SketchAA: Abstract Representation for Abstract Sketches

Lan Yang[1,2] Kaiyue Pang[2] Honggang Zhang[1] Yi-Zhe Song[2]
[1] PRIS, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, China
[2] SketchX, CVSSP, University of Surrey, United Kingdom
{ylan, zhhg}@bupt.edu.cn, {kaiyue.pang, y.song}@surrey.ac.uk

Abstract

What makes free-hand sketches appealing for humans lies with its capability as a universal tool to depict the visual world. Such flexibility at human ease, however, introduces abstract renderings that pose unique challenges to computer vision models. In this paper, we propose a purpose-made sketch representation for human sketches. The key intuition is that such representation should be abstract at design, so to accommodate the abstract nature of sketches. This is achieved by interpreting sketch abstraction on two levels: appearance and structure. We abstract sketch structure as a pre-defined coarse-to-fine visual block hierarchy, and average visual features within each block to model appearance abstraction. We then discuss three general strategies on how to exploit feature synergy across different levels of this abstraction hierarchy. The superiority of explicitly abstracting sketch representation is empirically validated on a number of sketch analysis tasks, including sketch recognition, fine-grained sketch-based image retrieval, and generative sketch healing. Our simple design not only yields strong results on all said tasks, but also offers intuitive feature granularity control to tailor for various downstream tasks. Code will be made publicly available.

1. Introduction

Sketches are different to photos. They exhibit a severe lack of visual cues, often made up of just a few coarse strokes other than full of color and texture. The remarkable thing is however despite its abstract nature, humans are still acute to recognizing sketches somewhat equally well to that for a full-blown color photo – one only needs to observe a smiley face to tell the emotion other than seeing a true photo. It is precisely this abstract nature that triggered much of the research on human sketches [13, 42, 56, 19, 16, 58]. With the proliferation of touchscreen devices, this interest has also resulted in a series of practical applications, from sketch-based image retrieval [55, 39, 57, 12, 4], to sketch to photo synthesis [59, 40, 7, 17, 6].

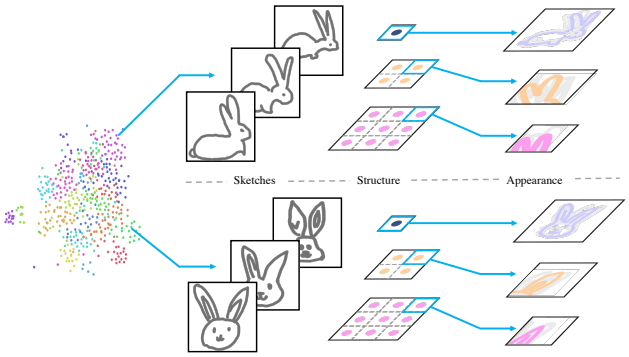

Figure 1: We represent human sketch based on the insight of sketch abstraction as a process happening on two fronts: appearance and structure.

At the very core of all such advancements is learning a feature representation that is most suitable for sketch data. The early days saw the use of HoG [9] descriptors re-purposed for sketch [14, 13, 37]. Coming to the deep era, sketch representation learning mainly takes two streams: (i) CNNs that treat sketches as pixel-maps [56, 12, 36], and (ii) RNNs that utilize the temporal stroke-by-stroke nature of sketches [19, 38, 30]. They each have its pros and cons, though what none of them did was accommodating for the abstract nature of sketches *at design*. This is also evident in that similar to how HoG was re-purposed for sketch, such CNN and RNN-based approaches were also mainly small deviations from their original photo forms [56, 19].

In this paper, we set out to change that. We aim to design a sketch feature learning scheme that directly tackles the abstract nature of sketches. Our key intuition is therefore the actual feature learning should resemble that of an abstraction process. We envisage this abstraction process to happen on two fronts – appearance and structure. We take the appearance abstraction process as just feature averaging within a local visual block, and abstract sketch structure as a hierarchy of multi-granularity grid blocks (see Figure 1).

Such representation of sketch data has a nice interpre-

tation. By abstracting appearance into a visual mean, feature learning is better regularized beyond individual drawing variations. Structural abstraction can then happen by traversing a coarse-to-fine hierarchy of these aggregated features. Figure 1 illustrates the feature embedding of the rabbit category learned by our model. By projecting similar features into their respective visual instances, one can see how our proposed representation groups thematically similar sketches as a clear sign of appearance abstraction taking place (e.g., over rabbit pose), and how different abstraction level focuses on summarizing different visual patterns (e.g., rabbit head vs. ears).

More specifically, at each level along the abstraction hierarchy, we divide a sketch into a pre-defined number of grid blocks. The sketch visual feature for each block is then computed as the mean of a collection of visual patches centered around sampled stroke points (see Figure 2). Naively aggregating appearance features across this hierarchy however does not work – we need to encourage information exchange across granularity levels to fully benefit from the said abstraction process. For that, we discuss three general aggregation strategies and find that hierarchical modeling with graph learning works best. Another intriguing property of our sketch representation is we can now control feature granularity and tailor model behaviors based on the target task. By increasing levels in the hierarchy, finer-grained visual feature representation can be achieved (Section 5). We show our method, albeit being simple, achieves state-of-the-art results on the task of sketch recognition. It can also be plug-and-play as a competitive sketch-specific feature extractor for a range of different applications such as sketch-based image retrieval and sketch healing.

The contributions of this work are as follows: (i) we provide a new method for representing human sketch data via explicit appearance and structure abstraction. (ii) a solution is introduced to foster the feature synergy in the multi-granularity modeling of sketch structure. (iii) the efficacy of our sketch-specific abstract representation has been demonstrated on diverse sketch analysis tasks, including sketch recognition, fine-grained sketch-based image retrieval and generative sketch healing.

2. Related Work

Our related works fall in the general field of representation learning and modeling for human sketches. We summarize the most relevant three categories of works here.

