	6700	827	851
	Total	11921	1486	1492

Table 4.1: Statistics for the 'chairs and tables' dataset [7] we use.

Layer	Kernel	Stride	Channels	IN	LR
conv1	3	2	32	Y	Y
conv2	3	1	64	Y	Y
max_pool2	3	2	-	-	-
conv3	3	1	128	Y	Y
max_pool3	3	2	-	-	-
conv4	3	1	256	Y	Y
max_pool4	3	2	-	-	-
conv5	3	2	512	Y	Y
adaptive_avg_pool	-	-	-	-	-
fc6	-	-	512	N	N

Table 4.2: Voxel encoder architecture for resolution 64^3

as well as the Chamfer Distance (CD), and (Abstract) Normal Consistency (NC) to calculate the similarity of GT shape and retrieved shapes. $F1^\tau$ is the harmonic mean of the fraction of points from retrieved shapes within τ of a point from GT (point-wise precision), and the fraction of points from GT within τ of a point from retrieved shapes (point-wise recall). We note that these are all point-wise metrics, and we sample 10K points uniformly on the mesh surface of GT and retrieved shapes for computing these metrics. Note that both CD and $F1^\tau$ depend on the absolute scale of meshes. To compute them, we follow Fan et al. [17] who define 1 unit as 1/10 of the largest length of the ground truth's bounding box and rescale the ground truth and retrieved meshes individually.

4.3 Implementation details

We use a one-layer bi-directional GRU [12] for the text encoder, and a 3D CNN architecture for the voxel encoder. The vocabulary contains 3587 unique words and 1 pad token. We use the pretokenized and lemmatized text from Chen et al. [7]. For the Bi-GRU, we use word embedding size of 256, and a hidden state size of 128. Word embeddings are initialized with a standard Normal distribution. For the 3D CNN, we use a 5 Conv3D layers with input resolution 64^3. Tab. 4.2 shows the architectural details. Here IN stands for Instance Normalization and LR stands for Leaky ReLU, each layer of convolution is followed by normalization then activation. The first and last layer have stride 2 and other layers are followed by max pooling operations. An adaptive pooling is placed after the last convolution layer to ensure the spatial size is 2^3 before feeding into the final fully connected layer. Note that for voxel resolution of 32^3 we change the stride of the last convolution layer to 1. We do not conduct any experiments larger than 64^3 as the memory used for 3D CNNs grows cubically.

For multi-view images we use the MVCNN [56] architecture with pretrained ResNet18 [30] 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 the values in images and voxels from

0-255 to 0-1. We implement our models using Pytorch [46] and train with the Adam optimizer [36]. Our learning rate is 0.0004 and experiments with other batch sizes use the linear scaling rule [24]. We train for a maximum of 20 epochs until convergence. We select the checkpoint that gives the minimum loss on the validation set. With smaller models we use RTX 2080 Ti GPUs with 11GB of memory for training, and for larger models with large batch size or three modalities we use V100 GPUs with 32GB of memory. Each experiment takes about 6-8 hours. We rendered the multiview images from the mesh representation of those 3D shapes. More specifically, for the rendering setup we use a Blender-based script[1]. The object is placed at the center (0, 0, 0). The camera is placed at (0, 1, 0.6) with the focal length set to 35mm and the sensor width to 32mm while being pointed towards the center (0, 0, 0). We modify the original script to only render 12 images by rotating the camera 30 degrees per render with render resolution set to 224. The rendering engine used is the Blender EEVEE rasterization engine with the Principled BSDF shader. For multiview experiments using fewer images, we subsample so images are evenly spaced.

4.4 Models

Baselines We compare to Text2shape [7], Y2Seq2Seq [28] and Part2word [57] (end2end, part). Text2shape [7] uses a triplet loss [53] with learning by association [27]. Y2Seq2Seq [28] uses a view-based model and a triplet constraint. Part2word [57] uses point clouds as input instead of voxels. The end-to-end model in Part2word uses PointNet [47] as the global feature encoder and Bi-GRU as the text encoder. The part model in Part2word jointly embeds point clouds and text by aligning parts from shapes and words from sentences. Both the end-to-end and the part-based model use a semi-hard negative mining triplet ranking loss. 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**), or in the case of trimodal embedding, we use a combination of the image and voxel when computing the similarity. We use **I+V** to denote that the retrieval was done by calculating similarity with text and sum of image and voxel representations.

	RR@1	RR@5	NDCG@5
Random (expected)	0.06	0.30	0.20
Random (weights)	0.08	0.32	0.20
Text2shape [7]	0.40	2.37	1.35
Y2Seq2Seq [28]	2.93	9.23	6.05
Part2Word [57] (end2end)	7.13	22.63	14.94
Part2Word [57] (part)	7.94	23.89	16.03
Bi(**I**) (ours)	8.28	24.52	16.52
Bi(**V**) (ours)	8.73	26.10	17.53
Tri(**I+V**) (ours)	**10.25**	**29.07**	**19.85**

Table 4.3: Text to shape retrieval comparison against prior work on the test set. We report the recall rate (RR@1, RR@5) and NDCG@5 as percentages. We train with a batch size of 128, 64^3 voxels, and 6 multi-view images at a resolution of 128^2 each. Our bimodal joint embedding (Bi(**I**), Bi(**V**)) trained using the NT-XEnt loss outperforms prior work, including Part2Word [57] which uses part annotations during training. Our trimodal embedding (Tri(**I+V**)) further improves retrieval performance.

	RR@1(↑)	RR@5(↑)	NDCG@5(↑)	MRR(↑)	CD(↓)	NC(↑)	$F1^{0.1}$(↑)	$F1^{0.3}$(↑)	$F1^{0.5}$(↑)
Bi(**I**)	8.69 ± 0.38	25.29 ± 0.46	17.14 ± 0.42	17.63 ± 0.38	2.01 ± 0.02	0.62 ± 0.002	11.97 ± 0.20	34.37 ± 0.31	48.89 ± 0.36
Tri(**I**)	9.25 ± 0.46	26.24 ± 0.73	17.89 ± 0.59	18.36 ± 0.51	1.91 ± 0.02	0.63 ± 0.002	12.49 ± 0.21	35.56 ± 0.28	50.28 ± 0.29
Bi(**V**)	8.86 ± 0.16	26.41 ± 0.50	17.79 ± 0.30	18.34 ± 0.20	1.96 ± 0.02	0.62 ± 0.002	12.21 ± 0.09	35.01 ± 0.18	49.60 ± 0.24
Tri(**V**)	9.42 ± 0.30	27.90 ± 0.56	18.87 ± 0.40	19.26 ± 0.34	1.89 ± 0.03	0.63 ± 0.002	12.64 ± 0.13	35.75 ± 0.30	50.44 ± 0.37
Tri(**I+V**)	**10.56 ± 0.43**	**29.50 ± 0.56**	**20.20 ± 0.49**	**20.46 ± 0.46**	**1.88 ± 0.02**	**0.63 ± 0.001**	**12.85 ± 0.17**	**36.02 ± 0.32**	**50.70 ± 0.35**

Table 4.4: Comparison of bimodal and trimodal models for text-to-shape retrieval on the validation set. Having a trimodal embedding (Tri(**I**),Tri(**V**)) gives better performance than the bimodal embeddings (Bi(**I**),Bi(**V**)). By summing the image and voxel representations from the trimodal embeddings (Tri(**I+V**)), we further improve retrieval performance.

