Fine-Tuned but Zero-Shot 3D Shape Sketch View Similarity and Retrieval

Gianluca Berardi[1,2] Yulia Gryaditskaya[2]

[1]Department of Computer Science and Engineering (DISI), University of Bologna, Italy
[2]CVSSP and Surrey Institute for People-Centred AI, University of Surrey, UK

Abstract

Recently, encoders like ViT (vision transformer) and ResNet have been trained on vast datasets and utilized as perceptual metrics for comparing sketches and images, as well as multi-domain encoders in a zero-shot setting. However, there has been limited effort to quantify the granularity of these encoders. Our work addresses this gap by focusing on multi-modal 2D projections of individual 3D instances. This task holds crucial implications for retrieval and sketch-based modeling. We show that in a zero-shot setting (without retraining on a specific shape category or sketch style), the more abstract the sketch, the higher the likelihood of incorrect image matches. Even within the same sketch domain, sketches of the same object drawn in different styles, for example by distinct individuals, might not be accurately matched. One of the key findings of our research is that meticulous fine-tuning on one class of 3D shapes leads to improved performance on other shape classes (fine-tuned but zero-shot), reaching or surpassing the accuracy of supervised methods. We compare and discuss several fine-tuning strategies. Additionally, we delve deeply into how the scale of an object in a sketch influences the similarity of features at different network layers, helping us identify which network layers provide the most accurate matching. Significantly, we discover that ViT and ResNet perform best when dealing with similar object scales. We believe that our work will have a significant impact on research in the sketch domain, providing insights and guidance on how to adopt large pretrained models as perceptual losses. Our code is available at https://github.com/GBerardi/ZS-SBSR.

1. Introduction

As image vision algorithms rapidly advance, we see a recent surge of interest in sketch understanding [58, 37, 25, 11, 39, 29] and generation [51, 6]. Sketch is the earliest form of visual communication for humanity as a whole, as well as for each individual. However, for vision algorithms, it poses a number of challenges caused by the diversity of sketching styles, skills, and sketch sparsity. Sketches can be very abstract and visually different from photos. Each sketching scenario results in visually very different renditions. People can easily interpret sketches from very abstract to highly detailed or stylized, but is there an algorithm or model that can reliably handle all styles and scenarios?

Inspired by the success of models trained on large datasets in a range of zero-shot applications in the image domain [34, 42, 35, 41, 28], several works exploit its application in the sketch domain. There are a number of inspiring attempts of adapting CLIP (Contrastive Language-Image Pre-Training) [40] as a perceptual loss for deep model training [17, 47, 51, 50], for model performance evaluation [64], or as a way to alleviate the need for the sketch data during training for a downstream task [44]. However, are the used encoders able to discriminate fine-grained differences within a sketch domain or across sketch and image domains? Several works indicate that these models do not necessarily perform that well in a zero-shot sketch-to-image comparison [12, 43]. The works then either resort to fine-tuning existing models [12, 43] or training from scratch by adding tailored losses or sketch-targeted solutions [29, 46].

With our work, firstly, we aim to shed light on the ability of the popular pretrained models to discriminate individual 3D instances in their multi-modal 2D projections. To achieve this, we evaluate encoders trained with CLIP [40] and via a classification task training on the ImageNet dataset [2]. Namely, we study their performance in matching viewpoints and object identities in sketches and images. Secondly, we investigate alternative fine-tuning strategies, inspired by [5, 22, 65]. In our work, we compare visual prompt learning [22], layer normalization weights learning [16, 43] with a careful fine-tuning of all weights. Thirdly, we show that well-designed fine-tuning on a single shape class can lead to improved performance on other shape classes, sometimes surpassing the accuracy of supervised methods. Importantly, we show that fine-tuning can be done on synthetically-generated sketches for a set of 3D

shapes without the requirement to use freehand sketches. We demonstrate the generalization of our approach to relatively abstract freehand sketches from the AmateurSketch dataset [36]. We refer to this scenario as *fine-tuned but zero-shot*. As a test application, we consider sketch-based 3D shape retrieval. Effectively, we introduce the first sketch-based 3D shape retrieval method with state-of-the-art performance that does not require per-class training or fine-tuning. Our fine-tuning only requires a set of 3D shapes of just one category. It is a reasonable assumption for the 3D shape retrieval task.

Fourthly, we perform a detailed performance analysis of different layers of ViT and ResNet-based encoders pretrained either with CLIP training or with a classification task on the ImageNet dataset. We study how the line width and object scale affect performance and find that similar settings can be considered optimal for ViT and ResNet-based encoders. We note that most works [12, 43, 29] use the activation of the final layer of an encoder. Yael et al. [51] observed that while these features excel at capturing semantic meaning, intermediate layers are more suitable when comparing spatial structures. In our research, we offer an in-depth analysis with regard to our specific problem.

In summary, our key contributions are:

- A comprehensive study of the ability of the popular pretrained encoders to discriminate individual 3D instances in their multi-modal 2D projections;
- Extensive analysis of the similarity estimation performance using various layer features and exploration of the impact of the object's scale;
- Comparison of various fine-tuning strategies on the task of matching sketches in distinctive sketch styles;
- Fine-tuning approach that requires as little as a set of 3D shapes of a single category, and generalizes to freehand sketches and other shape class categories, reaching the performance of fully supervised methods.

2. Related work

2.1. Sketch-based 3D shape retrieval

2.1.1 Category-level and fine-grained retrieval

Most of the works in sketch-based 3D model retrieval [15, 59, 30, 24, 52, 62, 27, 66, 54, 13, 21, 38, 23, 8, 57, 10, 63, 56] focus on the problem of *category level* retrieval: They aim to retrieve any instance of a particular object category. In other words, the retrieval is considered to be successful if, given a sketch of an object, the retrieved top N 3D models belong to the same category.

Only two works [36, 9] addressed fine-grained sketch-based 3D model retrieval in a supervised setting. Qi et al. [36] collected the first dataset of instance-level paired freehand sketches and 3D models, which we also use to test our model. They use triplet loss training [53], classic for retrieval tasks, and represent 3D models using multi-view RGB renderings. The main novelty of their paper lies in learning view attention vectors. In concurrent to our work, Chen et al. [9] train and test on the data by Qi et al. [36]. Unlike [36], they learn to project all sketch views to the same latent representation. The main performance gain is caused by dividing the images into three parts and learning to match the features of each part individually. Unlike both of these works, we represent 3D shapes using NPR (Non-Photorealistic Renderings) [1] rather than RGB images.

