Sketch Less for More: On-the-Fly Fine-Grained Sketch Based Image Retrieval

Ayan Kumar Bhunia[1]   Yongxin Yang[1]   Timothy M. Hospedales[1,2]   Tao Xiang[1]   Yi-Zhe Song[1]

[1]SketchX, CVSSP, University of Surrey, United Kingdom   [2]University of Edinburgh, United Kingdom.

{a.bhunia, yongxin.yang, t.xiang, y.song}@surrey.ac.uk, t.hospedales@ed.ac.uk

Abstract

Fine-grained sketch-based image retrieval (FG-SBIR) addresses the problem of retrieving a particular photo instance given a user's query sketch. Its widespread applicability is however hindered by the fact that drawing a sketch takes time, and most people struggle to draw a complete and faithful sketch. In this paper, we reformulate the conventional FG-SBIR framework to tackle these challenges, with the ultimate goal of retrieving the target photo with the least number of strokes possible. We further propose an on-the-fly design that starts retrieving as soon as the user starts drawing. To accomplish this, we devise a reinforcement learning based cross-modal retrieval framework that directly optimizes rank of the ground-truth photo over a complete sketch drawing episode. Additionally, we introduce a novel reward scheme that circumvents the problems related to irrelevant sketch strokes, and thus provides us with a more consistent rank list during the retrieval. We achieve superior early-retrieval efficiency over state-of-the-art methods and alternative baselines on two publicly available fine-grained sketch retrieval datasets.

1. Introduction

Due to the rapid proliferation of touch-screen devices, the computer vision community has witnessed significant research progress in sketch-related computer vision problems [49, 41, 29, 6, 9, 4]. Among these methods, sketch-based image retrieval (SBIR) [4, 6, 9] has received particular attention due to its potential commercial applications. SBIR was initially posed as a category-level retrieval problem. However, it became apparent that the key advantage of sketch over text/tag-based retrieval was conveying *fine-grained* detail [10] – leading to a focus on fine-grained SBIR that aims to retrieve a *particular* photo within a gallery. Great progress has been made on FG-SBIR [49, 41, 29], but two barriers hinder its widespread adoption in practice – the time taken to draw a complete sketch, and the drawing skill shortage of the user. Firstly, while sketch can convey fine-grained appearance details more easily than text, drawing a complete sketch is slow

Figure 1. Examples showing the potential of our framework that can retrieve (top-5 list) target photo using fewer number of strokes than the conventional baseline method.

compared to clicking a tag or typing a search keyword. Secondly, although state-of-the-art vision systems are good at recognising badly drawn sketches [36, 50], users who perceive themselves as someone who "can't sketch" worry about getting details wrong and receiving inaccurate results.

In this paper we break these barriers by taking a view of "less is more" and propose to tackle a new fine-grained SBIR problem that aims to retrieve the target photo with just a few strokes, as opposed to requiring the complete sketch. This problem assumes a "on-the-fly" setting, where retrieval is conducted at every stroke drawn. Figure 1 offers an illustrative example of our on-the-fly FG-SBIR framework. Due to stroke-by-stroke retrieval, and a framework optimised for few-stroke retrieval, users can usually "stop early" as soon as their goal is retrieved. This thus makes sketch more comparable with traditional search methods in terms of time to issue a query, and *more easily* – as those inexperienced at drawing can retrieve their queried photo based on the easiest/earliest strokes possible [1], while requiring fewer of the detailed strokes that are harder to draw correctly.

Solving this new problem is non-trivial. One might argue that we can directly feed incomplete sketches into the off-the-shelf FG-SBIR frameworks [49, 36], perhaps also enhanced by including synthesised sketches in the training

data. However, those frameworks are not fundamentally designed to handle incomplete sketches. This is particularly the case since most of them employ a triplet ranking framework where each triplet is treated as an independent training example. So they struggle to perform well across a whole range of sketch completion points. Also, the initial sketch strokes could correspond to many possible photos due to its highly abstracted nature, thus more likely to give a noisy gradient. Last, there is no specific mechanism that can guide existing FG-SBIR model to retrieve the photo with minimal sketch strokes, leaving it struggling to perform well across a complete sketching episode during on-the-fly retrieval.

A novel on-the-fly FG-SBIR framework is proposed in this work. First and foremost, instead of the de facto choice of triplet networks that learn an embedding where sketch-photo pairs lie close, we introduce a new model design that optimizes the rank of the corresponding photo over a sketch drawing episode. Secondly, the model is optimised specifically to return the true match within a minimum number of strokes. Lastly, efforts are taken to mitigate the effect of misleading noisy strokes on obtaining a consistent photo ranking list as users add details towards the end of a sketch.

More concretely, we render the sketch at different time instants of drawing, and feed it through a deep embedding network to get a vector representation. While the other SBIR frameworks [49, 36] use triplet loss [45] in order to learn an embedding suited for comparing sketch and photo, we optimise the *rank* of the target photo with respect to a sketch query. By calculating the rank of the ground-truth photo at each time-instant t and maximizing the sum of $\frac{1}{rank_t}$ over a complete sketching episode, we ensure that the correct photo is retrieved as early as possible. Since ranking is a non-differential operation, we use a Reinforcement Learning (RL) [16] based pipeline to achieve this goal. Representation learning is performed with knowledge of the whole sequence, as we optimize the reward non-myopically over the sketch drawing episode. This is unlike the triplet loss used for feature learning that does not take into account the temporal nature of the sketch. We further introduce a global reward to guard against harmful noisy strokes especially during later stages of sketching where details are typically added. This also stabilises the RL training process, and produces smoother retrieval results.