4.5 Quantitative evaluation

We conduct quantitative evaluations comparing our method to prior work, as well as examining the choice of different loss functions and hyperparameters. We train models with different seeds and report the mean and standard error across 7 runs.

Comparison with prior work We report the text-to-shape retrieval results in Tab. 4.3. The current SoTA Part2word [57] 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. Tab. 4.3 shows that our method performs better on all retrieval metrics. 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 [28] used multi-view images, and Part2Word [57] used colored point clouds. To better understand what factors are important for improved performance, we conduct additional experiments to study the effect of different modalities, loss functions, and hyperparameters.

Bimodal vs Trimodal We compare the trimodal joint embedding with bimodal ones (see Tab. 4.4). 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 if we sum the image and voxel embeddings. This indicates that the information in the voxels is complementary to the multi-view images.

Loss function comparison To validate the choice of NT-Xent as our loss function, we compare the performance of our model using a hinge-based triplet loss [53] instead of NT-Xent. We use semi-hard negative mining with margin of 0.025. Semi-hard negatives have been shown to improve performance for contrastive losses [8]. Specifically Tang et al. [57] showed it worked better than either triplet-loss by itself or hard negatives for retrieval with the Text2Shape dataset. Our results in Tab. 4.5 show that the text-to-shape retrieval performance with triplet loss is significantly lower than that with NT-Xent. Overall, our findings are consistent with prior work [11]. Note that our model outperforms Y2Seq2Seq [28] even with just triplet loss. We find that with NT-Xent loss, our bimodal models surpass the performance of Part2Word [57].

Numbers of input images We compare performance of the bimodal models on the validation set with different numbers of input images. We use the bimodal models as they are faster to train and require less memory than the trimodal model. For Bi(**I**), we conduct experiments 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. 4.6). The results indicate that multi-view images provide a benefit over a single view.

[1] https://github.com/panmari/stanford-shapenet-renderer

	RR@1(↑)	RR@5(↑)	NDCG@5(↑)	MRR(↑)
Bi(**I**)	5.65 ± 0.57	18.87 ± 0.90	12.32 ± 0.77	13.41 ± 0.69
Bi(**V**)	5.66 ± 0.40	19.66 ± 0.56	12.70 ± 0.49	13.79 ± 0.49
Tri(**I+V**)	7.87 ± 0.37	24.15 ± 0.68	16.08 ± 0.55	16.74 ± 0.50

Table 4.5: Text-to-shape retrieval performance on the validation set using triplet loss with semi-hard negative mining. The performance is lower compared to NT-Xent (Tab. 4.4).

	# of images	RR@1(↑)	RR@5(↑)	NDCG@5(↑)	MRR(↑)
Bi(**I**)	1	7.14 ± 0.38	22.18 ± 0.77	14.78 ± 0.59	15.5 ± 0.53
	3	8.02 ± 0.47	24.27 ± 0.74	16.27 ± 0.58	16.82 ± 0.53
	6	**8.69 ± 0.38**	**25.29 ± 0.46**	**17.14 ± 0.42**	**17.63 ± 0.38**
	12	8.54 ± 0.44	25.14 ± 0.57	16.98 ± 0.50	17.51 ± 0.46

Table 4.6: Comparison of number of images on shape retrieval for Bi(**I**) on the validation 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 performance. We believe that that 6 views is likely to be sufficient to capture the necessary information, and increasing it further increases the number of parameters and causes overfitting due to the limited size of the dataset.

Batch size We also compare batch sizes of $32, 64, 128$ for Bi(**I**) and Bi(**V**) and find that performance increases with increasing batch size from 32 to 128 (see Tab. 4.7). This is consistent with findings from prior work on contrastive learning [8, 45]. 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, 48], 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.

Image and voxel resolutions We conduct experiments for different resolutions of images (64^2, 128^2 and 224^2) and voxels (32^3 and 64^3). In Tab. 4.9 we see that the performance increases with higher resolutions. We limit our voxel experiments to 64^3 as the memory required for higher resolutions grows cubically. It is also possible to use sparse convolutions to handle the higher resolution, but we have focused our experiments on using solid voxelizations (the interior of each shape is filled with voxels), for which it is unclear whether sparse convolutions would help significantly.

Zero-shot performance of CLIP Given the generalizability of CLIP [48] on several other datasets, it is also interesting to check how it would perform in a zero shot transfer setting to the Text2Shape [7] dataset. To use CLIP for our retrieval task, we first feed the 12 multi-view images of an object into the image encoder for CLIP separately then average the vectors to get the image embedding. Specifically we use the ViT-B/32 pretrained model from CLIP. For retrieval, we encode the text using the CLIP text encoder and then retrieve relevant shapes by taking the dot product of the text and shape

	batch size	RR@1(↑)	RR@5(↑)	NDCG@5(↑)	MRR(↑)
Bi(**I**)	32	8.07 ± 0.20	23.68 ± 0.43	16.00 ± 0.31	16.67 ± 0.28
	64	8.25 ± 0.31	24.52 ± 0.49	16.52 ± 0.32	17.09 ± 0.29
	128	**8.69 ± 0.38**	**25.29 ± 0.46**	**17.14 ± 0.42**	**17.63 ± 0.38**
	256	7.73 ± 0.22	23.46 ± 0.51	15.70 ± 0.36	16.36 ± 0.32
Bi(**V**)	32	7.41 ± 0.20	23.59 ± 0.41	15.60 ± 0.26	16.36 ± 0.24
	64	8.35 ± 0.47	25.50 ± 0.44	17.06 ± 0.37	17.68 ± 0.38
	128	**8.86 ± 0.16**	26.41 ± 0.50	17.79 ± 0.30	18.34 ± 0.20
	256	8.81 ± 0.36	**26.78 ± 0.51**	**17.96 ± 0.40**	**18.45 ± 0.37**

Table 4.7: Comparison of batch-size on shape retrieval for Bi(**I**) and Bi(**V**) on the validation set. We find that increasing the batch size increases the performance. However, for Bi(**I**), the performance decreased for largest batch size we tried (256). This could be due to overfitting on the limited amount of negative, or the presence of more noisy negatives in the large batch.