To reduce the domain gap between sketch queries and 3D models, some works [33, 32] study fine-grained retrieval from a 3D sketch created while wearing a virtual reality headset. Our model aims for much more accessible inputs that can be created with a computer mouse or on paper.

2.1.2 Multi-view feature aggregation

Like majority of the works on sketch-based 3D model retrieval, we use multi-view shape representation, however many of these works differ in how they aggregate features across viewpoints. Thus, Xie et al. [54] use the Wasserstein barycentric of 3D shapes projections in the CNN feature space to represent 3D shapes. He et al. [21] follow MVCNN [49] and aggregate views features with element-wise maximum operation across the views in the view-pooling layer. Lei et al. [23] proposed a representative view selection module that aims to merge redundant features for similar views. Chen et al. [8] learn multi-view feature scaling vectors which are applied prior to average pooling vector, in order to deal with non-aligned 3D shape collections. Qi et al. [36] learn view attention vectors conditioned on the input sketch, which allow to reduce the domain gap between a sketch and multi-view projections of a 3D shape. Zhao et al. [63] leverages spatial attention [60] to exploit view correlations for more discriminative shape representation. In our work, we focus on learning view features that can be used to find the correct shape identity and view across different sketch styles: e.g. freehand and synthetic (generated using non-photorealistic rendering).

2.2. Multi-modal retrieval

Multi-modal retrieval is not directly related to our work, but two concurrent works [48, 46] are worth mentioning as they rely on encoders pretrained with the CLIP model. They explore CLIP embeddings for retrieval from multi-modal inputs such as 2D sketches or images and text. Sangkloy et al. [46] study image retrieval and focus on fine-tuning CLIP using triplets of synthetic sketches, images, and their captions. They rely on the availability of textual descriptions

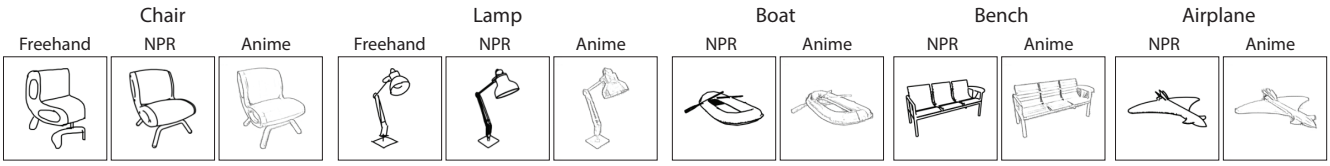

Figure 1. Examples of sketches in our datasets: *freehand* [36], *NPR*s are generated using Blender Freestyle [1], and *Anime* is obtained from an RGB rendering using image translation method [6] in a provided anime style. Please see Sec. 4.2 for details.

matching their images, while we require only the availability of 3D shapes from just one 3D shape class. Similarly to us, Schlachte et al. [48] study zero-shot 3D model retrieval using the CLIP model, but only explore the weighted fusion of CLIP features from multiple inputs for artistic control. Unlike them, we perform an in-depth study analyzing object scale, feature layers, and fine-tuning strategies.

2.3. Sketch datasets

With the advent of sketch datasets [15, 30, 3, 45, 20, 18, 36, 12] the research on sketching thrives. However, it is costly and challenging to collect a dataset of freehand sketches, especially when there is a requirement for instance-level pairing between several domains. The common practice is to let participants study a reference image for a short period of time and then let them draw from memory [15, 3, 45, 12, 46]. This task becomes increasingly challenging when the pairing is required to be between 3D shapes and sketches, as one has to ensure that the viewpoints are representative of those that people are more likely to sketch from [30, 62, 57, 36, 18].

To the best of our knowledge, there is only one dataset [36] of freehand sketches by participants with no prior art experience paired with 3D shapes, that takes views into account and follows the protocol of sketching from memory. The small dataset collected by Zhang et al. [61] for each object contains only one sketch viewpoint, and the viewpoints are non-representative, they are uniformly sampled around 3D shapes. It contains too few examples and is too noisy for retrieval performance evaluation. The recent dataset of paired sketches and 3D models of cars [19] similarly was collected without taking into account viewpoints preferences, and the sketches are drawn directly on top of image views and mostly contain outer shape contours. We, therefore, evaluate our approach on the dataset by Qi et al. [36], as the only existing representative dataset with instance-level pairing between sketches and 3D shapes.

3. Method

In this and the following sections, we present our method for zero-shot sketch-based 3D retrieval. We then provide a comparison to alternative strategies in Sec. 6. To enable sketch-based 3D shape retrieval, we represent 3D shapes using their multi-view projections, commonly used in sketch-based retrieval [54, 21, 23, 8, 63, 30, 62, 57]. To reduce the domain gap, we use NPRs (Non-Photorealistic Renderings) instead of RGB renderings for multi-view 3D shape representation. In the supplemental, we provide a detailed study of the ability of the popular pretrained models to discriminate individual 3D instances in their multi-modal 2D projections: we compare RGB renderings, NPRs, and freehand sketches.

3.1. Zero-shot

Given an encoder, trained on a pretext task, we first compute embeddings of a Query sketch Q and Gallery 3D shape G views using features of a chosen encoder's layer. We then assign the similarity between a sketch and a 3D shape as the maximum cosine similarity between a sketch embedding and individual 3D shape views embeddings. Formally, this can be written as follows:

$$\text{sim}(Q, G) = \max_{v \in views} \text{d}(\text{E}_\ell(Q), \text{E}_\ell(G_v)), \qquad (1)$$

G_v is a 3D shape view, $E_l(\cdot)$ denotes layer ℓ features extracted with the encoder E and d is the cosine similarity[1].

We center and scale 3D objects in query and shape views to fit the same bounding box in both representations.

3.2. Fine-tuned but zero-shot

We propose a contrastive view-based fine-tuning approach that leverages synthetically-generated sketches of single or multiple 3D shape classes. We represent all available 3D shapes with V views, using two different approaches to synthetic sketch generation: geometry-based non-photorealistic rendering [1] and an image-to-image translation method that supports different exemplar styles [6]. We describe the generation of views in Sec. 4.2.2.