Vector vs. raster Being a distinctive modality to photo, sketch is the result of human creation through a temporally iterative process. However, the way how many previous studies [27, 42, 29, 37, 55, 23, 35, 31] treat sketch comes no difference as that of an image – they rasterize a vector sketch and feed it into a contemporary deep convolutional neural network (CNN) for visual learning. The reason is of course partly due to the convincingly representational power of CNN that has dominated various vision tasks. What's more important is the lack of a general-purpose alternative to the ImageNet pre-training model in the temporal domain that ensures competitive performance under sketch data scarcity. Thanks to the availability of large-scale sketch dataset [19], there is a very recent resurgence of research interest of representing sketches in its original vector format [33, 32, 30, 10, 38]. These stems from the need of generative control of human sketches which otherwise proved to be extremely challenging using CNN-based methods [44]. Endeavors then extended to the discriminative domain as well and yield promising result [30, 38] with the advent of transformers [51]. Our approach combines the best of two worlds that leverages both the stroke-level information in vector sketch and the representational capability of CNN that takes raster visual patch as input.

Category vs. instance This dichotomy corresponds to how a sketch is *created* – rendered based on a category-name [13, 19] or a (real or mental) picture of a specific object instance [55, 39]. A sketch can thus present different granularities of visual cues (e.g., prototypical vs. specific object detail) and reflect different instantaneous mental process (e.g., drawing for recognizability vs. resemblance). In practice, these two sketch variations are utilized for the development of different tasks, i.e., category-level (many sketches have same ground-truth objective) [42, 56, 41] or fine-grained (a sketch corresponds to one definitive answer) sketch analysis [39, 28, 24]. In this paper, we show the efficacy of the proposed method is generally applicable to either sketch, and in both discriminative and generative tasks.

Sketch modeling In line with more general-purpose vision research, most sketch studies focus on invariant feature representation engineering or learning [5, 13, 25, 56, 41] to advance the benchmark performance of sketch tasks. Similarly, with great strides made in the field of neural image synthesis [49], models on generating sketches from either a vector sketch [19, 10, 16] or raster photo [44, 52] begin to emerge, including a generative agent that exhibits human-level performance at a Pictionary-like sketching game [3]. More recent works have attempted to leverage insights from the human sketching process. The two most explicit model of sketching to our best knowledge are [33, 34]. [33] abstracts sketch by estimating which constituent strokes can be safely removed without affecting recognizability. [34] identifies different human sketching behaviors in contour and detail rendering and argues that both parts can only be modeled effectively when they are factorized. These works albeit on abstraction, carry an entirely different focus of abstract sketch *generation*. We on the other handle model abstraction at design for the more general task of sketch representation learning.

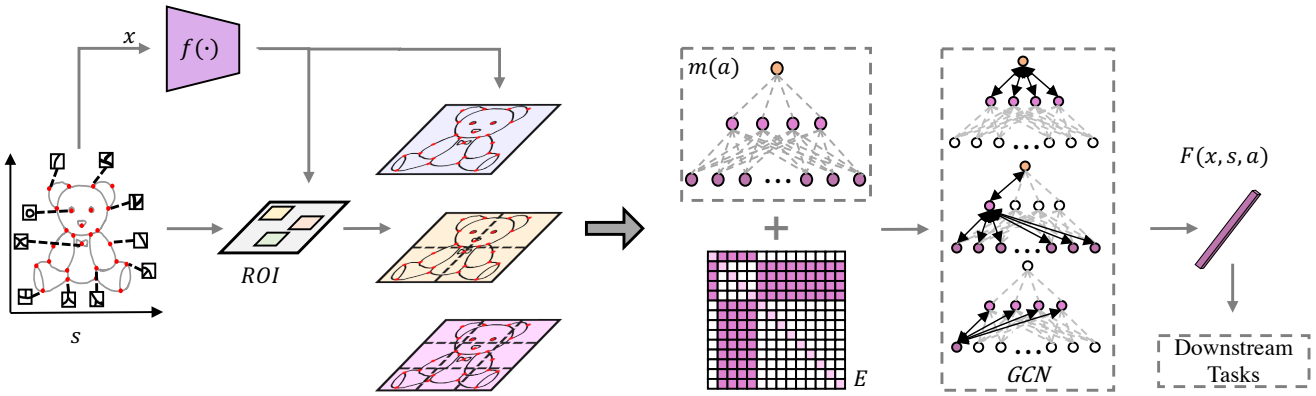

Figure 2: Schematic of our proposed framework with the choice of $h(\cdot)$ as graph convolutional network. More details in text.

3. Methodology

A sketch is a sequence of N stroke points $s = (s_1, s_2, ..., s_N)$. To assign each s_i with its associated visual patch, a sketch is also rendered as a raster image x and a local visual patch p_i is cropped centered around s_i. p_i then serves as the visual correspondence surrounding the receptive field of a stroke point and faithfully presents the drawing behaviors. We abstract a sketch structure by building a coarse-to-fine block hierarchy with q levels to organize the raw visual cues for more efficient learning. At each level l, we divide x into $l \times l$ grid blocks. This then introduces K multi-granularity visual blocks $a = (a_1, a_2, ..., a_K)$, where $K = \sum_{l=1}^{q} l^2$ and each a_k owns its unique group of p_is based on the geological proximity. The goal of this paper is to learn a CNN feature extractor $f(\cdot)$ and a parameterized feature aggregator $h(\cdot)$ that turns (x, s, a) into synergistic single visual representation $F(x, s, a)$ for different downstream sketch analysis tasks. A schematic of our framework is demonstrated in Figure 2.

3.1. Defining an Abstraction Model

Structural Abstraction As said, our choice of abstracting the structure of a sketch is to uniformly divide x into a across different granularities. Such representation, although seemingly simplistic, is intuitively amenable to structural deformations and thus vital for sketch modeling. Formally, given a raster sketch x of size $W \times H$ and one a_k at the l^{th} level along the structural hierarchy, we obtain the central coordinates of a_k with width and length $\lceil \frac{W}{l} \rceil$, $\lceil \frac{H}{l} \rceil$:

$$c(a_k) = ((0.5 + i) \times \lceil \frac{W}{l} \rceil, (0.5 + j) \times \lceil \frac{H}{l} \rceil) \quad (1)$$

where $i \in \{0, 1, ..., l-1\}, j \in \{0, 1, ..., l-1\}$ and $\lceil \cdot \rceil$ is the ceiling function to ensure real integer coordinate. The actual values of i and j depend on the location of a_k and l. If $l = 2$ and a_k represents the upper left quadrant, $i = j = 0$. Each stroke point s_i is then linked to their corresponding level-specific $c(a_k)$ based on coordinate-wise nearest neighbor. It's worthy to note when $q > 1$, we have multiple values of l which means a s_i can belong to multiple grid blocks and play different roles based on the specific abstraction level.