	RR@1	RR@5	NDCG@5
Text2shape [7]	0.40	2.37	1.35
Y2Seq2Seq [28]	2.93	9.23	6.05
Part2Word [57] (part)	7.94	23.89	16.03
CLIP [48]	1.40	4.08	2.72
CLIP [48](Norm)	1.63	5.34	3.47

Table 4.8: Comparison of text to shape retrieval performance using CLIP zero shot against prior work on the test set. We report the recall rate (RR@1, RR@5) and NDCG@5 as percentages. It can be seen that CLIP has relatively good performance considering that it has not been trained on the Text2Shape [7] dataset.

	resolution	RR@1(↑)	RR@5(↑)	NDCG@5(↑)	MRR(↑)
	64	7.41 ± 0.34	22.62 ± 0.50	15.13 ± 0.40	15.86 ± 0.35
Bi(**I**)	128	8.69 ± 0.39	25.30 ± 0.47	17.15 ± 0.43	17.64 ± 0.39
	224	**8.85 ± 0.21**	**25.51 ± 0.36**	**17.31 ± 0.18**	**17.81 ± 0.17**
	32	6.62 ± 0.24	21.81 ± 0.41	14.30 ± 0.28	15.20 ± 0.24
Bi(**V**)	64	**8.86 ± 0.16**	**26.41 ± 0.50**	**17.79 ± 0.30**	**18.34 ± 0.20**

Table 4.9: Comparison of resolution settings on shape retrieval for Bi(**I**) and Bi(**V**) on the validation set. We find that increasing the resolution increases the performance.

embeddings. We compare the unnormalized CLIP embedding as well as the normalized CLIP embedding (Norm), which is equivalent to taking the cosine similarity as we do with our model. The results can be seen in Tab. 4.8. Although its performance is not on par with recent SoTA, it is impressive that it can beat the baseline method from the original Text2Shape [7] without being trained on the dataset.

4.6 Qualitative evaluation

Custom sentences We tried several custom sentences which are not in the dataset. Fig. 4.1 shows the best matching shapes each model predicts. This shows that our network is able to allow users to easily and rapidly search through large 3D collections.

Sentences from the dataset

Fig. 4.2 shows successful retrievals of shapes using Tri(**I+V**), our best performing model. 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 5 the text describes *a boxy look gray chair*. The retrieved shapes all match the description, but the last four would be negatives in our training process and the retrieval metrics.

Failure cases

Fig. 4.3 shows example failure cases of our model. While the top 2 shapes in the first row have *a slot to keep things*, the other retrieved shapes in the top 5 do not. The second row shows the challenge of retrieving shapes with rare attributes. While there are many *plain square wooden table(s)* in the dataset, there are far fewer tables that *can be folded*. So the network focuses more on *square wooden* and ignores *can be fold*. We find that it is easier for the network to learn frequently occurring characteristics such as shape and texture, but some descriptions (e.g. for articulations) are likely too abstract and infrequent for our current approach.

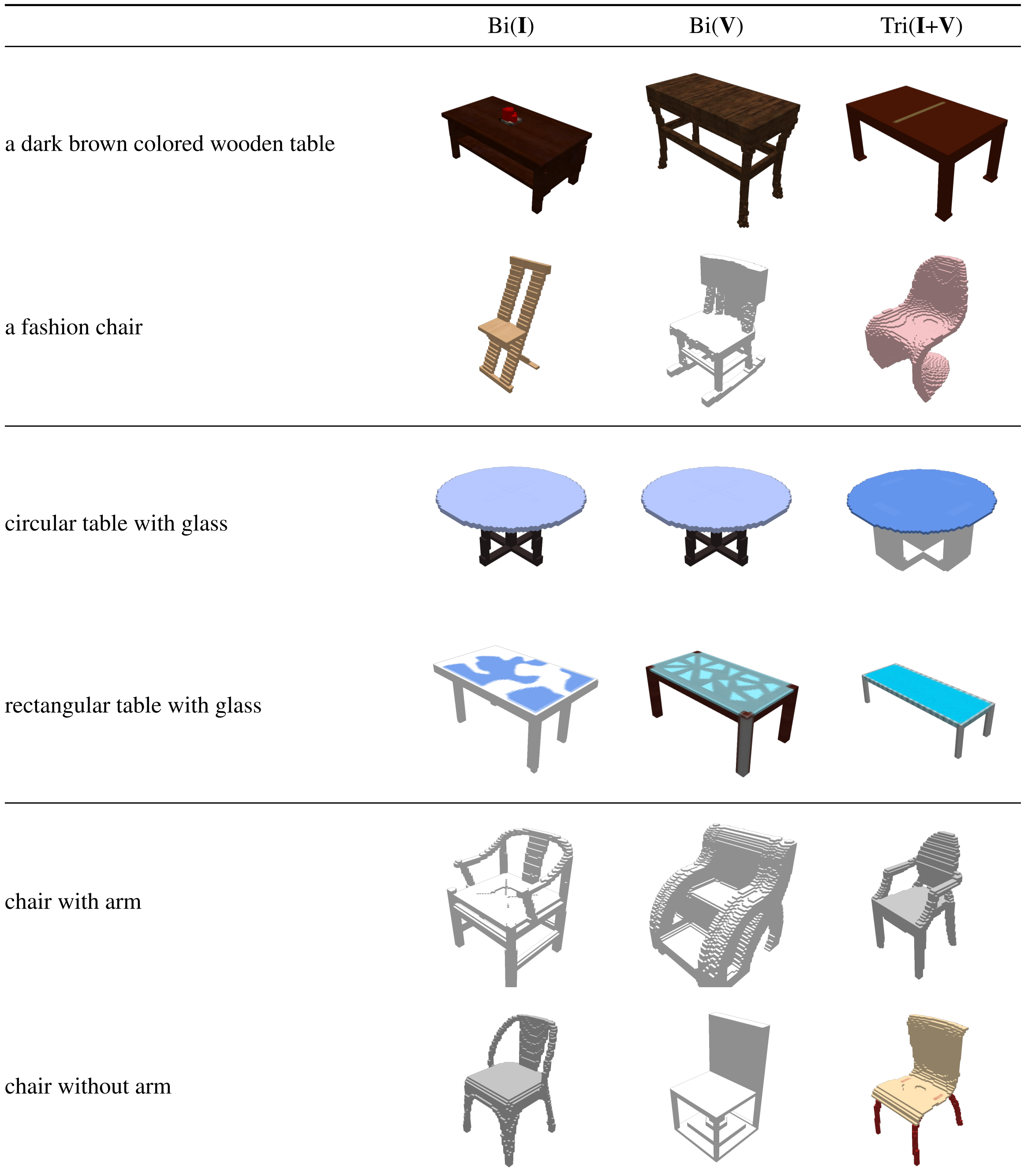


Figure 4.1: Retrieved shapes from test set using Bi(**I**), Bi(**V**), and Tri(**I**+**V**) for custom sentences. Note that all models are able to retrieve shapes that match the color (*dark brown*) and material appearance (*wooden*, *glass*), shape (*circular*, *rectangular*), and the presence and absence of arms (last two rows).