Our contributions can be summarised as follows: (a) We introduce a novel *on-the-fly* FG-SBIR framework trained using reinforcement learning to retrieve photo using an incomplete sketch, and do so with the minimum possible drawing. (b) To this end, we develop a novel reward scheme that models the early retrieval objective, as well as one based on Kendall-Tau [18] rank distance that takes into account the completeness of the sketch and associated uncertainty. (c) Extensive experiments on two public datasets demonstrate the superiority of our framework.

2. Related Works

Category-level SBIR: Category-level sketch-photo retrieval is now well studied [3, 43, 2, 6, 5, 47, 39, 9, 24, 4, 23]. Contemporary research directions can be broadly classified into traditional SBIR, zero-shot SBIR and sketch-image hashing. In traditional SBIR [3, 5, 4, 2], object classes are common to both training and testing. Whereas zero-shot SBIR [47, 6, 9, 24] asks models to generalise across disjoint training and testing classes in order to alleviate annotation costs. Sketch-image hashing [23, 39] aims to improve the computational cost of retrieval by embedding to binary hash-codes rather than continues vectors.

While these SBIR works assume a single-step retrieval process, a recent study by Collomosse *et al*. [4] proposed an interactive SBIR framework. Given an initial sketch query, if the system is unable to retrieve the user's goal in the first try, it resorts to providing some relevant image clusters to the user. The user can now select an image cluster in order to disambiguate the search, based on which the system generates new query sketch for following iteration. This interaction continues until the user's goal is retrieved. This system used Sketch-RNN [13] for sketch query generation after every interaction. However Sketch-RNN is acknowledged to be weak in *multi-class* sketch generation [13]. As a result, the generated sketches often diverge from the user's intent leading to poor performance. Note that though such interaction through clusters is reasonable in the case of *category*-level retrieval, it is not applicable to our FG-SBIR task where all photos belong to a single class and differ in subtle ways only.

Fine-grained SBIR: FG-SBIR is a more recent addition to sketch analysis and also less studied compared to the category-level SBIR task. One of the first studies [20] addressed it by graph-matching of deformable-part models. A number of deep learning approaches subsequently emerged [49, 41, 30, 29]. Yu *et al*. [49] proposed a deep triplet-ranking model for instance-level FG-SBIR. This paradigm was subsequently improved through hybrid generative-discriminative cross-domain image generation [30]; and providing an attention mechanism for fine-grained details as well as more sophisticated triplet losses [41]. Recently Pang *et al*. [29] studied cross-category FG-SBIR in analogy to the 'zero-shot' SBIR mentioned earlier. In this paper, we open up a new research direction by studying FG-SBIR framework design for *on-the-fly and early* photo retrieval.

Partial Sketch: One of the most popular areas for studying incomplete or partial data is image inpainting [48, 51]. Significant progress has been made in this area using contextual-attention [48] and Conditional Variational Autoencoder (CVAE) [51]. Following this direction, works have attempted to model partial sketch data [22, 13, 12].
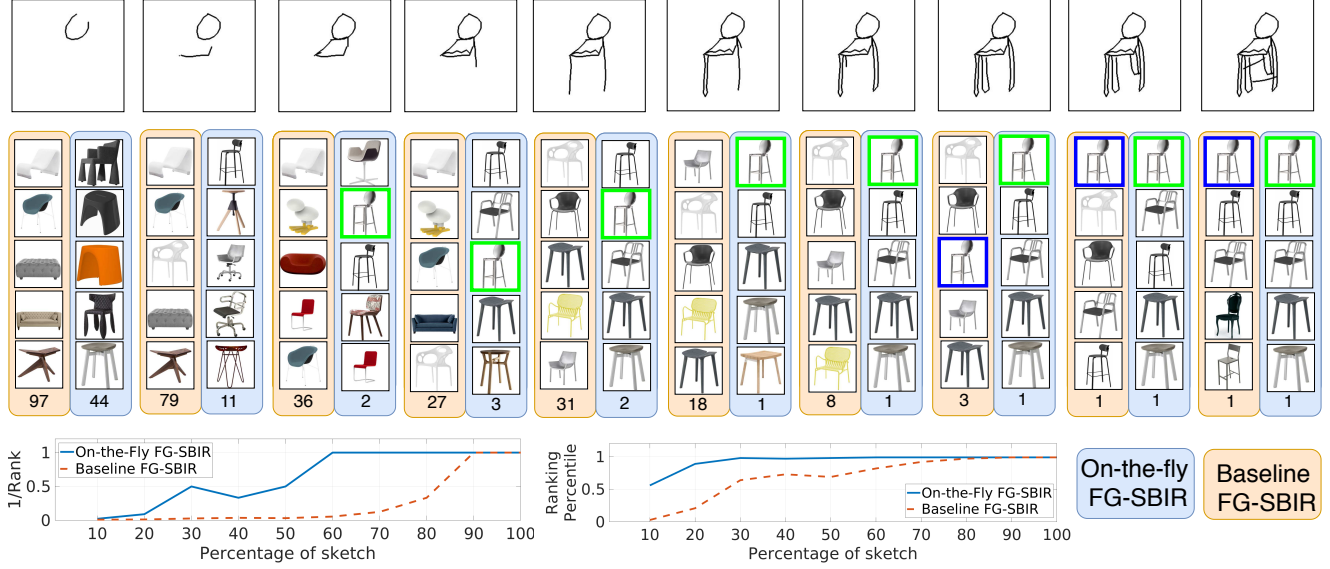
Figure 2. Illustration of proposed *on-the-fly* framework's efficacy over a baseline FG-SBIR method [41, 49] trained with completed sketches only. For this particular example, our method needs only 30% of the complete sketch to include the true match in the top-10 rank list, compared to 80% for the baseline. Top-5 photo images retrieved by either framework are shown here, in progressive sketch-rendering steps of 10%. The number at the bottom denotes the paired (true match) photo's rank at every stage.