Our contrastive loss aims to match identical shape views in these two synthetic sketch styles. Namely, given a batch with B objects, we randomly select one view in two styles for each object. We then compute the pairwise weighted dot product between any two views in two different styles:

$$s_{i,j} := s(G_i^{st1}, G_j^{st2}) := e^t < \text{E}_\ell(G_i^{st1}), \text{E}_\ell(G_j^{st2}) >, \qquad (2)$$

$< \cdot, \cdot >$ is a dot product, G_i^{st} is some view of the i-th object in the mini-batch in one of two styles, and t is a learned parameter.

[1] We experimented with the Mean Squared Error (MSE) distance, taking the minimum MSE distance between a query and shape individual views. We have not observed an obvious advantage of one other another.

Finally, we compute the following contrastive loss:

$$\mathcal{L} = -\frac{1}{2B}\sum_{i=1}^{B}\left(\log\frac{\exp(s_{i,i})}{\sum_{j=1}^{B}\exp(s_{i,j})} + \log\frac{\exp(s_{i,i})}{\sum_{j=1}^{B}\exp(s_{j,i})}\right). \quad (3)$$

Due to our batch construction, this objective trains the network to produce features such that the same views of the same object in different styles have similar embeddings. This objective neither pushes different views of the same object to have identical embeddings nor pushes them apart. Fine-tuning updates the weights of the visual encoder and the temperature parameter t.

Note that Eqs. (1) and (2) can be computed based on the features from any layer and not only the final one. In this case, we only updated the weights up to the layer whose features we use to compute similarity.

4. Implementation details

4.1. Encoder

In the default setting, as an encoder, we use ViT pre-trained with CLIP. We compute similarity using the 6-th layer.

4.2. Datasets

We use two types of datasets: (1) the dataset of freehand sketches by participants without art experience, and (2) the dataset of synthetically generated sketches in two styles for 11 classes of the ShapeNet 3D shape dataset [7]. Different styles are shown in Fig. 1, and described in detail below.

4.2.1 Freehand sketches

We use the dataset of freehand sketches by Qi et al. [36] to evaluate the models' performance. This dataset contains sketches for two shape categories: *chair* and *lamp*, representing 1,005 and 555 3D shapes from the respective class of the ShapeNet dataset [7]. The sketches are created by participants without any prior sketching experience, and fit well the scenario we are targeting. The sketches are drawn from a viewpoint with a zenith angle of around 20 degrees. For each category three settings of azimuth angles are used. For the *chair* category, they are $0°$, $30°$ and $75°$, while for the *lamp* category they are $0°$, $45°$ and $90°$. These particular viewpoints are selected as the most likely viewpoints based on sketching literature [14, 4, 55, 31, 26, 18] and pilot studies conducted by Qi et al. [36].

The dataset provides a split to training, validation, and test data. To facilitate comparison with previous supervised work, we only use a test set of sketches to test models. The test set consists of 201 and 111 sketch-3D shape quadruplets for the *chair* and *lamp* categories, respectively. We do not use any freehand sketches for training. Prior to testing, we re-scale and center objects' projections in freehand sketches to occupy the central image area of 129×129.

4.2.2 Synthetic sketches

Additionally, we create a dataset of synthetic sketches in two styles, representing 3D shapes from the ShapeNetCore 3D shape dataset [7]. We select 11 of the 13 ShapeNetCore classes, discarding two classes with the lowest number of 3D shapes.

Views and camera setting We follow camera settings used to collect sketches in the dataset of freehand sketches [36]. In particular, we use for all shape classes viewpoints with the following azimuth angles: $0°, 30°, 45°, 75°$ and $90°$. We set the camera distance to an object to 2.5 and the camera zenith angle to $20°$. The size of rendered views is 224×224 unless specified otherwise.

NPR (style-1) We render views using silhouettes and creases lines in Blender Freestyle [1]. We render views as SVGs and then re-scale and center objects' projections in freehand sketches to occupy the central image area of 129×129. Prior to rasterization, we assign each stroke a uniform stroke width of $2.2px$.

Anime (style-2) We obtain the second synthetic sketch style by first rendering RGB images of 3D shapes using Blender Freestyle with the same camera settings as for the first NPR synthetic style. We then re-scale and center objects' projections in RGB renderings to occupy the central image area of 129×129. Finally, we generate synthetic sketches in the second style, using the pre-trained network [6] in *anime* style.

4.3. Data usage

4.3.1 3D Shape representation

We represent a 3D shape with its multi-view NPR projections: We use the set of $0°, 30°, 45°, 75°$ and $90°$, common for chair and lamp category sketches from the *AmateurSketch* dataset [36], to represent 3D models.

4.3.2 Fine-tuning

We split 3D shapes in each class into training (70 %), validation (15%), and test (15%) sets. We set the learning rate to 10^{-7}, the batch size to 64, and use the Adam optimizer. *We note that the choice of the learning rate is critical, as larger learning rates will result in overfitting harming the performance.*