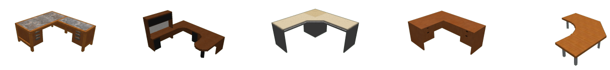
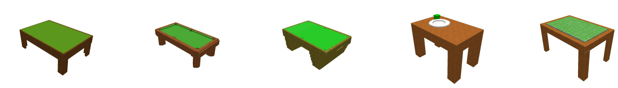
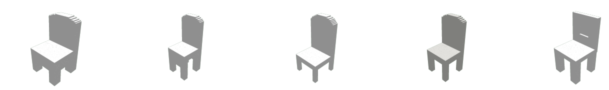
		top1	top2	top3	top4	top5
1	an L-shaped dark brown colored wooden table. 	17.31	6.60	20.34	GT	2.19
2	a luxurious gray leather modern concept plush chair with stainless steel frame foots 	GT	2.89	9.50	1.78	3.42
3	simple circular table with no leg and only one circular base. 	0.79	5.77	0.59	GT	6.32
4	This is greenish top wooden billiards table. 	15.79	GT	8.04	19.59	3.18
5	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. 	GT	4.64	22.32	12.97	11.42
6	wooden armless dining room chair with open nine-square back. 	GT	19.36	13.31	18.78	12.38

Figure 4.2: Successful retrieval results on the test set with Tri(**I+V**). For each description, we use our proposed model to retrieve the top-5 shapes. We show the $F1^{0.1}$ 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). This figure shows that our network has good language grounding ability overall. It can retrieve shapes that match *L-shaped* (row 1), *stainless steel frame foots* (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).

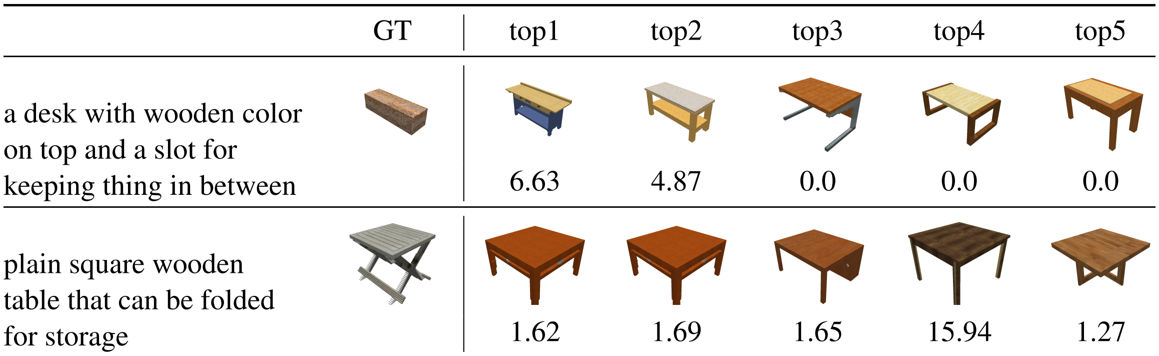


Figure 4.3: Examples of failed retrievals on the test set with Tri(**I**+**V**). The ground truth (GT) is shown in the first column, followed by the retrieved results with the $F1^{0.1}$ score for each. We see that some descriptions do not accurately describe the GT shape (first row), and retrieval of shapes with rare attributes such as being foldable is hard (second row).

	match	color mismatch	big shape error	small shape error	missing part
Bi(**I**)	106	65	22	85	5
Bi(**V**)	103	67	26	76	5
Tri(**I**+**V**)	113	64	17	74	5

Table 4.10: 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.

4.7 Error 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 Fig. 4.4 for examples and Tab. 4.10 for analysis summary). As expected from the quantitative results, Tri(**I**+**V**) has the most number of shapes that match the description. With the limited number of queries we examined, all models have similar performance on color and missing part. The Bi(**I**) model had difficulty getting small shape details correct, and Tri(**I**+**V**) obtained the best performance on matching the overall shape.

4.8 Evaluation on ShapeNet 13 categories

To show the effectiveness of our method for retrieval beyond 'chairs and tables', we also collected a set of descriptions for 11 additional categories of objects from ShapeNet [5]. Using a similar setting as Chen et al. [7], we asked Amazon Mechanical Turk workers to provide descriptions for a random set of 5 objects. We restricted workers to high quality workers (with acceptance rate

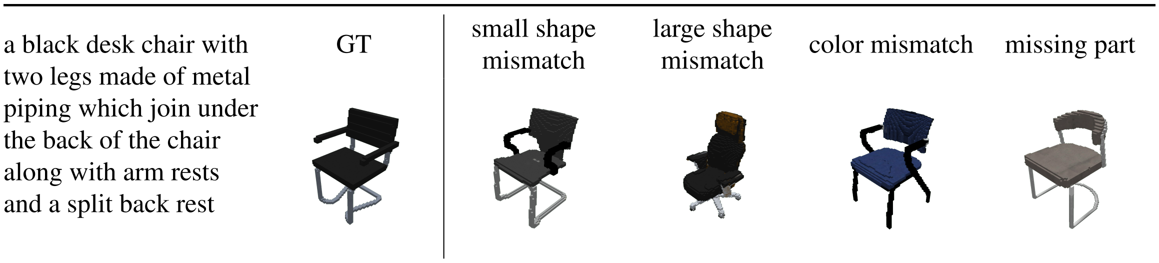


Figure 4.4: Examples of the four types of error we analyzed in our manual analysis.

Category	Modality	Train	Validation	Test	Modality	Train	Validation	Test
Table		33613	4170	4144		6726	831	826
Chair		26254	3503	3387		5222	696	675
Sofa		12645	1565	1588		2521	311	316
Lamp		9249	1077	1293		1844	215	258
Loudspeaker		6417	850	707		1279	170	141
Cabinet		6374	743	816		1266	146	159
Display		4397	520	552		875	104	110
Bathtub	Text	3535	360	397	Shape	703	72	79
Clock		2510	367	342		500	73	68
Bookshelf		1776	255	190		354	51	38
Trash bin		1451	145	125		289	29	25
File cabinet		1160	142	130		231	28	26
Bed		905	125	135		181	25	27
Total		110286	13822	13806		21991	2751	2748

Table 4.11: Statistics for the ShapeNet [5] C13 dataset we use.

of $> 95\%$ on more than 200 HITs) from countries that have English as a native language (US, Canada, Great Britain, Australia). We collected up to 5 descriptions per object. This together with the original 'chairs and tables' dataset, results in a dataset with over 138K descriptions for $27,510$ objects across 13 object categories (see Table 4.11 for statistics of each object category).

We use the same standard settings as in our experiments on the 'chairs and tables' dataset with resolution 128^2 and 64^3 for images and voxels respectively and a batch size of 128. The results shown are over 1 run. Experiments on this ShapeNet C13 dataset shows a similar trend as for the 'chairs and tables' dataset, with Tri(**I**+**V**) outperforming Bi(**I**) and Bi(**V**) (see Table 4.12). Comparison against CLIP and CLIP (Norm) show that the pretrained CLIP model is able retrieve relevant shapes in a zero-shot setting on this broader set of shapes.