Appearance Abstraction Given that each a_k now corresponds to a set of stroke points $\{s_i \in a_k\}$, it is straightforward to feed the respective sampled patches p_is into $f(\cdot)$ and extract their representations. Such practice however becomes less feasible in practical implementations. As the number of p_is one a_k attaches can be as many as few tens, this calls for the same number of forward passings of $f(\cdot)$ and creates optimization barriers. To this end, we introduce the popular Region Of Interest (ROI) pooling layer [18], which utilizes single feature map of the original image and generates representation for each patch proposal directly from it – thus saving the need for multiple inferences via $f(\cdot)$. We then abstract the appearance within a_k by averaging the visual features of all its belonging stroke points, denoted as $mean\{\text{ROI}(f(x), p_i)\}_{i=\{s_i \in a_k\}}$. Our way of appearance abstraction can be seen as resorting to a visual mean that cancels out the the variance in abstract drawings. Such reasoning leads to our preference say over seeking a visual max, which is sensitive to uncommon strokes, and computationally known to bias on detecting textures not present in sketch data [15]. To further compensate the inevitable loss of precision brought by the boundary approximation in ROI pooling, and provide the same global context along the feature hierarchy, we reinforce each a_k with $f(x)$ and derive our final formulation for a_k:

$$m(a_k) = f(x) + mean\{\text{ROI}(f(x), p_i)\}_{i=\{s_i \in a_k\}} \quad (2)$$

3.2. Designing Feature-Aggregator $h(\cdot)$

To aggregate $m(a)$ into one single representation requires more than simple element-wise fusion. Recall the

two key characteristics of $m(a_k)$: (i) It is the mean representation of all stroke points within a grid block of a certain location and size. This makes each $m(a_k)$ dramatically different. (ii) each $m(a_k)$ (e.g., $l = 2$) can contain the same stroke points with others (e.g., $\{a_k \in l = 4\}$). This indicates possible connections between $m(a_k)$ at different abstraction level, which can be beneficial if modeled properly. We draw inspirations from such analysis and put forward three choices of solutions.

$h(\cdot)$ as graph message passing *Graph construction:* We treat each $m(a_k)$ as a graph node and explore their dynamics via graph convolutional networks (GCN). Top-down nearest neighbor along the abstraction hierarchy is used to construct the edge links between nodes. That is, for each a_k corresponding to a value of l, we will connect it with all other grid blocks within an abstraction level of $l + 1$. An adjacency matrix $E \in \mathbb{R}^{K\times K}$ can then be formed, where each entry $e_{i,j} = 1$ represents the establishment of a link between a_i and a_j, and zero-valued otherwise. For self-connection e_{ii}, we simply set its value to 0.5 for regularization purpose. *Graph learning:* we choose [26], a popular variant of GCN that executes a simple layer-wise propagation rule via the first-order approximation of spectral graph convolution. Specifically, assume X^0 as the original node feature matrix with each row vector as $m(a_k)$, and $h(X^0) = X^t$ computed after t GCN layers, we formulate our representation learning as follows:

$$X^t = ReLU(\tilde{D}^{-\frac{1}{2}}\tilde{E}\tilde{D}^{-\frac{1}{2}}X^{t-1}W^t) \tag{3}$$

where $\tilde{E} = E + I$, $\tilde{D} = \sum_j \tilde{E}_{ij}$, W^t is a layer-wise learnable weight matrix. We obtain our final representation $F(x, s, a)$ with a mean aggregator.

$h(\cdot)$ as relational critic [48, 2] We can also directly predict the compatibility between every two $m(a_k)$ with a non-negative scalar and leverage it as a critical guidance for multi-granularity feature aggregation. The idea is then to concatenate the features, $(m(a_i), m(a_j))_{i,j\in[1,K],i\neq j}$, and feed it into MLP for relational identification. This results in a matrix $M = \mathbb{R}^{K\times(K-1)}$ with each entry M_{ij} formulated as: $\texttt{softplus}(W_2(\texttt{concat}(W_1(m(a_i)), W_1(m(a_j)))))$. W_1, W_2 are two learnable weight matrices. We obtain our final representation $F(x, s, a)$ by computing a weighted sum of features with reference to M:

$$F(x, s, a) = \sum_{i=1}^{K}\sum_{j=1}^{K-1} M_{i,j} * \texttt{concat}(W_1(m(a_i)), W_1(m(a_j))) \tag{4}$$

$h(\cdot)$ as feature extrapolation [47] Our last approach sees $\{m(a_k)\}_{k=1,2...,K}$ as a sequence of inputs at different time steps in a recurrent neural network (RNN). This means rather than attempting to best interpolate between the features of different grid blocks, like the two choices of $h(\cdot)$ above, we rely on the inner mechanism of RNN (LSTM [22] in this paper) to extrapolate a new representation based on its past experience. We take the output of the final time step as our final representation $F(x, s, a)$.

3.3. Deploying $F(x, s, a)$ to Downstream Tasks

Category-level recognition $F(x, s, a)$ is used here for sketch recognition. We append a fully-connected layer $\phi(\cdot)$ to transform the dimension of $F(x, s, a)$ to the number of categories C required to discriminate. With the one-hot label y, our training objective is then to minimize the cross-entropy softmax loss common for classification problem:

$$L_c = y \log(\frac{\exp^{\phi(F(x,s,a))_c}}{\sum_{u=1}^{C}\exp^{\phi(F(x,s,a))_u}}) \tag{5}$$

Instance-level retrieval To examine the efficacy of the proposed method for instance-level sketch analysis, we apply $F(x, s, a)$ to the problem of fine-grained sketch-based image retrieval (FG-SBIR) [29]. Given a sketch as input, FG-SBIR aims to find one particular photo that shares similar instance-level visual traits. We take $f(\cdot)$ as the shared network backbone for both sketch and photo, i.e., Siamese network, and optimize it under triplet ranking loss, which are two near-ubiquitous choices in the FG-SBIR literature [55, 39]. This leads to our training objective as:

$$L_{tri} = max(0, \Delta + d(F(x, s, a), f(p^+)) - d(F(x, s, a), f(p^-))) \tag{6}$$

where Δ is a hyper-parameter and if the two photos are ranked correctly within the margin, the triplet term will be not penalized. Note that since photo can't be organized in vector format, we obtain its features using $f(\cdot)$ only. $d(\cdot, \cdot)$ is the ℓ_2 distance between its two elements.