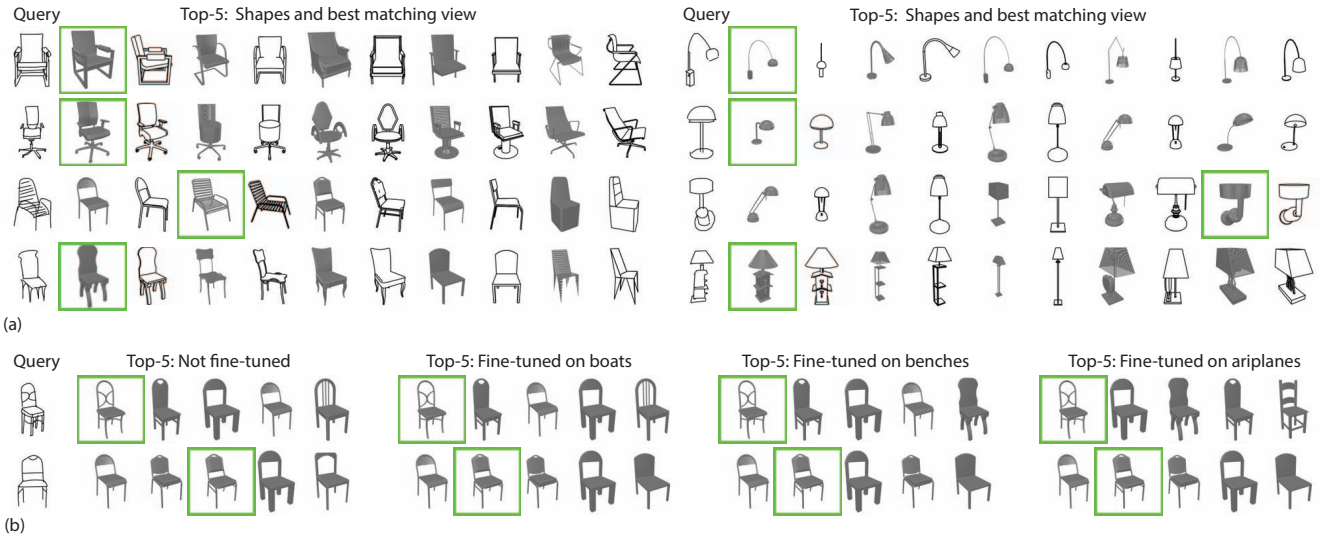

Figure 2. Qualitative results obtained with features of the 6th layer of the ViT encoder pretrained on CLIP and fine-tuned using our method. The queries are freehand sketches from the AmateurSketch dataset [36]. Green boxes highlight groundtruth shapes. (a) shows retrieved shapes and the best matching view according to Eq. (1); (b) shows retrieval results without our fine-tuning and with fine-tuning on each of the free classes: boats, benches, and airplanes.

Data augmentation While fine-tuning, we augment synthetic sketches in the anime style with random affine transformation, translation, rotation, and scaling operations. This augmentation simulates the type of distortions that we can encounter in freehand query sketches. Even if we scale and center objects in freehand sketches in processing, sketches might contain small rotations. The translation moves an image along the x and y axes for a random number of pixels in the range $[-10\%, +10\%]$ of the image size. The rotation is sampled between $[-10, +10]$ degrees. Finally, we increase or decrease the object's bounding box size by a random value in the range [-10%, +10%] of the image size.

Checkpoint selection We train our fine-tuning model for 500 epochs. At test time we use the weights from the last epoch.

4.3.3 Test time

We test our retrieval models on the freehand sketches. We also test on synthetic sketches to show generalization to other shape classes. By default, we use sketches in the anime style with azimuth angles set to $0°$, $45°$, and $90°$ as queries. To facilitate comparisons with performance on freehand sketches, for each shape class we form the final test sets by randomly selecting just 200 3D shapes non-overlapping with training or validation sets.

5. Results

To evaluate retrieval accuracy, we use the standard for retrieval tasks Top-1 (Acc@1), and Top-5 (Acc@5) accuracy measures. They evaluate the percentage of times the ground-truth is returned among the top 1 and top 5 ranked retrieval results, respectively.

	Chairs		Lamps		Avg. score. Anime → NPR	
Method	acc@1	acc@5	acc@1	acc@5	acc@1	acc@5
[36]	56.72	87.06	57.66	87.39	n.a.	n.a.
[9]	**83.08**	**97.01**	78.08	**95.50**	n.a.	n.a.
ViT-CLIP L-6	74.79	89.39	73.27	89.49	82.48	93.82
ViT-CLIP* L-6	77.11	92.32	**78.38**	92.39	**87.84**	**97.13**

Table 1. Our zero-shot results versus supervised methods: [36] and concurrent to our work [9]. Neither [36] nor [9] provide code, therefore, we use the numbers provided in their respective papers. For the *ViT-CLIP* methods, we center and scale objects in reference and query views according to optimal scaling. *L-6* indicates the layer whose features we use for similarity computation. *ViT-CLIP** represent the average performance results of three individual fine-tuning experiments on the three classes: *boat*, *airplane*, and *bench*, using synthetic sketches. *Avg. score. anime* represents average results on 11 classes where queries are in anime style and gallery shapes are represented using muti-view NPR projections. The boldface font highlights the best results, and the underscore highlights the second-best results.

Tab. 1 (*ViT-CLIP L-6*) shows the retrieval accuracy of our zero-shot setting on the freehand sketches and synthetic sketches in anime style. We then perform three individual fine-tuning experiments on three classes: *boat*, *airplane*, and *bench*, using synthetic sketches, and report an average accuracy over the three experiments in Tab. 1 (*ViT-CLIP* L-6*). We compare with two supervised works by Qi et al. [36] and Chen et al.[9] who train on one class at a time and use freehand sketches from [36]. As no code is available for the competitors, we report the numbers from their re-

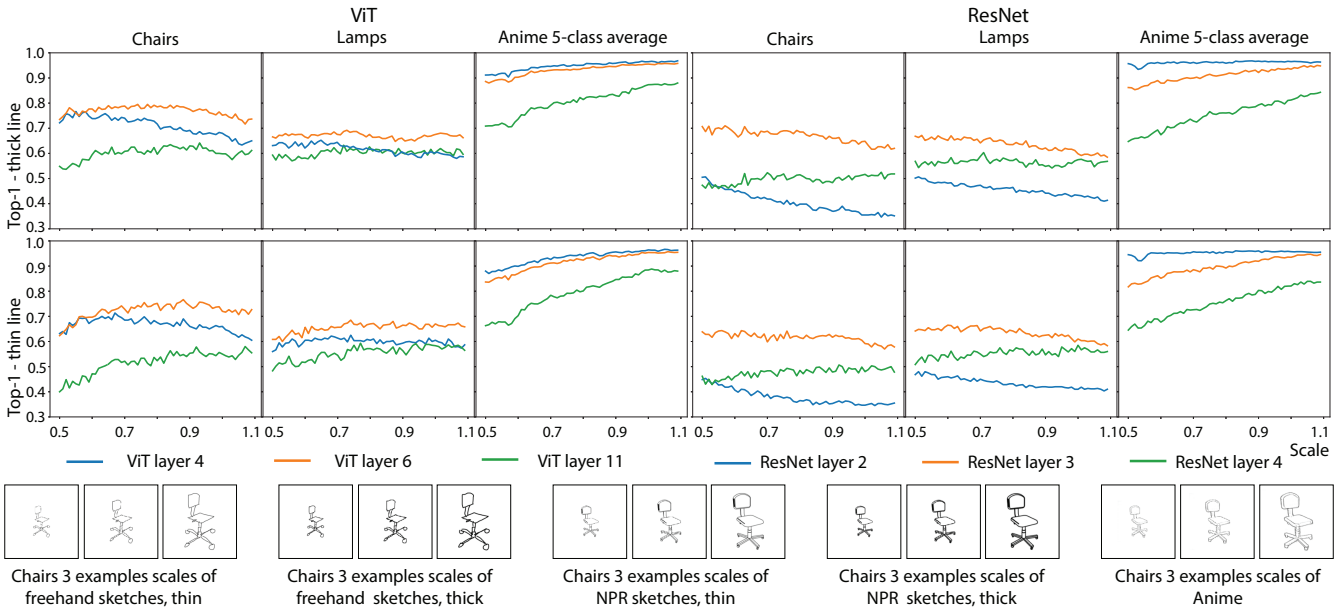

Figure 3. Role of the object projection area, line width, and feature layer in the ability to predict similarity between views in different domains. Please see Sec. 6.2 for the details.

spective papers.