4.9 Evaluation on the Primitives dataset

We also verify our results on the primitives dataset introduced by Chen et al. [7]. The primitives dataset is a diagnostic dataset consisting of simple shapes with different colors and sizes. Unlike

	RR@1(↑)	RR@5(↑)	NDCG@5(↑)
CLIP	1.69	5.85	3.77
CLIP (Norm)	2.23	7.3	4.76
Bi(**I**)	8.44	25.64	17.15
Bi(**V**)	8.87	27.91	18.52
Tri(**I+V**)	**10.63**	**31.01**	**21.03**

Table 4.12: Shape retrieval results on ShapeNet c13 val set. We observe a similar trend as on the 'chairs and tables' dataset with Tri(**I+V**) outperforming the other models

	RR@1(↑)	RR@5(↑)	NDCG@5(↑)
Text2shape [7]	95.07	99.08	95.51
Y2Seq2Seq [28]	96.66	97.57	95.87
Bi(**V**)	**98.18**	**99.78**	**99.18**

Table 4.13: Shape retrieval quantitative results on primitives test set.

the 'chairs and tables' from ShapeNet [5], the descriptions are generated using templates and it is known exactly what shapes each description should match On this simplified dataset, our model clearly outperforms prior work as shown in Tab. 4.13. Here we do only one run from a random seed. Note that the performance for primitives is already quite saturated, so we do not run other bimodal models or trimodal models on it

4.10 Extending to shape-to-text retrieval

While our focus is text to shape retrieval, it is also possible to use our joint embedding for shape-to-text retrieval. Tab. 4.14 shows that our trimodal embedding trained with NT-XEnt loss is able to outperform prior work on shape-to-text retrieval. Fig. 4.5 shows some qualitative examples for shape-to-text retrieval.

	RR@1	RR@5	NDCG@5
Text2shape [7]	0.94	3.69	0.85
Y2Seq2Seq [28]	6.77	19.30	5.30
Part2Word [57] (end2end)	9.55	28.45	8.01
Part2Word [57] (part)	13.18	34.52	9.94
Bi(**I**) (ours)	11.91	32.69	9.37
Bi(**V**) (ours)	13.07	35.62	10.33
Tri(**I+V**) (ours)	**16.33**	**42.52**	**12.73**

Table 4.14: Shape to text retrieval comparison against prior work on the test set. We report the recall rate (RR@1, RR@5) and NDCG@5 as percentages. We train with a batch size of 128, 64^3 voxels, and 6 multi-view images at a resolution of 128^2 each. Our trimodal embedding (Tri(**I+V**)) outperforms prior work.

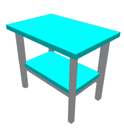
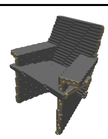
	the table is circular with three legs . the table is black and the legs stick out from the top. a black color round shaped wooden table with three legs a black colored round table with four slim shaped legs black round metal outdoor table with long curled legs . black round table three legs wooden material
	a wooden chair red in color it is a wooden chair . it is red in color . a wooden chair with red colour back and seat with spindle and strong four legs a red wooden kitchen chair with detached back and slightly rounded seat this is wooden chair with four legs and it is in red texture light weight
	this oval light wood topped table is on a dark wood base . a wooden oval brown small table . it has a rectangular hole at the middle below the table top seems like it has two legs . an oval shaped table with two legs . it is also wooden and brown . an brown oval table with three section base brown color rectangle shape wood material and physical appearance table
	a lounge style wooden chair for a porch . a wooden deckchair that you can stretch your legs on it is a white wooden adirondack beach chair . rectangular resting swinging chair light brown coloured solid physical appearance wooden with hands for resting lawn chair made of wood with a reclining back and arm green in color .
	rectangular blue table with wheels . a light colour rectangular horizontal table top has blue colour four legs with centralized ladder like bottom . a two tiered table with bright blue surfaces . the top tier is a rectangle and the bottom tier a slightly smaller rectangle with silver metal legs connecting them blue colour rectangular shape wooden table with moving wheels a bright cyan coloured table supported by four legs and there is another floor under the table top . the legs has wheels
	it is a gray rocking chair . wooden rocking chair with armrests and gray cushion . a wooden rocking chair with rest and back gray color cloth . a chair with wooden arms both side . gray technically designed chair with flat armrest and backrest . a chair designed well .

Figure 4.5: Retrieved descriptions from test set using Tri(**I**+**V**) for example shapes. The ground-truth description is shown in green. Parts of the descriptions that do not accurately match the shape are underlined. The retrieved descriptions mostly match the input shape and can capture the color *black* and overall shape well. However, the model has trouble with fine details (*spindle*, *wheels*), part-level colors (*light wood topped*, *arm green*), and functionality (*swinging*, *rocking*).

Chapter 5

Experiments on the SNARE Dataset

In addition to applying our method to text-to-shape retrieval, we also investigate how well it can work on the reference game task, where given two shapes and a description, the goal is to determine which shape the description refers to. To explore this, we apply our method to the SNARE dataset [58], which covers 262 shape categories. This also allows us to investigate how well our method works on a broader range of shape categories (more than just the tables and chairs from the Text2Shape dataset [7]).

5.1 Dataset

Shape**N**et **A**nnotated with **R**eferring **E**xpressions(SNARE) [58] is a benchmark for using natural language referring expressions to distinguish 3D objects. The SNARE dataset is built on ACRONYM [15], a dataset composed of 3D models selected from ShapeNetSem [5, 52] for the robot grasp planning. The SNARE dataset contains 7,881 ACRONYM object models and over 50K natural language referring expressions to distinguish between two objects. We show several examples from the SNARE dataset in the Fig. 5.1.

During the annotation process, crowdworkers from Amazon Mechanical Turk (AMT) are asked to provide natural language expressions that can discriminately describe an object. Two ShapeNet objects from the same category were presented side-by-side to AMT workers. These workers had to answer the question: *In order to differentiate Object A from Object B, how to describe Object A?* These AMT workers were asked to look at the object and provide visual expressions, or imagine they are blindfolded and provide tactile expressions. Hence, visual expressions involve colors and category, while tactile expressions involve shapes and contours. An example object with three visual expressions and three tactile expressions is shown in Fig. 5.2.

The vocabulary of the SNARE dataset contains 6567 unique words. For the images, the SNARE dataset uses the 8 views of rendered images for each model provided by ShapeNetSem [5, 52]. The cameras pointed towards the center of the object while changing the azimuth. The azimuth angles are spaced by 45 degrees relative to the previous camera. We follow the train/val/test split established by SNARE [58] (see Tab. 5.1 for statistics).

Data	Split	# Cats	# Objs	# Ref Exps
Text2Shape	Train	2	11921	59777
	Val	2	1486	7435
	Test	2	1492	7452
	Total	2	14899	74664
SNARE	Train	207	6153	39104
	Val	7	371	2304
	Test	48	1357	8751
	Total	262	7881	50159

Table 5.1: Dataset breakdown statistics contrast for Text2Shape and SNARE.