Healing partial sketches By removing a fraction of points from s, the goal of sketch healing task is to generate a novel and complete sketch stroke-by-stroke that best resembles this corrupted partial input $\hat{s}$, other than filling in the missing parts. To maximally demonstrate the efficacy of our approach, we take $F(\hat{x}, \hat{s}, a)$ as conditional latent vector input, and feed it to the same generative LSTM decoder adopted by almost all contemporary sketch generative models without any changes [19, 8, 46]. Assume the generative decoder as q_θ, we derive our formulation as:

$$L_{heal} = minE_{q(z|\hat{s})}[\log p_\theta(s|W_{\mu,\sigma}(F(\hat{x}, \hat{s}, a))] \tag{7}$$

where $W_{\mu,\sigma}$ are two fully-connected layers that transform $F(\hat{x}, \hat{s}, a)$ to two vectors that uniquely determine the mean and variance of an i.i.d Gaussian distribution. In practice, what p_θ models is more than the offset distance $(\Delta x, \Delta y)$ between two consecutive stroke points. Pen states are also estimated including touching, lifting and ending. Readers please refer to [19] for more details.

4. Experimental Settings

To pinpoint the advantages of our approach, we control all baselines and ablated variants to use the same network backbone of ResNet-50 [20] and optimization strategy, wherever possible. Learning rates and hyper-parameters are not grid-searched for optimal performance. Only training iterations may vary across methods and datasets.

Dataset and pre-processing **Dataset:** QuickDraw [19] is by far the largest free-hand sketch dataset, which is collected via an online game and where the players are asked to draw objects belonging to a particular object class in less than 20 seconds. It contains 345 object categories with each containing 70K training samples, 2.5K validation and 2.5K testing samples. For the recognition task: we sample 7K samples within each category as our training set. This gives us a total of 2.415M training data, which is slightly smaller than that in [38] (2.5M). We use all the testing samples (862K) for evaluation. For the generative healing task: following the category selection principle of [46][1], we choose 7 categories including `airplane`, `angel`, `bear`, `bird`, `butterfly`, `cat`, `pig`. For retrieval task: QMUL_Shoe_V2 and QMUL_Chair_V2 [1] are two largest single-category product-level FG-SBIR datasets to date with 6,648 and 2,000 sketch-photo paired data respectively. Of them, we follow the standard split that uses 5982 and 964 pairs for training and the rest for testing. **Pre-processing:** It requires more care when we work with sketch data in both vectorized and rasterized forms. The length of human sketching sequence varies from a few to thousands, which can destabilize learning and raise meaningless inference time. We thus restrict the maximum length of s as N_{max} and when N is larger than N_{max}, we start sampling. To ensure sampling uniformity, we continue to divide a sketch into grid blocks and sample one stroke point within each (skip if all its stroke points have been sampled). Such division strategy is conducted from coarse to fine until the total number of sampled points reaches the limits, N_{max}. We set $N_{max} = 20$ throughout our experiments.

Implementation details All experiments are conducted on a single NVIDIA V100 GPU with $f(\cdot)$ initialized with the pre-trained weights from ImageNet [11]. We use Adam optimizer for training both recognition and healing tasks with initial learning rate as 1e-3 for 5 epochs and decreased to 1e-4 for another 5 epochs. We use a batch size of 128 for both tasks. To ensure the gradient stability of the generative LSTM decoder in our healing task, we clip the gradient when it exceeds the numerical value 1. Due to the relative smaller size of FG-SBIR dataset, we follow the tradition of FG-SBIR community [55, 36] and adopt SGD optimizer with momentum value 0.9 throughout. We train 50K and 30K iterations on QMUL_Shoe_V2 and QMUL_Chair_V2 respectively with a triplet batch size of 16. Δ is set to 0.1. To offset the data bias introduced by human sketching, e.g., stroke width, blurriness, we process all sketch data with a one-stop post-processing solution via [43]. The size of sampled local visual patch p_i we extract for each s_i is 32×32 if without explicitly mentioned.

Competitors **Recognition**: **ResNet-50** is our network backbone, and that by building our method directly on top explicitly demonstrates the efficacy of our abstract modeling of human sketches. **SketchMate** [53], **SketchGCN** [54] and **SketchFormer** [38] are three contemporary sketch recognition methods with various advanced designs, including dual-branch networks, designing static-dynamic graph convolutions and stacking transformer layers. We include three variants of the proposed method based on the different choices of feature-aggregator $h(\cdot)$, namely **Ours-Graph**, **Ours-Critic** and **Ours-Extrapolate**. Without further ablation, we take $q = 2$ (K = $1 + 2 \times 2 = 5$) throughout our recognition experiment. **Retrieval:** we compare with two FG-SBIR baselines, **Siamese-Tri** [55] and **Siamese-Tri-SA** [45]. Siamese-Tri is the pioneering FG-SBIR work that still underpins the basis of contemporary FG-SBIR models. We differ from it by extracting sketch representation with our $F(x, s, a)$ and keep the photo feature learning unchanged with $f(x)$. Siamese-Tri-SA advances Siamese-Tri by introducing spatial attention module in network backbone and modifying the heuristics-based triplet ranking loss to a learnable higher-order energy function. Like recognition task, we also include our own three variants under different choices of $h(\cdot)$, but we take a different value of $q = 3$ since we believe task requires instance-level differentiation naturally requires a finer-grained structural abstraction. **Generation:** Our competitors comprise of three existing works for vector sketch generation, **SketchRNN** [19], **SketchPix2seq** [8] and **SketchHealer** [46]. SketchRNN is the pioneering work for generative sketch modelling using deep learning methods with a encoder-decoder architecture. Subsequent works focus on improving the RNN encoder of SketchRNN, with CNN and GCN alternatives in SketchPix2seq and SketchHealer respectively. Our $F(x, s, a)$ can also be seen as an attempt to advance generative sketch encoder and thus directly comparable. Since sketch healing task takes corrupted partial sketch as input, smaller grid blocks will be inevitably suffered more by the visual discontinuities. We therefore take $q = 2$. We empirically validate on a corruption level of sketch with 30% throughout, i.e., $\hat{s}$ is always formed by randomly removing 30% stroke points from s. Finally, we regard Ours-Graph as our full model, i.e., **Ours**, because of its superior performance across three sketch analysis tasks.