Sketch-RNN [13] learns to predict multiple possible endings of incomplete sketches using a Variational Autoencoder (VAE). While Sketch-RNN works on sequential pen-coordinates, Liu *et al.* [22] extend conditional image-to-image translation to rasterized sparse sketch domain for partial sketch completion, followed by an auxiliary sketch recognition task. Ghosh *et al.* [12] proposed an interactive sketch-to-image translation method, which completes an incomplete object outline, and thereafter it generates a final synthesised image. Overall, these works first try to complete the partial sketch by modelling a conditional distribution based on image-to-image translation, and subsequently focus on specific task objective, be it sketch recognition or sketch-to-image generation. Unlike these two-stage inference frameworks, we focus on instance-level photo retrieval with a minimum number of sketch strokes, thus enabling partial sketch queries in a single step.

Reinforcement Learning in Vision: There has been significant progress in leveraging Reinforcement Learning (RL) [16] techniques in various computer vision problems [44, 14]. Vision applications benefiting from RL include visual relationship detection [21], automatic face aging [8], vision-language navigation [44] and 3D scene completion [14]. In terms of sketch analysis, RL was leveraged to study abstraction and summarisation by trading off between recognisability of a sketch and number of strokes [34, 27]. While these studies aimed to discover salient strokes by using RL to filter out unnecessary strokes from a given complete sketch, we focus on leveraging RL to retrieve a photo on-the-fly with a minimum number of strokes.

3. Methodology

Overview: Our objective is to design an 'on-the-fly' FG-SBIR framework, where we perform live analysis of the sketch as the user draws. The system should re-rank candidate photos based on the sketch information up to that instant and retrieve the target photo at the earliest stroke possible (see Figure 2 for an example of how the framework works in practice). To this end, we first pre-train a state-of-the-art FG-SBIR model [49, 41] using triplet loss. Thereafter, we keep the photo branch fixed, and fine-tune the sketch branch through a non-differentiable ranking based metric over complete sketch drawing episodes using reinforcement-learning.

Formally, a pre-trained FG-SBIR model learns an embedding function $F(\cdot) : I \to \mathbb{R}^D$ mapping a rasterized sketch or photo I to a D dimensional feature. Given a gallery of M photo images $G = \{X_i\}_{i=1}^{M}$, we obtain a list of D dimensional vectors $\hat{G} = \{F(X_i)\}_{i=1}^{M}$ using $F(.)$. Now, for a given query sketch S, and some pairwise distance metric, we obtain the top-q retrieved photos from G, denoted as $Ret_q(F(S), \hat{G})$. If the ground truth (paired) target photo appears in the top-q list, we consider top-q accuracy to be true for that sketch sample. Since we are dealing with on-the-fly retrieval, a sketch is represented as $S \in \{p^1, p^2, p^3, ...p^N\}$, where p^i denotes one sketch coordinate tuple (x, y), and N stands for maximum number of points. We assume that there exists a sketch rendering operation $\varnothing$, which takes a list S^K of the *first K coordinates* in S, and produces one rasterized sketch image. Our
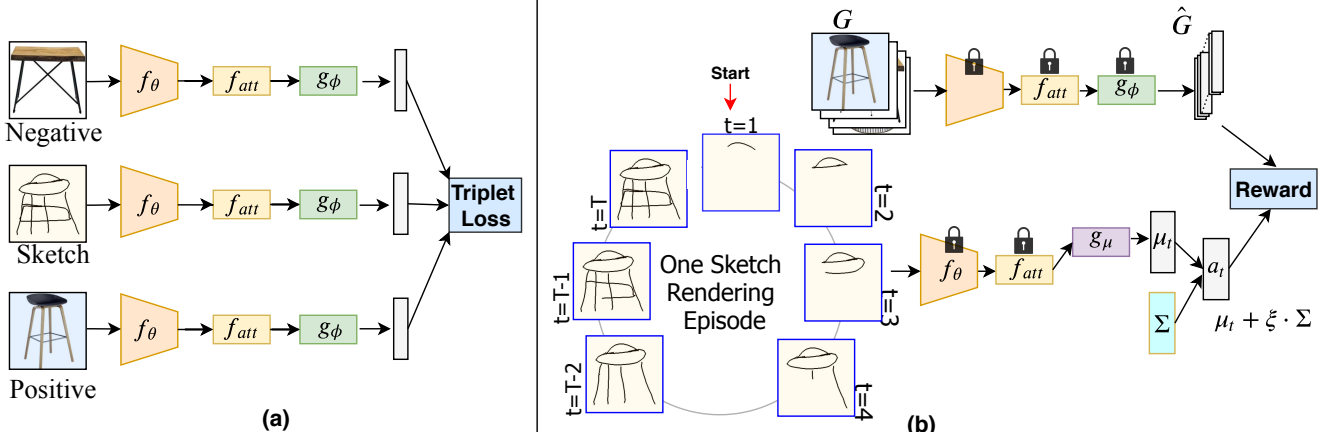
Figure 3. (a) A conventional FG-SBIR framework trained using triplet loss. (b) Our proposed reinforcement learning based framework that takes into account a complete sketch rendering episode. Key locks signifies particular weights are fixed during RL training.