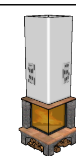
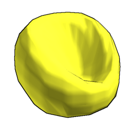
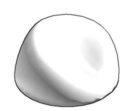
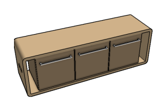
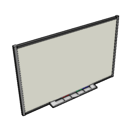
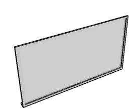
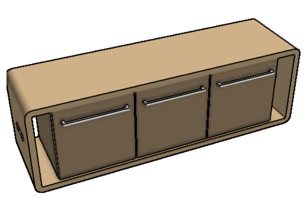
referring expression		Object A	Object B	ans
top of mantle is long, flat and narrow, back of fireplace extends well backward.	tactile			A
has a wide top area	tactile			A
beige counter	visual			B
black flat screen television	visual			A

Figure 5.1: Examples of SNARE dataset. Each entry includes one referring expression, one value showing this expression is visual or tactile), two objects, and the referent object of this referring expression.



visual	light brown bench beige counter brown storage area
tactile	has three compartments in the front storage with 3 sections that open from the front rectangle with openings toward ends

Figure 5.2: Sample of object and its visual/tactile referring expressions in SNARE. Visual expressions focus on colors and category. Tactile expressions focus on contours and shapes.

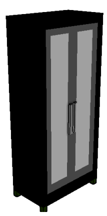
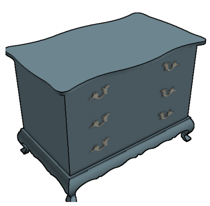
Object \ Expr	pizza topped with pepperoni and basil	pizza with ham	tall wardrobe	has six handles on the front
	positive	negative	negative	negative
	negative	positive	negative	negative
	negative	negative	positive	negative
	negative	negative	negative	positive

Figure 5.3: Example of a batch with size 2. We load two entries each step. One entry gives two referring expression and two objects. So in total there are four referring expressions and four objects.

5.2 Contrastive learning for SNARE

To train our model for SNARE dataset, we construct our training data as follows. We randomly sample a minibatch of N tuples, each of which contains a pair of objects (Object A and Object B) and a referring expression describing one of the objects. We then create additional positive pairs for each tuple, by taking the distractor object (the object that does did not have a matching referring expression) and randomly sampling a referring expression from the dataset that matched the distractor object. This way of building a minibatch results in $2N$ positive pairs. For negative samples, rather than explicitly sampling them, for each positive pair, we treat the object paired with the other $2N-1$ referring expressions as negative. Fig. 5.3 gives an example of a batch with size 2.

After constructing positive pairs and negative pairs, we can apply the NT-Xent loss on these pairs. 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 $2N$ positive pairs $(u_{m_1 j}, u_{m_2 j})$ for embeddings belonging to the same object and $(2N)^2 - (2N)$

negative pairs $(u_{m_1 j}, u_{m_2 k})_{j \neq k}$ for different objects. The contrastive loss is the same as the loss in Chapter 3.

5.3 Implementation details

The architecture and hyperparameters in the network are exactly the same as the network we used on Text2Shape dataset [7]. We use a one-layer BiGRU as the text encoder, a 3D CNN architecture as the voxel encoder, and an MVCNN with a pretrained ResNet18 backbone as the image encoder. We use random weights initialization for text encoder, pretrained weights for image encoder and xavier [21] uniform weights initialization for voxel encoder.

For text input, we perform word tokenization by building a dictionary from unique word to index and mapping expressions to arrays. For voxel input, we generate colored solid voxels based on the solid voxels provided by ShapeNetSem [5, 52]. For image input, we use the pre-rendered screenshots provided as part of ShapeNetSem [5, 52]. We resize the original screenshots at resolution 512×512 to 128×128. ShapeNetSem [5, 52] provides 14 rendered images of each model, 6 canonical orientations (front, back, left, right, bottom, top) and 8 images rendered around every object at 45 degree intervals. Thomason et al. [58] skip the first 6 canonical orientated images and only use the remaining 8 images, which is also what we do to be consistent with SNARE [58].

Similar to TriCoLo, the range of values of images and voxels is normalized from 0-255 to 0-1. We use Adam optimizer with learning rate 0.0004. Our model is trained for a maximum of 30 epochs until convergence. The checkpoint that gives the best validation accuracy is saved. With batch size 128, we use 1 V100 GPU (32 GB of memory) to train our models. Training Bi(**I**) takes around 3 hours, training Bi(**V**) takes around 5 hours, and training Tri(**I+V**) takes around 6 hours.

5.4 Experiments

5.4.1 Models

We compare our bimodal and trimodal models on the SNARE dataset against CLIP [48] based models introduced by Thomason et al. [58].

CLIP-based methods Thomason et al. [58] introduces three methods: zero-shot CLIP, Language-View Match (MATCH) and Language Grounding through Object Rotation (LAGOR). Their zero-shot CLIP method is a zero-shot classifier which uses embeddings from frozen transformer-based sentence encoder and ViT-B/32 image encoder in CLIP [48]. The CLIP method identifies the referred object according to the larger cosine similarity between the text and image embeddings. MATCH method adds a classification network after a frozen CLIP [48] backbone. Text embeddings and image embeddings from CLIP encoders are concatenated and pass through the multi-layer perceptron (MLP). The MLP layers calculate a matching score for the expression and the object. MATCH method can be interpreted as a fine-tuned CLIP method implemented by adding a predictive head over a frozen CLIP backbone. The LAGOR method builds on top of MATCH and adds view estimation as an auxiliary loss. LAGOR uses a pretrained-MATCH module taking in two images and learns an additional multi-layer perceptron to predict the view indices of the two input images. Thomason et al. [58] trained MATCH method with single, two and eight views, and trained LAGOR with two views. When training with less than 8 views, the views are randomly selected from the 8 views at each step.

Our models We compare our bimodal models trained with text-image (Bi(**I**)), text-voxels (Bi(**V**)), and all three modalities (Tri(**I+V**)). We provide Bi(**I**) with one, two, and eight images. Following Thomason et al. [58] the views are chosen at random during each step for one image and two images. We provide Bi(**V**) with texts and voxels at 64^3 resolution, and we provide Tri(**I+V**) with texts, voxels at 64^3 resolution and all eight images. We choose the object as the model's prediction whose shape embedding has the larger cosine similarity to the given text embedding

5.4.2 Evaluation metrics

We follow prior work on the SNARE dataset and use the discriminative accuracy for quantitative comparisons. We count how many predicted referents from the expression are correct and calculate the accuracy of the validation set. We report the overall accuracy (acc(All)), as well as the accuracy for the visual referring expressions (acc(Visual) and the tactile referring expressions (acc(Tactile)).

$$\text{acc(All)} = \frac{\#\text{Correct predictions}}{\#\text{All expressions}} \tag{5.1}$$

$$\text{acc(Visual)} = \frac{\#\text{Correct predictions given visual expressions}}{\#\text{Visual expressions}} \tag{5.2}$$

$$\text{acc(Tactile)} = \frac{\#\text{Correct predictions given tactile expressions}}{\#\text{Tactile expressions}} \tag{5.3}$$

5.4.3 Quantitative evaluation

Comparison with prior work Tab. 5.2 shows the comparison of our results and the results from SNARE [58]. These numbers are in percentage. For Bi(**I**), our method is better than zero-shot CLIP but a bit lower than MATCH method irrespective of the number of views used. For Bi(**V**), there is an obvious drop in accuracy. The reason might be that our voxel encoder is a self-designed 3D CNN, and we train it from scratch. However, methods in SNARE [58] use pre-trained CLIP ViT-B/32 image encoder which has seen lots of images and categories, and has learned sufficient prior knowledge. This voxel encoder also has a negative impact on the performance of Tri(**I+V**). While the accuracy of Bi(**I**) with eight views is 78.2%, the accuracy of Tri(**I+V**) with eight views has a significant decrease and reached to 73.5%.