[1] Some QuickDraw categories are less prominent for a learning objective than others. For example, categories like `line`,`circle`,`hexagon` naturally present very little intra-category variations and thus does not serve as competitive data when validating a model's generative capability.

5. Analysis for Discriminative Tasks

Our first discovery is that introducing explicit abstraction model for sketch feature learning can be a simple but highly effective performance booster for sketch discriminative tasks. With as few as one abstraction level, it is able to consistently outperform the baselines and achieves competitive results. Below is a more detailed analysis with reference to Table 1, 2, 3 and 4.

What does the competitiveness of ResNet-50 tell us? It is somehow surprising to see that by fine-tuning a simple prototypical ImageNet pre-trained model (ResNet-50) leads to significantly better performance over some baselines (SketchGCN [46], SketchFormer [38]) specifically designed for the sketch recognition task. A closer inspection reveals the reason behind: these methods all aim to directly utilize either the absolute or relative coordinates of human sketches for feature learning. It is then natural to disregard CNNs and adopt transformer or dynamic graph pooling layer that inherently admit sequence input. These attempts originate from the observation that compared with static photos, human sketching is a temporal process and the dynamics in between should be beneficial if mined properly. Our approach shares the intuition on using vector sketches (as per the input s) but also advocates the representational power of CNN (as per $f(x)$), i.e., striving the best from both worlds (vector and raster).

Does the choice of $h(\cdot)$ matter? Yes. The conclusion is evident from the poor performance of non-learning based methods (Ours-Mean/Max/Concat), which are detrimental and worse than baselines (ResNet-50). By introducing learnable parameters (Ours-Graph/Critic/Extrapolate), positive impact begins to emerge but the learning strategy still matters. The fact that Ours-Critic is even inferior to Ours-Extrapolate (an RNN model only implicitly exploits the dynamics between abstraction levels), further shows that how delicate a choice of $h(\cdot)$ can be. Ours-Graph comes to the rescue by graph learning on a hierarchical affinity matrix which is more fitting given the nature of our representation – a coarse-to-fine abstraction of sketch data.

Are more abstraction levels always better? From Table 3, we can see that different tasks demand different abstraction levels. For tasks requiring finer-grained visual discrimination, more abstraction levels with local control are naturally called into play, and indeed proved so in our empirical evaluation – best FG-SBIR performance is obtained with $q = 3$ compared with $q = 2$ for that in recognition task. On the other hand, we argue that the size of sketch data is another key factor that determines the best value of q: only when the visual source is large enough can it be able to support more abstraction levels. We validate the effect of input sizes in Table 4 and the result confirms our hypothesis. With input size enlarging from $(W = 224, H = 224)$, performance of $q = 4$ gets increasingly improved and surpasses that of $q = 2$ and $q = 3$ for recognition and FG-SBIR task respectively, when the dimension of visual input reaches $(W = 512, H = 512)$.

ResNet-50	SketchMate [53]	SketchGCN [54]	SketchFormer [38]
78.76%	79.44%	77.31%	78.34%
Ours-Graph	**Ours-Critic**	**Ours-Extrapolate**	**Ours-Mean**
81.51%	80.42%	80.91%	77.41%
Ours-Max	**Ours-Concat**		
77.80%	78.22%		

Table 1: Comparisons of performance for sketch recognition task. Numbers reported represent top-1 classification accuracy.

Method \ Dataset	QMUL_Shoe_V2		QMUL_Chair_V2	
	Acc@1	Acc@10	Acc@1	Acc@10
Siamese-Tri [55]	30.83	79.28	45.25	89.64
Siamese-Tri-SA [45]	31.08	80.03	47.24	90.85
Ours-Graph	**32.33**	**79.63**	**52.89**	**94.88**
Ours-Critic	31.07	79.63	47.60	90.08
Ours-Extrapolate	31.59	80.71	49.22	90.44
Ours-Mean	29.13	80.48	43.24	89.25
Ours-Max	29.58	79.88	44.00	90.42
Ours-Concat	30.18	80.78	44.83	90.41

Table 2: Comparisons of performance for FG-SBIR task. Acc@K represents whether the correct photo corresponding to the query is within the first K position in the ranking list.

Task \ Level	(q=1, K=1)	(q=2, K=5)	(q=3, K=14)	(q=4, K=30)
Recognition	80.04%	**81.51%**	80.31%	80.07%
FG-SBIR (Shoe)	30.93%	31.23%	**32.33%**	31.52%
FG-SBIR (Chair)	46.47%	47.64%	**52.89%**	50.41%

Table 3: Effects of different abstraction levels q for two sketch discriminative tasks under our full model.

What does abstraction at feature-level looks like? In Figure 3, we further carry out model visualization to show how different abstraction levels in our model capture different perspectives of visual traits and offer insights on how better feature learning is taking place. We start with a set of sketch samples from one category and extract their features from a random a_k at each abstraction level l. We cluster the level-specific features using t-SNE [50] and visualize the sketches close in the embedding space. We can observe that our learned representation aligns well to our abstract modeling of sketch data with clear sign of hierarchical behaviors. At $l = 1$ our model recognizes the general shapes and poses of an objects, and gradually moves onto discern and summarize more subtle and local visual patterns as l increases – a strong evidence for appearance abstraction.