Our zero-shot models are able to achieve remarkable results, surpassing [36] in all respects. This shows the generalization ability of our method to different styles and diverse shape classes. Compared to concurrent to our work [9], we can see that accuracy of our zero-shot method can be further improved. Note that on the *lamp* category we outperform the concurrent supervised method in top-1 accuracy, while our method is zero-shot! Our *fine-tuning but zero-shot* improves top-1/5 retrieval accuracy on average by 4.3 and 3 points, respectively, over zero-shot performance. The visual results for our method are shown in Fig. 2.

6. Ablation studies

6.1. Choice of an encoder and pretext task

In this section, we compare (1) two types of encoders: ViT and ResNet, and (2) two types of pretext tasks: CLIP training and classification task training on the ImageNet dataset [2]. All the models share the same input image size of 224×224 except for ViT pre-trained on ImageNet. In the latter case, the image size is 384×384, and we rescale and center objects' projections in freehand and synthetic sketches to occupy the central area of 291×291.

Tab. 2 shows the comparison in retrieval accuracy in a zero-shot setting without fine-tuning for several best-performing layers. It shows that ViT encoder pretrained with CLIP model achieves the best results, and justifies the use of it as our default in most of the experiments. Interstingly, training on the ImageNet for the RestNet encoder gives slightly better performance than training with the CLIP model.

	Chairs → NPR		Lamps → NPR		Avg.score Anime → NPR	
Method	acc@1	acc@5	acc@1	acc@5	acc@1	acc@5
ViT CLIP L-6	**74.79**	**89.39**	**73.27**	**89.49**	82.48	93.82
ViT ImageNet L-5	63.35	82.92	66.67	87.99	81.77	94.14
ResNet CLIP L-3	65.17	84.08	69.97	87.99	76.97	90.36
ResNet ImageNet L-3	66.50	82.09	65.77	88.89	**83.82**	**95.29**

Table 2. Comparison of ResNet and ViT encoders trained either with CLIP model or classification task on the ImageNet dataset. See Sec. 6.1 for the details. In all cases, objects in sketches are optimally scaled and centered.

6.2. Object projection area, line width and feature layer

In our preliminary experiments, we observed that scaling sketches and 3D model projections to fit the same bounding box area results in improved retrieval accuracy (See Tab. 3). These findings also align with the experiments in [9]. We then are interested in how sensitive different backbones (ViT and ResNet) are to (1) the scale of the object in the image plane; (2) the line width, and (3) how the accuracy of feature similarities according to features from different layers varies with object scale.

	Chair		Lamp	
	acc@1	acc@5	acc@1	acc@5
ViT-CLIP L-6 w/o alignment	69.82	86.40	67.27	87.69
ViT-CLIP L-6	**74.93**	**89.39**	**73.27**	**89.49**

Table 3. Comparison of the zero-shot retrieval performance of the ViT encoder trained with CLIP on the datasets without objects centering and rescaling vs. on the datasets where objects in sketches are centered and scaled as described in Sec. 4.3.

	Chairs → NPR		Chairs → Anime		Lamps → NPR		Lamps → Anime		Avg.score Anime → NPR		Avg.score NPR → Anime	
Method	acc@1	acc@5	acc@1	acc@5	acc@1	acc@5	acc@1	acc@5	acc@1	acc@5	acc@1	acc@5
ViT CLIP L-6	**74.79**	**89.39**	63.35	80.93	**73.27**	**89.49**	62.16	82.88	**82.48**	**93.82**	77.03	90.94

Table 4. NPR vs. Anime 3D shape representation. In the notation $X \rightarrow Y$, X is a query domain and Y is a 3D shape representation domain.

6.2.1 Object bounding box size & line width

We first obtain an initial common bounding box size (170×170) by taking the smallest square bounding box that fully encompasses objects in all sketches in the dataset of freehand sketches in the form they are provided by Qi et al. [37]. We rescale and center all object projections in all freehand and synthetic SVG sketch versions to this bounding box. We then use two settings of line width: thick (set to 2.2px) and thin (set to 1.0px) that we assign to all strokes (Fig. 3: 1st vs. 2nd rows), and rasterize the sketches. We evaluate varying scaling of the original 170×170 bounding box size, by rescaling raster images so that the object projections are within varying bounding box sizes from 85×85 to 187×187 with 60 uniform steps (Fig. 3: scale in horizontal axes).

First, we observe that among the two considered line settings, thicker lines result in better retrieval accuracy. For freehand sketches, scaling between 0.7 and 0.8 that represents bounding boxes with sizes 119×119 and 136×136, respectively, result on average in top performance across encoder architectures and feature layers. For synthetic sketches, the large the object in a sketch is, the more accurate is the prediction. We believe that is caused by two factors (1) the great degree of spatial alignment between two types of synthetic sketches and (2) the presence of very thin lines in anime style sketches at smaller object scales.

6.2.2 Feature layers

We study how retrieval accuracy varies when feature similarity is computed on features from different layers for different object projections bounding box sizes. In Fig. 3, we plot accuracy for similarity in Eq. (1) computed with features from layers 4, 6 and 11 of the ViT encoder, and layers 2, 3 and 4 of the ResNet, both trained with CLIP (Fig. 3: first three columns vs. last three columns).