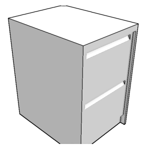
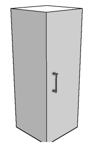
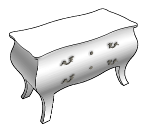
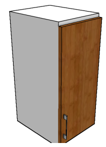
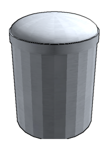
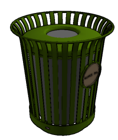
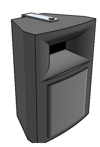
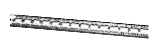
referring expression		Object A	Object B	ans	pred
a white file cabinet	tactile			A	A
short and wide wooden stand	visual			A	A
white cabinet with a brown door	visual			B	B
bin with lid	visual			A	A
round speaker	tactile			A	A
clear transparent ruler	visual			B	B

Figure 5.4: Visualization of successful predictions given by our best performing model Bi(**I**) with two images. Our model can ground language about colors(*white*, *brown*), shapes(*short*, *wide*, *round*), texture(*wooden*, *transparent*) and parts(*lid*).

Model	Views	Visual	Tactile	All
Bi(**V**)	-	79.7	65.3	72.6
CLIP	Single	79.0±0.0	63.0±0.0	71.1±0.0
MATCH	Single	88.4±0.4	73.3±0.6	80.9±0.4
Bi(**I**)	Single	88.2	68.7	78.6
CLIP	Two	81.0±0.0	64.1±0.0	72.6±0.0
MATCH	Two	89.2±0.6	74.4±0.7	81.8±0.4
LAGOR	Two	89.8±0.4	75.3±0.7	82.6±0.4
Bi(**I**)	Two	88.5	70.2	79.5
ViLBERT	Eight	89.5	76.6	83.1
CLIP	Eight	83.7±0.0	65.2±0.0	74.5±0.0
MATCH	Eight	89.2±0.9	75.2±0.7	82.2±0.4
Bi(**I**)	Eight	88.4	67.8	78.2
Tri(**I**+**V**)	Eight	82.3	64.5	73.5
Human(U)	Eight	94.0	90.6	92.3
Human(M)	Eight	100.0	100.0	100.0

Table 5.2: Accuracy on SNARE dataset. Shaded rows indicate our method.

5.4.4 Qualitative evaluation

Successful cases Fig. 5.4 shows successful discriminations when we use Bi(**I**) with two views, our best performing model on this dataset. Our model successfully grounds language which describes the color(*white*, *brown*, *transparent*), shape(*short*, *wide*, *round*) and texture(*wooden*).

Failure cases Through looking into all the failure cases when we use our best performing model Bi(**I**) with two views, we identify several challenges. Fig. 5.5 shows it is difficult for Bi(**I**) to capture tiny details because the resolution of images is too low to show the details clearly. For example, *knob*, *numbers* and *wheels* are very small on the images, so there are only a small amount of pixels showing them. Fig. 5.6 shows the challenge of understanding number of parts for our Bi(**I**) model. For instance, both objects have the door handle in row 2, but the left object has only one handle and the right object has three handles. Our model would predict the wrong drawer. Fig. 5.7 illustrates it is demanding for our Bi(**I**) model to understand the articulation functions. To be specific, in Fig. 5.7 our model cannot discriminate the status of the drawer or door is open or closed.

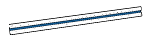
referring expression		Object A	Object B	ans	pred
simple rectangle with knob near bottom	tactile			A	B
ruler with numbers	tactile			A	B
wheels are at base of can.	tactile			A	B

Figure 5.5: Examples of failure cases due to tiny details given by our best performing model Bi(**I**) with two images. It is hard for the mdoels (and people) to notice *knob* on the cabinet (row 1), *numbers* on the ruler (row 2), *wheels* on the can (row 3).

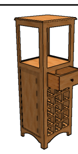
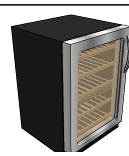
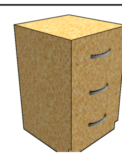
referring expression		Object A	Object B	ans	pred
Wardrobe with two doors	tactile			A	B
box with one door handle	tactile			A	B
speaker with four circles on it	tactile			A	B

Figure 5.6: Examples of failure cases due to the number of parts given by our best performing model Bi(**I**) with two images. In row 1, the Object A has two doors, while the Object B has many shelves. In row 2, the Object A has only one handle, while Object B has three handles. In row 3, Object A has four circles, while Object B has only two circles. When the two candidates share a common part but have different number of it and the referring expressions specify this divergence, our model failed to understand.

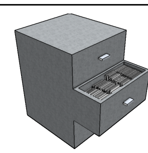
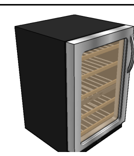
referring expression		Object A	Object B	ans	pred
cabinet with closed drawers	tactile			A	B
rectangular cabinet with both doors open	tactile			A	B
down part is opened	tactile			A	B

Figure 5.7: Examples of failure cases due to the articulation state given by our best performing model Bi($\mathbf{I}$) with two images. In row 1, the drawers in the Object A are all closed, while one drawer in the Object B is open. In row 2, both doors are open in the Object A, while only one door is open in the Object B. In row 3, the two doors at the bottom of the left objects are open, while the door on the right object is closed. When referring expressions involve the state of the door, drawer or other articulations on the object, our model failed to understand the state.

Chapter 6

Conclusion

In this thesis, we show our trimodal embedding outperforms the bimodal embeddings for text-to-shape retrieval both on the ‘chairs and tables’ dataset from Text2Shape [7] as well as a larger dataset from ShapeNet [5] with 13 categories (ShapeNet C13). We also demonstrate that contrastive loss is able to build effective joint embeddings for the SNARE [58] dataset without requiring models to be pretrained on CLIP [48]. But we point out that our work has several limitations:

- We have restricted our study to voxel-based 3D representations, with which it is often hard to capture geometric details and fine-grained surface textures. It would be interesting to add other modalities such as point clouds, depth images, and textured 3D polygonal meshes which may help alleviate these limitations.
- One big challenge of incorporating additional modalities is the memory cost. Future work might use momentum encoder [31] to work around this obstacle.
- In addition, we focused on a specific type of contrastive loss. It would be possible to consider other contrastive losses, data augmentation, as well as introducing other loss terms such as captioning loss and reconstruction loss. Also, the current contrastive loss ignores the fact that there might be false negative pairs in a mini-batch due to the descriptions being ambiguous.