objective is to train the framework so that the ground-truth paired photo appears in $Ret_q(F(\varnothing(S^K)), \hat{G})$ with a minimum value of K.

3.1. Background: Base Models

For pre-training, we use a state-of-the-art Siamese network [41] with three CNN branches with shared weights, corresponding to a query sketch, positive and negative photo respectively (see Figure 3(a)). Following recent state-of-the-art sketch feature extraction pipelines [6, 41], we use soft spatial attention [46] to focus on salient parts of the feature map. Our baseline model consists of three specific modules: (a) f_θ is initialised from pre-trained InceptionV3 [42] weights, (b) f_{att} is modelled using 1x1 convolution followed by a softmax operation, (c) g_ϕ is a final fully-connected layer with l_2 normalisation to obtain an embedding of size D. Given a feature map $B = f_\theta(I)$, the output of the attention module is computed by $B_{att} = B + B \cdot f_{att}(B)$. Global average pooling is then used to get a vector representation, which is again fed into g_ϕ to get the final feature representation used for distance calculation. We considered f_θ, f_{att}, and g_ϕ to be wrapped as an overall embedding function F. The training data are triplets $\{a, p, n\}$ containing sketch anchor, positive and negative photos respectively. The model is trained using *triplet loss* [45] that aims to reduce the distance between sketch anchor and positive photo $\beta^+ = \|F(a) - F(p)\|_2$, while increasing the distance between sketch anchor and negative photo $\beta^- = \|F(a) - F(n)\|_2$. Hence, the triplet loss can be formulated as $max\{0, \mu + \beta^+ - \beta^-\}$, where μ is the margin hyperparameter.

3.2. On-The-Fly FG-SBIR

Overview: We model an on-the-fly FG-SBIR model as a sequential decision making process [16]. Our agent takes *actions* by producing a feature vector representation of the

sketch at each rendering step, and is *rewarded* by retrieving the paired photo early. Due to computation overhead, instead of rendering a new sketch at every coordinate instant, we rasterize the sketch a total of T times, i.e., at steps of interval $\lfloor \frac{N}{T} \rfloor$. As the photo branch remains constant, we get $\hat{G}$ using the baseline model. We train the agent (sketch-branch) to deal with partial sketches. In this stage we fine-tune the sketch branch only, aiming to make it competent in dealing with partial sketches. Considering one sketch rendering episode as $S \in \{p_1, p_2, p_3, ..., p_T\}$, the agent takes state $s_t = \varnothing(p_t)$ as input at every time step t, producing a continuous 'action' vector a_t. Based on that, the retrieval environment returns one reward r_t, mainly taking into account the pairwise distance between a_t and $\hat{G}$. The goal of our RL model, is to find the optimal policy for the agent that maximises the total reward under a complete sketch rendering episode.

Triplet loss [41, 45] considers only a single instant of a sketch. However, due to creation of multiple partially-completed instances of the same sketch, a diversity is created which confuses the triplet network. In contrast, our approach takes into account the complete episode of progressive sketch generation before updating the weights, thus providing a more principled and practically reliable way to model partial sketches.

Model: The sketch branch acts as our agent in RL framework, based on a stochastic continuous Gaussian policy [7], where action generation is represented with a multivariate Normal distribution. Following the typical RL notation, we define our policy as $\pi_\Theta(a|s)$. Θ encases the parameters of policy network comprised of pre-trained f_θ and f_{att} which remains fixed, and a fully-connected trainable layer g_μ that finally predicts the mean vector μ of the multivariate Gaussian distribution. Please refer to Figure 3(b) for an illustration. At each time step t, a policy distribution $\pi_\Theta(a_t|s_t)$ is constructed from the distribution parameters predicted by

our deep network. Following this, an action is sampled from this distribution, acting as a D dimensional feature representation of the sketch at that instant, i.e. $a_t \sim \pi_\Theta(\cdot|s_t)$. Mathematically, this Gaussian policy is defined as:

$$\pi_\Theta(a_t|s_t) = \sqrt{\frac{1}{(2\pi)^D |\Sigma|}} \times$$
$$\exp\left\{-\frac{1}{2}(a_t - \mu_t)^\top \Sigma^{-1}(a_t - \mu_t)\right\} \quad (1)$$

where the mean $\mu_t = g_\mu(s'_t) \in \mathbb{R}^D$, and s'_t is obtained via a pre-trained f_θ and f_{att} that take state $s_t = \varnothing(p_t)$ as its input. Meanwhile, Σ is a standalone trainable diagonal covariance matrix. We sample action $a_t = \mu_t + \xi \cdot \Sigma$, where $\xi \sim \mathcal{N}(0, I)$ and $a_t \in \mathbb{R}^D$.

Local Reward: In line with existing works leveraging RL to optimize non-differentiable task metrics in computer vision (e.g., [33]), our optimisation objective is the non-differentiable *ranking* metric. The distance between a query sketch embedding and the paired photo should be lower than the distance between the query and all other photos in G. In other words, our objective is to minimise the rank of paired photo in the obtained rank list. Following the notion of maximising the reward over time, we maximise the *inverse rank*. For T sketch rendering steps under a complete episode of each sketch sample, we obtain a total of T scalar rewards that we intend to maximize:

$$R_t^{Local} = \frac{1}{rank_t} \quad (2)$$

From a geometric perspective, assuming a high value of T, this reward design can be visualised as maximising the area under a curve, where the x and y axes correspond to *percentage of sketch* and $\frac{1}{rank_t}$ respectively. Maximising this area therefore requires the model to achieve early retrieval of the required photo.

Global Reward: During the initial steps of sketch rendering, the uncertainty associated with the sketch representation is high because an incomplete sketch could correspond to various photos (e.g., object outline with no details yet). The more it progresses towards completion, the representation becomes more concrete and moves towards one-to-one mapping with a corresponding photo. To model this observation, we use Kendall-Tau-distance [18] to measure the distance between two rank lists obtained from sequential sketching steps L_t and L_{t+1}. Kendall-Tau measures the distance between two ranking lists [32] as the number of pairwise disagreements (pairwise ranking order change) between them. Given the expectation of more randomness associated with early ambiguous partial sketches, the Kendall-Tau-distance between two successive rank lists from the initial steps of an episode is expected to be higher. Towards its completion, this value should decrease as the sketch

becomes more unambiguous. With this intuition, we add a regularizer that encourages the *normalised* Kendall-Tau-distance τ between two successive rank lists to be monotonically decreasing over a sketch rendering episode:

$$R_t^{Global} = -max(0, \tau(L_t, L_{t+1}) - \tau(L_{t-1}, L_t)) \quad (3)$$

This global regularisation reward term serves three purposes: (a) It models the extent of uncertainty associated with the partial sketch. (b) It discourages excessive change in the rank list later in an episode, making the retrieved result more consistent. This is important for user experience: if the returned top-ranked photos are changing constantly and drastically as the user adds more strokes, the user may be dissuaded from continuing. (c) Instead of simply considering the rank of the target, it considers the behaviour of the full ranking list and its consistency at each rendering step.

Training Procedure: We aim at maximising the sum of two proposed rewards

$$R_t = \gamma_1 \cdot R_t^{Local} + \gamma_2 \cdot R_t^{Global} \quad (4)$$

The RL literature provides several options for optimisation. The vanilla policy gradient [25] is the simplest, but suffers from poor data efficiency and robustness. Recent alternatives such as trust region policy optimization (TRPO) [37] are more data efficient, but involves complex second order derivative matrices. We therefore employ Proximal Policy Optimization (PPO) [38]. Using only first order optimization, it is data efficient as well as simple to implement and tune. PPO tries to limit how far the policy can change in each iteration, so as to reduce the likelihood of taking wrong decisions. More specifically, in vanilla policy gradient, the current policy is used to compute the policy gradient whose objective function is given as:

$$J(\Theta) = \hat{\mathbb{E}}_t [\log \pi_\Theta(a_t|s_t)R_t] \quad (5)$$

PPO uses the idea of importance sampling [28] and maintains two policy networks, where evaluation of the current policy $\pi_\Theta(a_t|s_t)$ is done by collecting samples from the older policy $\pi_{old}(a_t|s_t)$, thus helping in sampling efficiency. Hence, along-with importance sampling, the overall objective function written as:

$$J(\Theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\Theta(a_t|s_t)}{\pi_{old}(a_t|s_t)}r_t\right] = \hat{\mathbb{E}}_t [m_t(\Theta)R_t] \quad (6)$$

where $m_t(\Theta)$ is the probability ratio $m_t(\Theta) = \frac{\pi_\Theta(a_t|s_t)}{\pi_{old}(a_t|s_t)}$, which measures the difference between two polices. Maximising Eq. 6 would lead to a large policy update, hence it penalises policies moving $m_t(\Theta)$ away from 1, and the new clipped surrogate objective function becomes:

$$J(\Theta) = \hat{\mathbb{E}}_t [\min(m_t(\Theta)r_t, clip(m_t(\Theta), 1-\varepsilon, 1+\varepsilon)R_t)] \quad (7)$$

where ε is a hyperparameter set to 0.2 in this work. Please refer to [38] for more details. Empirically, we found the actor-only version of PPO with clipped surrogate objective to work well for our task. More analysis is given in Sec. 4.3.