Fig. 3 shows that on the two categories of the dataset of freehand sketches features from mid-layers – ViT layer 6 and ResNet layer 3 – result in the best performance for both architectures. On synthetic sketches, slightly better performance is achieved with features from lower layers: layer 4 of ViT and layer 2 of ResNet. It can be also observed that for lower layers (ViT layer 4 and ResNet layer 2) the performance is increasing as object area is decreasing, while for higher layers (ViT layer 11 and ResNet layer 4) the behavior is opposite. The intuition is that the features from higher layers are better suited for more abstract sketches and extracting sketch semantic meaning, while lower layers focus more on spatial details. Indeed, as NPR and anime synthetic sketches are spatially more similar than NPR and freehand sketches, the lower layers result in better performance when anime sketch is used as a query.

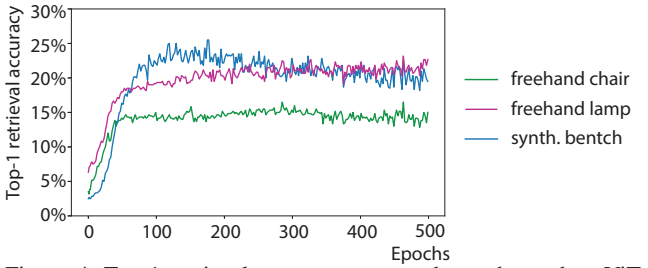

Figure 4. Top-1 retrieval accuracy vs. epoch number, when ViT encoder is trained from scratch as described in Sec. 6.3 on synthetic sketches of the *bench* class.

6.3. Fine-tuning vs. training from scratch

To show the advantage of fine-tuning in the zero-shot scenario, we compare our approach with training from scratch on a single *bench* class. We use our fine-tuning training objective to train ViT encoder from scratch. Features from the 6th layer are used. Therefore, we keep only the network part up to the 6th layer including. Since we train from scratch, we set a larger starting learning rate of 10^{-5} for the Adam optimizer.

Fig. 4 shows that training from scratch is prone to overfitting: It results in a drop in Top-1 retrieval accuracy on the test set of the *bench* class starting from the 150th epoch. After the 80th epoch, the accuracy improves very slowly for the *lamp* class and does not improve anymore for the *chair* class. Note that during training the contrastive loss Eq. (3) decreases over all 500 epochs. Moreover, for all three considered classes, the retrieval accuracy is quite low: it is below 30%, while the Top-1 retrieval accuracy of our approach surpasses 70%. The overfitting result is similar to observations in [9] when they use only one branch to represent both sketch and image modalities.

6.3.1 3D shape representation: NPR or anime

As we have two synthetic sketch styles (Sec. 4.2.2), we evaluate our choice of representing 3D shapes with NPR views against views in the anime style. Tab. 4 shows a clear advantage of representing 3D shapes using NPR renderings in

both considered cases: when query sketches are freehand sketches or synthetic sketches.

6.3.2 Feature aggregation strategy

We evaluate our similarity computation strategy between a query sketch and 3D shape, given by Eq. (1), against an alternative strategy of computing the cosine similarity between the query sketch embedding and the average of 3D shape views embeddings:

$$\mathrm{sim}(Q, G) = \mathrm{d}\left(\mathrm{E}_\ell(Q), \frac{1}{V}\sum_{v \in views} \mathrm{E}_\ell(G_v)\right), \quad (4)$$

where, as in Eq. (1), Q and G denote a query sketch and a gallery shape; G_v is a 3D shape view, V is the number of views for an object (5 in our case), $E_\ell(\cdot)$ denotes ℓ-th layer features of the encoder E, and d stands for the cosine similarity.

	Chair		Lamp	
	acc@1	acc@5	acc@1	acc@5
Avg. - ViT-CLIP L-6	70.32	**89.72**	63.06	78.08
Max. - ViT-CLIP L-6	**74.93**	89.39	**73.27**	**89.49**
Avg. - ViT-CLIP* L-6	74.72	90.71	66.97	82.28
Max. - ViT-CLIP* L-6	**77.11**	**92.32**	**78.38**	**92.39**

Table 5. Comparison of feature selection strategies on the test set of the freehand sketch dataset.

Tab. 5 shows the comparison of the two similarity computations strategies for the ViT encoder trained with the CLIP model in zero-shot or our fine-tuned setting. It can be seen that in all settings our strategy is superior to this alternative strategy, with a gap of almost 3 points in Top-1 retrieval accuracy on *chairs*, and of more than 10 points in both Top-1 and Top-5 on *lamps*.

6.3.3 Fine-tuning strategies

We compare our fine-tuning strategy with two alternative strategies of fine-tuning only the weights of layer normalization layers [16] and Visual Prompt Tuning (VPT) [22], which we refer to as *ViT-CLIP LayerNorm* and *ViT-CLIP VPT*, respectively. We train the two additional strategies under the same conditions and loss as our fine-tuning strategy but set a higher learning rate of 10^{-5}.

The VPT approach consists in adding learnable tokens to the attention layers of the feature extractor. During training, all the original network weights are fixed and only the new tokens are updated. We use the deep prompt setting and add 5 additional tokens on the first 6 layers of ViT. As we observe that with VPT the performance on the validation set of the freehand sketch dataset starts to decrease after 100 epochs, we stop the training at 100th epoch and use the last checkpoint.

	Chair		Lamp	
	acc@1	acc@5	acc@1	acc@5
ViT-CLIP L-6	74.93	89.39	73.27	89.49
ViT-CLIP LayerNorm L-6	74.96	90.71	73.87	91.59
ViT-CLIP VPT L-6	73.80	90.22	73.57	90.99
ViT-CLIP* L-6 (Ours)	**77.17**	**92.32**	**78.38**	**92.39**

Table 6. Comparison with the alternative fine-tuning strategies on the test set of the dataset of freehand sketches.

Tab. 6 shows that both, the layer normalization layer tuning (ViT CLIP LayerNorm L-6) and VPT (ViT CLIP VPT L-6), allow for increased performance compared to the zero-shot ViT (ViT-CLIP L-6) without fine-tuning. However, our fine-tuning strategy (ViT-CLIP* L-6) achieves the best performance.

7. Limitations and Future Work

While the ViT transformer was proven to be a very efficient encoder for an image domain, it might be not the best for sparse sketches. In the case of sketches, non-overlapping patches can contain too little meaningful information and alternative encoder designs should be considered. One such design was recently proposed by Lin et al. [29]. Another direction to explore is to combine vector and raster sketch encoders. To achieve zero-shot performance, the models with tailored encoders then can be trained in a multi-modal setting.