In our work, we showed that incorporating 3D voxels was useful to the text-to-shape retrieval task on both the ‘chairs and tables’ Text2Shape dataset and ShapeNet C13. However, incorporating 3D voxels for the SNARE dataset was not as useful. We believe that this is because the 2D image encoder was pre-trained on more data, and thus more robust. The 2D images also has higher resolution and contained richer texture information which can be used for fine-grained disambiguation. Nevertheless, we believe that having 3D (voxel) representation could potentially help with text-to-shape retrieval and the investigation of joint language and 3D representations would be an important direction. Unlike 2D images, 3D representations have spatial consistency, are not subject to occlusions, and are helpful for view agnostic tasks. To thoroughly investigate this, more targeted datasets and more fine-grained analysis is necessary.

In summary, we carried out a systematic study of contrastive losses for text to shape retrieval. With careful tuning of hyperparamters, we show that using simple contrastive losses can outperform the current SoTA text to shape retrieval method which relies on extra annotation. In addition, we proposed a trimodal contrastive loss which further improves over the text to shape retrieval SoTA by considering both 2D and 3D representations. We believe our work can serve as a good foundation for followup work in text to shape retrieval and will inspire further analysis of other datasets and tasks.

Bibliography

[1] Panos Achlioptas, Judy Fan, Robert Hawkins, Noah Goodman, and Leonidas J Guibas. ShapeGlot: Learning language for shape differentiation. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019.

[2] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. ReferIt3D: Neural listeners for fine-grained 3D object identification in real-world scenes. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[3] Hassan Akbari, Linagzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. VATT: Transformers for multimodal self-supervised learning from raw video, audio and text. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[4] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelovic, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[5] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[6] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. ScanRefer: 3D object localization in RGB-D scans using natural language. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[7] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, 2018.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.

[9] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[10] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2Cap: Context-aware dense captioning in RGB-D scans. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[11] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[13] Herbert H Clark and Deanna Wilkes-Gibbs. Referring as a collaborative process. *Cognition*, 22(1):1–39, 1986.

[14] Jasmine Collins, Shubham Goel, Achleshwar Luthra, Leon Xu, Kenan Deng, Xi Zhang, Tomas F Yago Vicente, Himanshu Arora, Thomas Dideriksen, Matthieu Guillaumin, and Jitendra Malik. ABO: Dataset and benchmarks for real-world 3D object understanding. *arXiv preprint arXiv:2110.06199*, 2021.

[15] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. Acronym: A large-scale grasp dataset based on simulation. In *International Conference on Robotics and Automation (ICRA)*, 2021.

[16] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. VSE++: Improving visual-semantic embeddings with hard negatives. In *Proceedings of the British Machine Vision Conference*, 2018.

[17] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[18] Michael C Frank and Noah D Goodman. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998, 2012.

[19] Andrea Frome, Greg Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. DeViSE: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[20] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3D-FUTURE: 3D Furniture shape with TextURE. *arXiv preprint arXiv:2009.09633*, 2020.

[21] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2010.

[22] Dave Golland, Percy Liang, and Dan Klein. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2010.

[23] Google. Google scanned objects. `https://app.ignitionrobotics.org/GoogleResearch/fuel/collections/GoogleScannedObjects`, 2021. Accessed: 2021-10-30.

[24] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[25] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *NeurIPS*, 2020.

[26] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2006.

[27] Philip Haeusser, Alexander Mordvintsev, and Daniel Cremers. Learning by association–a versatile semi-supervised training method for neural networks. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[28] 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.

[29] Zhizhong Han, Chao Chen, Yu-Shen Liu, and Matthias Zwicker. ShapeCaptioner: Generative caption network for 3D shapes by learning a mapping from parts detected in multiple views to sentences. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.

[30] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[31] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[32] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[33] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3D instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[34] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[35] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[36] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

[37] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[38] Weicheng Kuo, Anelia Angelova, Tsung-Yi Lin, and Angela Dai. Mask2CAD: 3D shape prediction by learning to segment and retrieve. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[39] Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[40] Yunze Liu, Qingnan Fan, Shanghang Zhang, Hao Dong, Thomas Funkhouser, and Li Yi. Contrastive multimodal fusion with TupleInfoNCE. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

[41] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[42] Sijie Mai, Ying Zeng, Shuangjia Zheng, and Haifeng Hu. Hybrid contrastive learning of tri-modal representation for multimodal sentiment analysis. *arXiv preprint arXiv:2109.01797*, 2021.

[43] Patrick Min, Michael Kazhdan, and Thomas Funkhouser. A comparison of text and shape matching for retrieval of online 3D models. In *International Conference on Theory and Practice of Digital Libraries*, 2004.

[44] Will Monroe, Robert XD Hawkins, Noah D Goodman, and Christopher Potts. Colors in context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics*, 5:325–338, 2017.

[45] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[47] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[49] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

[50] Junha Roh, Karthik Desingh, Ali Farhadi, and Dieter Fox. LanguageRefer: Spatial-language model for 3D visual grounding. In *Proceedings of Conference on Robot Learning (CoRL)*, 2021.

[51] Yue Ruan, Han-Hung Lee, Ke Zhang, and Angel X Chang. Tricolo: Trimodal contrastive loss for fine-grained text to shape retrieval. *arXiv preprint arXiv:2201.07366*, 2022.

[52] Manolis Savva, Angel X Chang, and Pat Hanrahan. Semantically-enriched 3d models for common-sense knowledge. cvpr 2015 workshop on functionality. *Physics, Intentionality and Causality*, 7, 2015.

[53] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[54] SketchFab. Sketchfab. `https://sketchfab.com/features/free-3d-models`, 2021. Accessed: 2021-10-30.

[55] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[56] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[57] Chuan Tang, Xi Yang, Bojian Wu, Zhizhong Han, and Yi Chang. Part2Word: Learning joint embedding of point clouds and text by matching parts to words. *arXiv preprint arXiv:2107.01872*, 2021.

[58] Jesse Thomason, Mohit Shridhar, Yonatan Bisk, Chris Paxton, and Luke Zettlemoyer. Language grounding with 3D objects. In *Proceedings of Conference on Robot Learning (CoRL)*, 2022.

[59] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[60] Trimble. Sketchup 3D warehouse. `https://3dwarehouse.sketchup.com`, 2021. Accessed: 2021-10-30.

[61] TurboSquid. Turbosquid. `https://www.turbosquid.com/Search/3D-Models`, 2021. Accessed: 2021-10-30.

[62] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[63] Xiaoshi Wu, Hadar Averbuch-Elor, Jin Sun, and Noah Snavely. Towers of babel: Combining images, language, and 3D geometry for learning multimodal vision. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

[64] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D shapenets: A deep representation for volumetric shapes. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[65] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[66] Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Sheng Wang, Zhen Li, and Shuguang Cui. InstanceRefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

[67] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis Langlotz. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*, 2021.

[68] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3DVG-transformer: Relation modeling for visual grounding on point clouds. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.