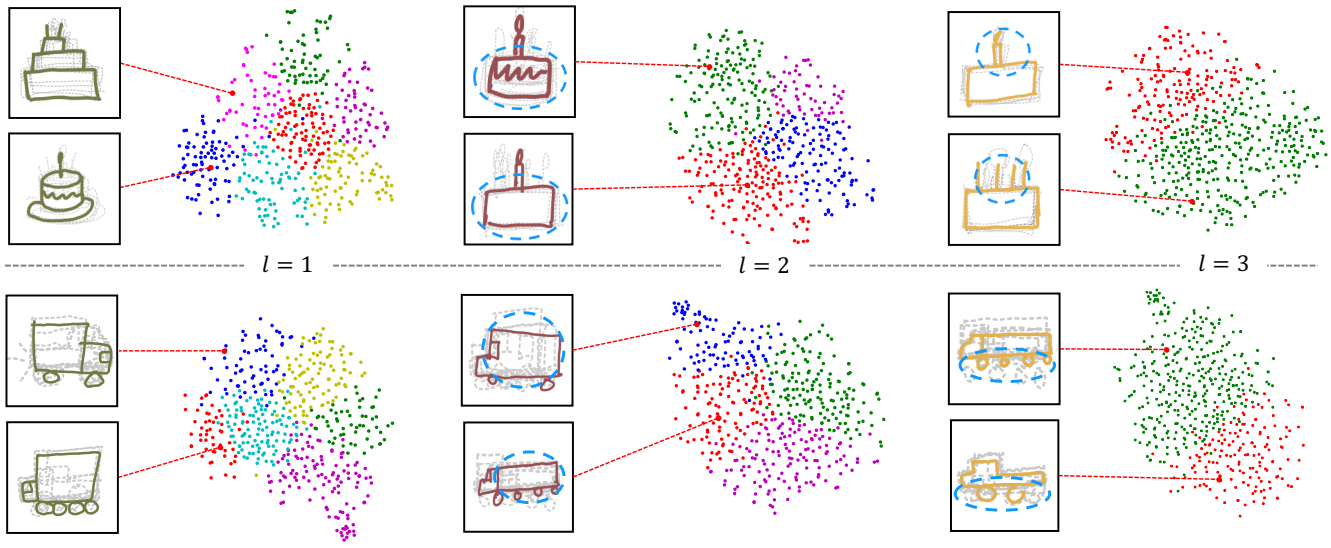

Figure 3: Qualitative evidence of abstraction happening at feature level. We visualize the sketch samples that are close in the feature embeddings across different abstraction levels. More details in text.

(H=224, W=224)	**Recognition**	(q=2, K=5)	81.51%
		(q=4, K=30)	80.07%
	FG-SBIR (Shoe)	(q=3, K=14)	32.33%
		(q=4, K=30)	31.52%
	FG-SBIR (Chair)	(q=3, K=14)	52.89%
		(q=4, K=30)	50.41%
(H=384, W=384)	**Recognition**	(q=2, K=5)	81.67%
		(q=4, K=30)	80.90%
	FG-SBIR (Shoe)	(q=3, K=14)	32.84%
		(q=4, K=30)	32.90%
	FG-SBIR (Chair)	(q=3, K=14)	53.63%
		(q=4, K=30)	53.81%
(H=512, W=512)	**Recognition**	(q=2, K=5)	81.89%
		(q=4, K=30)	**82.05%**
	FG-SBIR (Shoe)	(q=3, K=14)	33.27%
		(q=4, K=30)	**33.51%**
	FG-SBIR (Chair)	(q=3, K=14)	54.07%
		(q=4, K=30)	**54.25%**

Table 4: Impact of larger sketch size on the best abstraction levels for two sketch discriminative tasks. $q = 2$ and $q = 3$ correspond to the best q value for recognition and FG-SBIR respectively when H=W=224.

6. Analysis for Generative Tasks

Quantitative results Evaluating image synthesis models remain an open question with few existing advanced metric like FID [21] designed for natural images only. Consequently, most previous studies on generative human sketch modeling either run human perceptual studies or explore computational metrics to predict human perceptual similarity judgment. We perform both quantitative evaluations. **Computational metric:** Given a corrupted partial sketch, we measure the generative healing capability of different models by testing the recognizability of their synthetic outcomes. The results in Table 5 show that our model outperforms all competitors. Particularly, the significant gap between Ours and SketchRNN and SketchPix2seq echoes our findings in discriminative task analysis in Sec. 5 that it is critical to leverage both sequential nature and CNN-based visual learning for robust sketch representation. **Human perceptual study:** We first form a test set for human study with 50 unseen sketches randomly selected across all 7 object categories. Each sketch is subjected to manual corruption before feeding into four methods for healing effect. We recruit 10 human judges and ask each to complete 100 comparative trials. In each trial, each worker is shown one corrupted partial sketch input and four synthetic sketches from different methods with orders randomized, and asked to choose one generation result based on two criteria: (i) correspondence: which sketch is a more faithful resemblance of the corrupted input; (ii) naturalness: which sketch after healing is more visually pleasant with natural sketching curves and less discontinuities. In Table 6, we can see that our model is preferred over the other competitors, drawing the same conclusion as our computational metric.

Qualitative results Figure 5 shows some examples produced by our method and other competitors. Following observations can be made: (i) our method is not only able to heal the corrupted partial sketch input with a complete novel sketch rendition just like humans do, but can also able to keep the general appearances and structures, e.g., cat whiskers and nose. (ii) the visualization aligns well with the quantitative evaluation that the superiority of our method over SketchRNN and SketchPix2seq is significant,