4. Experiments

Datasets: We use QMUL-Shoe-V2 [29, 34, 40] and QMUL-Chair-V2 [40] datasets that have been specifically designed for FG-SBIR. Both datasets contain coordinate-stroke information, enabling us to render the rasterized sketch images at intervals, for training our RL framework and evaluating its retrieval performance over different stages of a complete sketch drawing episode. QMUL-Shoe-V2 contains a total of 6,730 sketches and 2,000 photos, of which we use 6,051 and 1,800 respectively for training, and the rest for testing. For QMUL-Chair-V2, we split it as 1,275/725 sketches and 300/100 photos for training/testing respectively.

Implementation Details: We implemented our framework in PyTorch [31] conducting experiments on a 11 GB Nvidia RTX 2080-Ti GPU. An Inception-V3 [42] (excluding the auxiliary branch) network pre-trained on ImageNet datasets [35], is used as the backbone network for both sketch and photo branches. In all experiments, we use Adam optimizer [17] and set $D = 64$ as the dimension of final feature embedding layer. We train the base model with a triplet objective having a margin of 0.3, for 100 epochs with batch size 16 and a learning rate of 0.0001. During RL based fine-tuning of sketch branch, we train the final g_μ layer of sketch branch (keeping f_θ and f_{att} fixed) with Σ for 2000 epochs with an initial learning rate of 0.001 till epoch 100, thereafter reducing it to 0.0001. The rasterized sketch images are rendered at $T = 20$ steps, and the gradients are updated by averaging over a complete sketch rendering episode of 16 different sketch samples. In addition to normalising the sampled action vector a_t, l_2 normalisation is also used after global adaptive average pooling layer as well as after the final feature embedding layer g_ϕ in image branch. The diagonal elements of Σ are initialised with 1, and γ_1, γ_2 and ε are set to 1, $1e - 4$ and 0.2 respectively.

Evaluation Metric: In line with on-the-fly FG-SBIR setting, we consider results appearing at the top of the list to matter more. Thus, we quantify the performance using Acc.@q accuracy, i.e. percentage of sketches having true-match photos appearing in the top-q list. Moreover, in order to capture the early retrieval performance, shadowing some earlier image retrieval works [19], we use plots of (i) *ranking percentile* and (ii) $\frac{1}{rank}$ versus *percentage of sketch*. In this context, a higher value of the mean area under the curve for (i) and (ii) signifies better early sketch retrieval performance, and we use m@A and m@B as shorthand notation for them in the rest of the paper, respectively.

4.1. Baseline Methods

To the best of our knowledge, there has been no prior work dealing with early retrieval in SBIR. Thus, based on some earlier works, we chose existing FG-SBIR baselines and their adaptations towards the new task to verify the contribution of our proposed RL based solution.

• **B1**: Here, we use the baseline model [41, 49] trained only with triplet loss. This basically represents our model (see Section 3.1) before RL based fine-tuning. • **B2**: We train a standard triplet model, but use all intermediate sketches as training data, so that the model also learns to retrieve incomplete sketches. • **B3**: We train 20 different models (as $T = 20$) for the sketch branch, and each model is trained to deal with a specific percentage of sketch (like 5%, 10%, ..., 100%), thus increasing the number of trainable parameters 20 times than the usual baseline. Different models are deployed at different stages of completion – again not required by any other compared methods. • **B4**: While RL is one of the possible ways of dealing with non-differentiable metrics, the recent work of Engilberge *et al.* [11] introduced a generalized deep network based solution to approximate non-differentiable objective functions such as ranking. This can be utilized in a plug-and-play manner within existing deep architectures. We follow a similar setup of cross-modal retrieval as designed by Engilberge *et al.* [11] and impose combination of triplet loss and ranking loss at T different instants of the sketch.