Next, our fine-tuning strategy can be expanded to include multi-modal training. For example, if textual descriptions of 3D shapes are available, they can be seamlessly integrated into our fine-tuning process.

8. Conclusion

In this work, we introduced an effective zero-shot sketch-based 3D shape retrieval method. We demonstrated how to efficiently adapt models pretrained on different pretext tasks, like CLIP, to the studied problem. We show that it is possible to fine-tune a model leveraging only synthetic sketches of a single shape category and demonstrated generalization to freehand abstract sketches of other shape categories. We also showed that performance is similar independently of the choice of a shape category for fine-tuning. We bring insights into the role of object scale in the image plane and provide recommendations taking into account query abstraction. We compare the performance of two popular image encoders ViT and ResNet and show that the same object scale is beneficial for the two encoders under consideration regardless of the pretext task used. We also carefully study the role of object scale in the image plane and provide recommendations taking into account query abstraction. We believe that our work provides valuable information for methods aimed at assessing the perceptual similarity between sketches in different styles.

References

[1] Blender Freestyle: https://docs.blender.org/manual/en/latest/render/freestyle/introduction.html. 2, 3, 4

[2] ImageNet: https://www.image-net.org/. 1, 6

[3] Stanislaw Antol, C Lawrence Zitnick, and Devi Parikh. Zero-shot learning via visual abstraction. In *European conference on computer vision*. Springer, 2014. 3

[4] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. Ilovesketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 151–160, 2008. 4

[5] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv e-prints*, pages arXiv–2203, 2022. 1

[6] Caroline Chan, Frédo Durand, and Phillip Isola. Learning to generate line drawings that convey geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1, 3, 4

[7] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. arXiv:1512.03012, 2015. 4

[8] Jiaxin Chen, Jie Qin, Li Liu, Fan Zhu, Fumin Shen, Jin Xie, and Ling Shao. Deep sketch-shape hashing with segmented 3d stochastic viewing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2, 3

[9] Xu Chen, Zheng Zhong, and Dongbo Zhou. Spatially aligned sketch-based fine-grained 3d shape retrieval. *Neural Computing and Applications*, 2023. 2, 5, 6, 7

[10] Zhixiang Chen, Haifeng Zhao, Yan Zhang, Guozi Sun, and Tianjian Wu. Self-supervised learning for sketch-based 3d shape retrieval. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer, 2022. 2

[11] Pinaki Nath Chowdhury, Ayan Kumar Bhunia, Aneeshan Sain, Subhadeep Koley, Tao Xiang, and Yi-Zhe Song. What can human sketches do for object detection? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15083–15094, 2023. 1

[12] Pinaki Nath Chowdhury, Aneeshan Sain, Yulia Gryaditskaya, Ayan Kumar Bhunia, Tao Xiang, and Yi-Zhe Song. FS-COCO: Towards understanding of freehand sketches of common objects in context. In *ECCV*, 2022. 1, 2, 3

[13] Guoxian Dai, Jin Xie, Fan Zhu, and Yi Fang. Deep correlated metric learning for sketch-based 3D shape retrieval. In *Proc. Associat. Adv. Artif. Intell.*, 2017. 2

[14] Koos Eissen and Roselien Steur. Sketching: the basics; the prequel to sketching: drawing techniques for product designers. *BIS, Amsterdam. OCLC*, 756275344, 2011. 4

[15] Mathias Eitz, Ronald Richter, Tamy Boubekeur, Kristian Hildebrand, and Marc Alexa. Sketch-based shape retrieval. *ACM Trans. Graph.*, 31(4), 2012. 2, 3

[16] Jonathan Frankle, David J Schwab, and Ari S Morcos. Training batchnorm and only batchnorm: On the expressive power of random features in cnns. *arXiv e-prints*, 2020. 1, 8

[17] Kevin Frans, Lisa Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *Advances in Neural Information Processing Systems*, 35, 2022. 1

[18] Yulia Gryaditskaya, Mark Sypesteyn, Jan Willem Hoftijzer, Sylvia C Pont, Frédo Durand, and Adrien Bousseau. Opensketch: a richly-annotated dataset of product design sketches. *ACM Trans. Graph.*, 38(6), 2019. 3, 4

[19] Benoit Guillard, Edoardo Remelli, Pierre Yvernay, and Pascal Fua. Sketch2mesh: Reconstructing and editing 3d shapes from sketches. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 3

[20] David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018. 3

[21] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-Center Loss for Multi-View 3D Object Retrieval. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018. 2, 3

[22] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*. Springer, 2022. 1, 8

[23] Yinjie Lei, Ziqin Zhou, Pingping Zhang, Yulan Guo, Zijun Ma, and Lingqiao Liu. Deep point-to-subspace metric learning for sketch-based 3d shape retrieval. *Pattern Recognition*, 96, 2019. 2, 3

[24] Bo Li, Yijuan Lu, Chunyuan Li, Afzal Godil, Tobias Schreck, Masaki Aono, Martin Burtscher, Hongbo Fu, Takahiko Furuya, Henry Johan, et al. SHREC'14 track: Extended large scale sketch-based 3D shape retrieval. In *3D Object Retrie.*, volume 2014, 2014. 2

[25] Changjian Li, Hao Pan, Adrien Bousseau, and Niloy J Mitra. Free2cad: Parsing freehand drawings into cad commands. *ACM Transactions on Graphics (TOG)*, 41(4):1–16, 2022. 1

[26] Changjian Li, Hao Pan, Yang Liu, Xin Tong, Alla Sheffer, and Wenping Wang. Bendsketch: Modeling freeform surfaces through 2d sketching. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 4

[27] Lei Li, Zhe Huang, Changqing Zou, Chiew-Lan Tai, Rynson WH Lau, Hao Zhang, Ping Tan, and Hongbo Fu. Model-driven sketch reconstruction with structure-oriented retrieval. In *SIGGRAPH ASIA 2016 Technical Briefs*. 2016. 2

[28] Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. Clip surgery for better explainability with enhancement in open-vocabulary tasks. *arXiv preprint arXiv:2304.05653*, 2023. 1

[29] Fengyin Lin, Mingkang Li, Da Li, Timothy Hospedales, Yi-Zhe Song, and Yonggang Qi. Zero-shot everything sketch-based image retrieval, and in explainable style. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2, 8