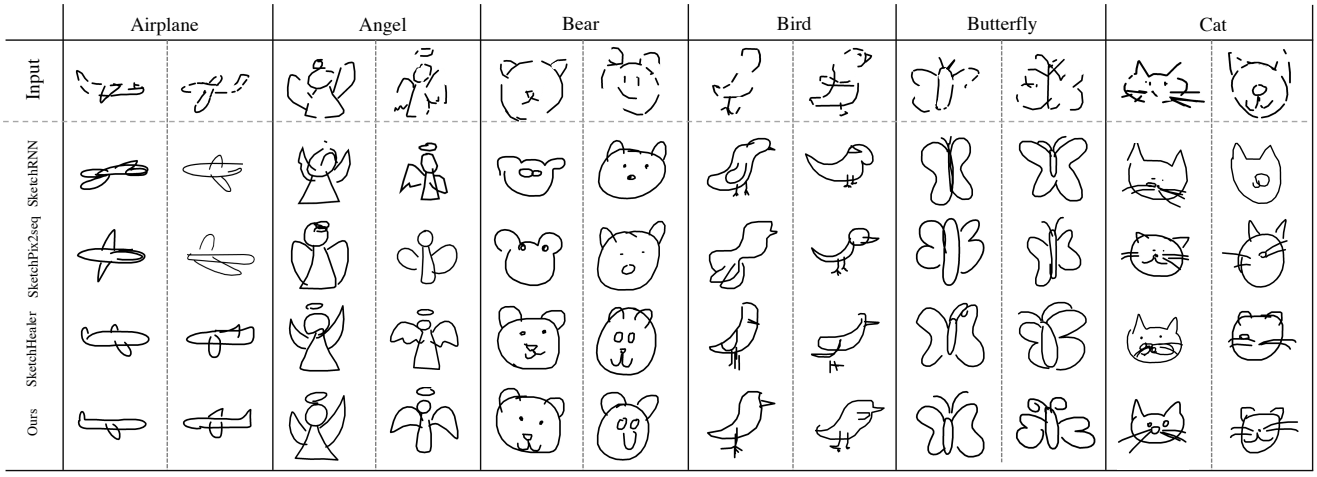

Figure 4: Qualitative comparisons on sketch healing tasks. Input corruption level is at 30%. Illustrations here have never been seen by its corresponding model during training.

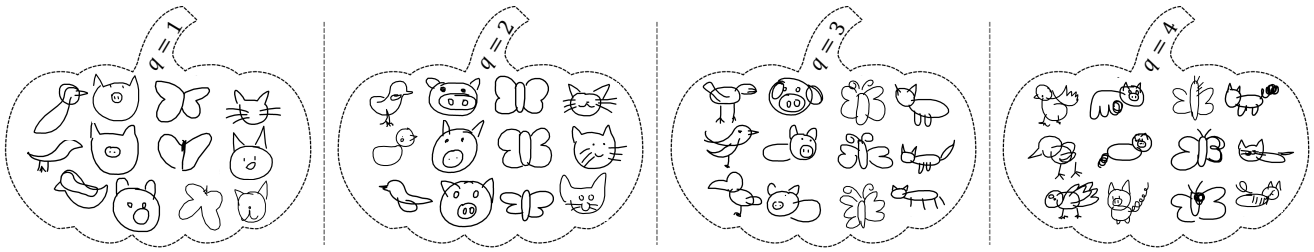

Figure 5: Comparisons of typical sketches generated by our methods under different abstraction levels. Larger q value means finer structural abstraction with finer-grained visual control.

SketchRNN [19]	**SketchPix2seq** [8]	**SketchHealer** [46]	**Ours**
37.14%	33.25%	58.67%	**60.01%**

Table 5: Recognition accuracy of s from generative healing of $\hat{s}$ under different methods.

SketchRNN [19]	**SketchPix2seq** [8]	**SketchHealer** [46]	**Ours**
13.83%	11.44%	35.51%	**39.22%**

Table 6: Preference of humans on the sketch healing results under different methods. Chance is at 25%.

and despite the gap becomes less evident when comparing with SketchHealer, it is still able to identify our advantageous subtleties with a zoom-in look. e.g., the beak of the bird and antennas of the butterfly. (iii) the effect brought with explicit abstraction of sketch data is clearly manifested in the rendering of more regularized visual structures, e.g., the airplane rudder and the angel wings. We further demonstrate the effects of different abstraction levels on the generative model behavior in Figure 5, and draw the same conclusion throughout the paper – more abstraction levels, finer-grained visual learning control.

7. Conclusion

We recognized the need for learning sketch representations that specifically capture their inherent abstract nature. We presented a simple yet very effective representation by factorizing the sketch abstraction process into appearance and structure. We explored different frameworks that best learn a synergistic feature across multiple granularity levels. We are able to control the expressive granularity a sketch representation, and tailor solutions for different target tasks. Albeit with its simple design, we report state-of-the-art performance on a variety of sketch tasks, echoing the importance of modeling sketch abstraction at design. Last but not least, we hope this paper can trigger potential discussions on how to interpret the abstract nature of human sketch data and model accordingly for efficient sketch representation.

Acknowledgements We thank the anonymous reviewers for their valuable comments. This work was supported in part by the National Natural Science Foundation of China (NSFC) under joint grant # 62076034 and # 61806184. We especially thank the China Scholarship Council (CSC) for funding the first author to conduct the entirety of this project under # 202006470075.

References

[1] SketchX!-Shoe/Chair Fine-grained-SBIR dataset. http://sketchx.eecs.qmul.ac.uk, 2017. 5

[2] Antreas Antoniou and Amos Storkey. Learning to learn via self-critique. In *NeurIPS*, 2019. 4

[3] Ayan Kumar Bhunia, Ayan Das, Umar Riaz Muhammad, Yongxin Yang, Timothy M Hospedales, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. Pixelor: A competitive sketching ai agent. so you think you can sketch? *ACM Transactions on Graphics*, 2020. 2

[4] Ayan Kumar Bhunia, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Sketch less for more: On-the-fly fine-grained sketch-based image retrieval. In *CVPR*, 2020. 1

[5] Yang Cao, Changhu Wang, Liqing Zhang, and Lei Zhang. Edgel index for large-scale sketch-based image search. In *CVPR*, 2011. 2

[6] Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. Deepfacedrawing: Deep generation of face images from sketches. *ACM Transactions on Graphics*, 2020. 1

[7] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *CVPR*, 2018. 1

[8] Yajing Chen, Shikui Tu, Yuqi Yi, and Lei Xu. Sketch-pix2seq: a model to generate sketches of multiple categories. *arXiv preprint arXiv:1709.04121*, 2017. 4, 5, 8

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1

[10] Ayan Das, Yongxin Yang, Timothy Hospedales, Tao Xiang, and Yi-Zhe Song. Béziersketch: A generative model for scalable vector sketches. In *ECCV*, 2020. 2

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[12] Sounak Dey, Pau Riba, Anjan Dutta, Josep Llados, and Yi-Zhe Song. Doodle to search: Practical zero-shot sketch-based image retrieval. In *CVPR*, 2019. 1