4.2. Performance Analysis

The performance of our proposed on-the-fly sketch based image retrieval is shown in Figure 4 against the baselines methods. We observe: (i) State-of-the-art triplet loss based baseline **B1** performs poorly for early incomplete sketches, due to the absence of any mechanism for learning incomplete retrieval. (ii) **B2**'s imposition of triplet-loss at every sketch-rendering step provides improved retrieval performance over **B1** for a few initial instants, but its performance declines towards the completion of sketch. This is mainly due to the fact that imposing triplet loss over incomplete sketches destabilises the learning process as it generates noisy gradients. In contrast, our RL based pipeline takes into account a complete sketch rendering episode along-with the associated uncertainty of early incomplete sketches before updating the gradients. (iii) Designing 20 different sketch models in **B3** for $T = 20$ different sketch rendering steps, improves performance towards the end of sketch rendering episode after 40% of sketch rendering in comparison to **B1**. However, it is poor for sketches before that stage due to its incompleteness which could correspond to various possible photos. (iv) An alternative of RL method – differential sorter described in **B4**, fares well against baseline **B1**, but is much weaker in comparison to our RL based non-differentiable ranking method. A qualitative result can
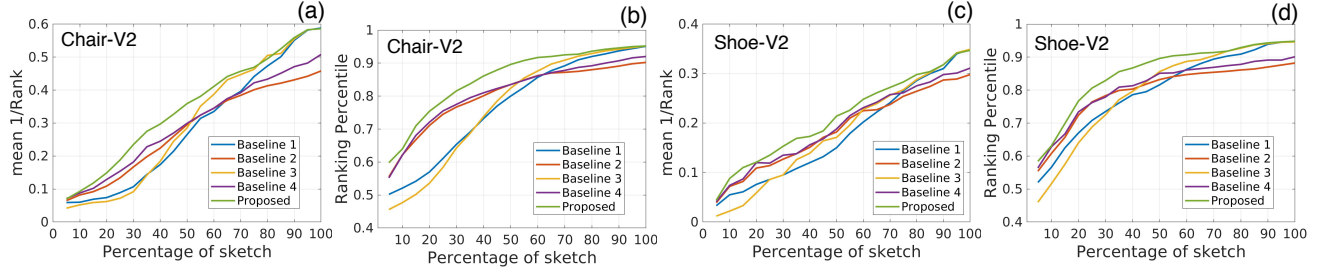
Figure 4. Comparative results. Note that instead of showing T=20 sketch rendering steps, we visualize it through percentage of sketch here. A higher area under these plots indicates better *early retrieval performance*.

be seen in Figure 2 where **B1** is the baseline.

In addition to the four baselines, following the recent direction of dealing with partial sketches [12, 22], we tried a two stage framework for our early retrieval objective referred as **TS** in Table 1. At any given drawing step, a conditional image-to-image translation model [15] is used to generate the complete sketch. Thereafter, it is fed to an off-the-shelf baseline model for photo retrieval. However, this choice of using an image translation network to complete the sketch from early instances, fails miserably. Moreover, it merely produces the input sketch with a few new random noisy strokes.

To summarise, our RL framework outperforms a number of baselines by a significant margin in context of early image retrieval performance as seen from the quantitative results in Figure 4 and Table 1, without deteriorating top-5 and top-10 accuracies in retrieval performance.

4.3. Ablation Study

Different RL Methods: We compare Proximal Policy Optimization (PPO), used here for continuous-action reinforcement learning, with some alternative RL algorithms. Although we intended to use a complete actor-critic version of PPO combining the policy surrogate Eq. 7 and value function error [38] term, using the actor-only version works better in our case. Additionally we evaluate this performance with (i) vanilla policy gradient [25] and (ii) TRPO [37]. Empirically we observe a higher performance with clipped surrogate objective when compared with the adaptive KL penalty (with adaptive co-efficient 0.01) approach of PPO. Table 2 shows that our actor-only PPO with clipped surrogate objective outperforms other alternatives.

Reward Analysis: In contrast to the complex design of efficient optimization approaches for non-differentiable rank based loss functions [26, 11], we introduce a simple reinforcement learning based pipeline that can optimise a CNN to achieve any non-differentiable metric in a cross modal retrieval task. To justify our contribution and understand the variance in retrieval performance with different plausible reward designs, we conduct a thorough ablative study. The positive scalar reward value is assigned to
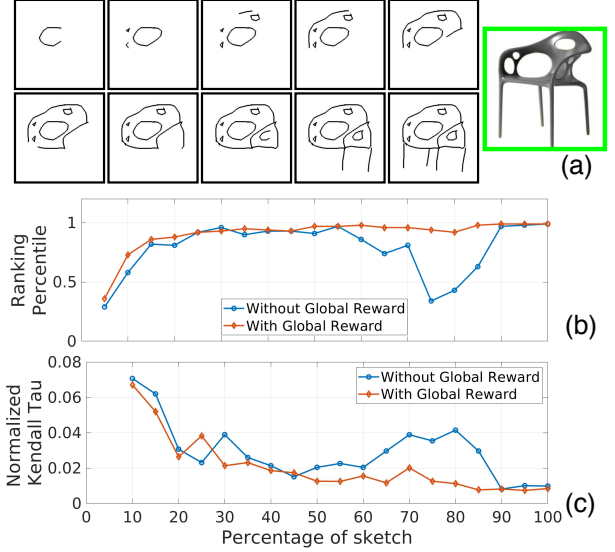

Figure 5. (a) Example showing progressive order of completing a sketch. (b) shows the drop in percentile whenever an irrelevant stroke is introduced while drawing (blue). (c) shows the corresponding explosive increase of Kendall-Tau distance signifying the percentile drop (blue). Our global reward term (red) nullifies these negative impacts of irrelevant sketch strokes thus maintaining consistency of the rank-list overall.

1 (otherwise zero), when the paired photo appears in top-q list. This q value could be controlled based on the requirements. Instead of reciprocating the rank value, taking its negative is also a choice. To address the concern that our inverse rank could produce too small a number, we alternatively evaluate the square root of reciprocal rank. From the results in Table 3, we can see that our designed reward function (Eq. 4) achieves the best performance.