[30] Dawei Lu, Huadong Ma, and Huiyuan Fu. Efficient sketch-based 3d shape retrieval via view selection. In *Pacific-Rim Conference on Multimedia*. Springer, 2013. 2, 3

[31] Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang. 3d shape reconstruction from sketches via multi-view convolutional networks. In *Proc. 3D Vision*, pages 67–77, 2017. 4

[32] Ling Luo, Yulia Gryaditskaya, Tao Xiang, and Yi-Zhe Song. Structure-aware 3d vr sketch to 3d shape retrieval. *arXiv preprint arXiv:2209.09043*, 2022. 2

[33] Ling Luo, Yulia Gryaditskaya, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. Fine-grained vr sketching: Dataset and insights. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021. 2

[34] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13492–13502, 2022. 1

[35] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022. 1

[36] Anran Qi, Yulia Gryaditskaya, Jifei Song, Yongxin Yang, Yonggang Qi, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Toward fine-grained sketch-based 3d shape retrieval. *IEEE Transactions on Image Processing*, 30, 2021. 2, 3, 4, 5, 6

[37] Anran Qi, Yulia Gryaditskaya, Tao Xiang, and Yi-Zhe Song. One sketch for all: One-shot personalized sketch segmentation. *IEEE transactions on image processing*, 31:2673–2682, 2022. 1, 7

[38] A. Qi, Y. Song, and T. Xiang. Semantic Embedding for Sketch-Based 3D Shape Retrieval. In *Proc. Brit. Mach. Vis. Conf.*, 2018. 2

[39] Zhiyu Qu, Yulia Gryaditskaya, Ke Li, Kaiyue Pang, Tao Xiang, and Yi-Zhe Song. Sketchxai: A first look at explainability for human sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23327–23337, 2023. 1

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PMLR, 2021. 1

[41] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *arXiv preprint arXiv:2302.01721*, 2023. 1

[42] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1

[43] Aneeshan Sain, Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Subhadeep Koley, Tao Xiang, and Yi-Zhe Song. Clip for all things zero-shot sketch-based image retrieval, fine-grained or not. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2

[44] Aditya Sanghi, Pradeep Kumar Jayaraman, Arianna Rampini, Joseph Lambourne, Hooman Shayani, Evan Atherton, and Saeid Asgari Taghanaki. Sketch-a-shape: Zero-shot sketch-to-3d shape generation. *arXiv preprint arXiv:2307.03869*, 2023. 1

[45] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Trans. Graph.*, 35(4), 2016. 3

[46] Patsorn Sangkloy, Wittawat Jitkrittum, Diyi Yang, and James Hays. A sketch is worth a thousand words: Image retrieval with text and sketch. In *European Conference on Computer Vision*. Springer, 2022. 1, 2, 3

[47] Peter Schaldenbrand, Zhixuan Liu, and Jean Oh. Styleclipdraw: Coupling content and style in text-to-drawing synthesis. *arXiv preprint arXiv:2111.03133*, 2021. 1

[48] Kristofer Schlachter, Benjamin Ahlbrand, Zhu Wang, Ken Perlin, and Valerio Ortenzi. Zero-shot multi-modal artist-controlled retrieval and exploration of 3d object sets. In *SIGGRAPH Asia 2022 Technical Communications*. 2022. 2, 3

[49] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, 2015. 2

[50] Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. CLIPascene: Scene Sketching with Different Types and Levels of Abstraction. *arXiv preprint arXiv:2211.17256*, 2022. 1

[51] Yael Vinker, Ehsan Pajouheshgar, Jessica Y. Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. Clipasso: Semantically-aware object sketching. *ACM Trans. Graph.*, 41(4), 2022. 1, 2

[52] Fang Wang, Le Kang, and Yi Li. Sketch-based 3D shape retrieval using convolutional neural networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015. 2

[53] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proc. of IEEE/CVF CVPR*, 2014. 2

[54] Jin Xie, Guoxian Dai, Fan Zhu, and Yi Fang. Learning barycentric representations of 3D shapes for sketch-based 3d shape retrieval. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017. 2, 3

[55] Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James McCrae, and Karan Singh. True2form: 3d curve networks from 2d sketches via selective regularization. *ACM Transactions on Graphics*, 33(4), 2014. 4

[56] Rui Xu, Zongyan Han, Le Hui, Jianjun Qian, and Jin Xie. Domain disentangled generative adversarial network for zero-shot sketch-based 3d shape retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2022. 2

[57] Yongzhe Xu, Jiangchuan Hu, Kanoksak Wattanachote, Kun Zeng, and YongYi Gong. Sketch-based shape retrieval via best view selection and a cross-domain similarity measure. *IEEE Transactions on Multimedia*, 22(11), 2020. 2, 3

[58] Lumin Yang, Jiajie Zhuang, Hongbo Fu, Xiangzhi Wei, Kun Zhou, and Youyi Zheng. Sketchgnn: Semantic sketch segmentation with graph neural networks. *ACM Transactions on Graphics (TOG)*, 40(3):1–13, 2021. 1

[59] Sang Min Yoon, Maximilian Scherer, Tobias Schreck, and Arjan Kuijper. Sketch-based 3D model retrieval using diffusion tensor fields of suggestive contours. In *Proc. IEEE Trans. Multimedia*, 2010. 2

[60] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *CoRR*, 2018. 2

[61] Song-Hai Zhang, Yuan-Chen Guo, and Qing-Wen Gu. Sketch2model: View-aware 3d modeling from single free-hand sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 3

[62] Long Zhao, Shuang Liang, Jinyuan Jia, and Yichen Wei. Learning best views of 3d shapes from sketch contour. *The Visual Computer*, 31(6), 2015. 2, 3

[63] Yue Zhao, Qi Liang, Ruixin Ma, Weizhi Nie, and Yuting Su. Jfln: Joint feature learning network for 2d sketch based 3d shape retrieval. *Journal of Visual Communication and Image Representation*, 2022. 2, 3

[64] Xin-Yang Zheng, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Heung-Yeung Shum. Locally Attentional SDF Diffusion for Controllable 3D Shape Generation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 2023. 1

[65] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 1

[66] Fan Zhu, Jin Xie, and Yi Fang. Learning cross-domain neural networks for sketch-based 3D shape retrieval. In *Proc. Associat. Adv. Artif. Intell.*, 2016. 2