[13] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Transactions on Graphics*, 2012. 1, 2

[14] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. A descriptor for large scale image retrieval based on sketched feature lines. In *SBM*, 2009. 1

[15] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *NeurIPS*, 2015. 3

[16] Songwei Ge, Vedanuj Goswami, C Lawrence Zitnick, and Devi Parikh. Creative sketch generation. In *ICLR*, 2020. 1, 2

[17] Arnab Ghosh, Richard Zhang, Puneet K Dokania, Oliver Wang, Alexei A Efros, Philip HS Torr, and Eli Shechtman. Interactive sketch & fill: Multiclass sketch-to-image translation. In *ICCV*, 2019. 1

[18] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 3

[19] David Ha and Douglas Eck. A neural representation of sketch drawings. In *ICLR*, 2018. 1, 2, 4, 5, 8

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[21] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 7

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997. 4

[23] Conghui Hu, Da Li, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Sketch-a-classifier: Sketch-based photo classifier generation. In *CVPR*, 2018. 2

[24] Forrest Huang, John F Canny, and Jeffrey Nichols. Swire: Sketch-based user interface retrieval. In *CHI*, 2019. 2

[25] Zhe Huang, Hongbo Fu, and Rynson WH Lau. Data-driven segmentation and labeling of freehand sketches. *ACM Transactions on Graphics*, 2014. 2

[26] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 4

[27] Brendan Klare, Zhifeng Li, and Anil K Jain. Matching forensic sketches to mug shot photos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010. 2

[28] Ke Li, Kaiyue Pang, Jifei Song, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Honggang Zhang. Universal sketch perceptual grouping. In *ECCV*, 2018. 2

[29] Yi Li, Timothy M Hospedales, Yi-Zhe Song, and Shaogang Gong. Fine-grained sketch-based image retrieval by matching deformable part models. In *BMVC*, 2014. 2, 4

[30] Hangyu Lin, Yanwei Fu, Xiangyang Xue, and Yu-Gang Jiang. Sketch-bert: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. In *CVPR*, 2020. 1, 2

[31] Fang Liu, Changqing Zou, Xiaoming Deng, Ran Zuo, Yu-Kun Lai, Cuixia Ma, Yong-Jin Liu, and Hongan Wang. Scenesketcher: Fine-grained image retrieval with scene sketches. In *ECCV*, 2020. 2

[32] Raphael Gontijo Lopes, David Ha, Douglas Eck, and Jonathon Shlens. A learned representation for scalable vector graphics. In *CVPR*, 2019. 2

[33] Umar Riaz Muhammad, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning deep sketch abstraction. In *CVPR*, 2018. 2

[34] Kaiyue Pang, Da Li, Jifei Song, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep factorised inverse-sketching. In *ECCV*, 2018. 2

[35] Kaiyue Pang, Ke Li, Yongxin Yang, Honggang Zhang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Generalising fine-grained sketch-based image retrieval. In *CVPR*, 2019. 2

[36] Kaiyue Pang, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Solving mixed-modal jigsaw puzzle for fine-grained sketch-based image retrieval. In *CVPR*, 2020. 1, 5

[37] Yonggang Qi, Yi-Zhe Song, Tao Xiang, Honggang Zhang, Timothy Hospedales, Yi Li, and Jun Guo. Making better use of edges via perceptual grouping. In *CVPR*, 2015. 1, 2

[38] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. Sketchformer: Transformer-based representation for sketched structure. In *CVPR*, 2020. 1, 2, 5, 6

[39] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics*, 2016. 1, 2, 4

[40] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *CVPR*, 2017. 1

[41] Ravi Kiran Sarvadevabhatla and Jogendra Kundu. Enabling my robot to play pictionary: Recurrent neural networks for sketch recognition. In *ACM MM*, 2016. 2

[42] Rosália G Schneider and Tinne Tuytelaars. Sketch classification and classification-driven analysis using fisher vectors. *ACM Transactions on Graphics*, 2014. 1, 2

[43] Edgar Simo-Serra, Satoshi Iizuka, and Hiroshi Ishikawa. Mastering sketching: adversarial augmentation for structured prediction. *ACM Transactions on Graphics*, 2018. 5

[44] Jifei Song, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning to sketch with shortcut cycle consistency. In *CVPR*, 2018. 2

[45] Jifei Song, Qian Yu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In *ICCV*, 2017. 5, 6

[46] Guoyao Su, Yonggang Qi, Kaiyue Pang, Jie Yang, and Yi-Zhe Song. Sketchhealer a graph-to-sequence network for recreating partial human sketches. In *BMVC*, 2020. 4, 5, 6, 8

[47] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *NeurIPS*, 2015. 4

[48] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 4

[49] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. *Computer Graphics Forum*, 2020. 2

[50] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008. 6

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[52] Alexander Wang, Mengye Ren, and Richard Zemel. Sketchembednet: Learning novel concepts by imitating drawings. *arXiv preprint arXiv:2009.04806*, 2020. 2

[53] Peng Xu, Yongye Huang, Tongtong Yuan, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, Zhanyu Ma, and Jun Guo. Sketchmate: Deep hashing for million-scale human sketch retrieval. In *CVPR*, 2018. 5, 6

[54] Lumin Yang, Jiajie Zhuang, Hongbo Fu, Kun Zhou, and Youyi Zheng. Sketchgcn: Semantic sketch segmentation with graph convolutional networks. *arXiv preprint arXiv:2003.00678*, 2020. 5, 6

[55] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In *CVPR*, 2016. 1, 2, 4, 5, 6

[56] Qian Yu, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy Hospedales. Sketch-a-net that beats humans. In *BMVC*, 2015. 1, 2

[57] Jingyi Zhang, Fumin Shen, Li Liu, Fan Zhu, Mengyang Yu, Ling Shao, Heng Tao Shen, and Luc Van Gool. Generative domain-migration hashing for sketch-to-image retrieval. In *ECCV*, 2018. 1

[58] Qingyuan Zheng, Zhuoru Li, and Adam Bargteil. Learning to shadow hand-drawn sketches. In *CVPR*, 2020. 1

[59] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016. 1