Significance of Global Reward: While using our local reward (Eq. 2) achieves an excellent rank in early rendering steps, we noticed that the rank of a paired photo might worsen at a certain sketch-rendering step later on, as illustrated in Figure 5. As the user attempts to convey more fine-grained detail later on in the process, they may draw some noisy, irrelevant, or outlier stroke that degrades the performance. Our global-reward term in Eq. 4 alleviates this issue by imposing a monotonically decreasing constraint on

Table 1. Comparative results with different baseline methods. Here A@5 and A@10 denotes top-5 and top-10 retrieval accuracy for complete sketch (at t=T), respectively, whereas, m@A and m@B quantify the retrieval performance over a sketch rendering episode (see Section 4 for metric definition).

	Chair-V2				Shoe-V2			
	m@A	m@B	A@5	A@10	m@A	m@B	A@5	A@10
B1	77.18	29.04	**76.47**	88.13	80.12	18.05	65.69	**79.69**
B2	80.46	28.07	74.31	86.69	79.72	18.75	61.79	76.64
B3	76.99	30.27	**76.47**	88.13	80.13	18.46	65.69	79.69
B4	81.24	29.85	75.14	87.69	81.02	19.50	62.34	77.24
TS	76.01	27.64	73.47	85.13	77.12	17.13	62.67	76.47
Ours	**85.44**	**35.09**	76.34	**89.65**	**85.38**	**21.44**	**65.77**	79.63

Table 2. Results with different Reinforcement Learning (RL) methods, where A stands for actor-only version of the algorithm, and AC denotes the complete actor-critic design.

RL Methods	Chair-V2		Shoe-V2	
	m@A	m@B	m@A	m@B
Vanilla Policy Gradient	80.36	32.34	82.56	19.67
PPO-AC-Clipping	81.54	33.71	83.47	20.84
PPO-AC-KL Penalty	80.99	32.64	83.84	20.04
PPO-A-KL Penalty	81.34	33.01	83.51	20.66
TRPO	83.21	33.68	83.61	20.31
PPO-A-Clipping (Ours)	**85.44**	**35.09**	**85.38**	**21.44**

Table 3. Results with different candidate reward designs

Reward Schemes	Chair-V2		Shoe-V2	
	m@A	m@B	m@A	m@B
rank $\leq 1 \Rightarrow$ reward $= 1$	82.99	32.46	82.24	19.87
rank $\leq 5 \Rightarrow$ reward $= 1$	81.36	31.94	81.74	19.37
rank $\leq 10 \Rightarrow$ reward $= 1$	80.64	30.57	80.87	19.08
$-$rank	83.71	32.84	83.81	20.71
$\frac{1}{\sqrt{\text{rank}}}$	83.71	33.97	83.67	20.49
$\frac{1}{\text{rank}}$	84.33	34.11	84.07	20.54
Ours (Eq. 4)	**85.44**	**35.09**	**85.38**	**21.44**

the *normalised* Kendall-Tau-distance [18] between two successive rank lists over an episode (Figure 5). We quantify the identified adverse impacts of inconsistent strokes, via a new metric, termed **stroke-backlash index**. It is formulated as $\frac{\sum_{t=2}^{T} |min(RP_t - RP_{t-1}, 0)|}{T-1}$, where RP_t denotes the ranking percentile of paired photo at t^{th} sketch-rendering step which is averaged over all sketch samples in test split. Whenever a newly introduced stroke produces a decline in the ranking percentile, it is considered as a negative performance. Please note that *the lower the value of this index, the better will be the ranking list consistency performance*. We get a decline in stroke-backlash index from 0.0421 (0.0451) to 0.0304 (0.0337) in Chair-V2 (Shoe-V2) dataset when including the global reward. Furthermore, as shown in Table 3, this global reward term improves the early retrieval performance m@A and (m@B) by 1.11% (0.98%) and 1.31% (0.90%) for Chair-V2 and Shoe-V2 respectively. Instead of imposing the monotonically decreasing constraint over Kendall-Tau-distance that actually considers the relative ranking position of all the photos, we could have imposed the same monotonically decreasing constraint on the

Table 4. Performance on varying feature-embedding spaces

	Chair-V2			Shoe-V2		
	m@A	m@B	A@5	m@A	m@B	A@5
32	82.61	34.67	72.67	82.94	19.61	62.31
64	**85.44**	**35.09**	76.34	**85.38**	**21.44**	65.77
128	84.71	34.49	**78.61**	84.61	20.81	**67.64**
256	81.39	31.37	77.41	80.69	19.68	66.49

specific ranking of the paired-photo only. However, we notice that the stroke-backlash index arises to 0.0416 (0.0446) and overall m@A value decreases by 0.78% (0.86%) for Chair-V2 (Shoe-V2), thus justifying the importance of using Kendall-Tau distance in our context.

Further Analysis: (i) We evaluate the performance of our framework with a varying embedding space dimension in Table 4, confirming our choice of $D = 64$. (ii) Instead of using a standalone-trainable diagonal covariance matrix Σ for the actor network, we tried employing a separate fully-connected layer to predict the elements of Σ. However, the m@A and m@B performance deteriorates by 5.64% (4.67%) and 4.48% (3.51%), for Chair-V2 and (Shoe-V2) datasets respectively. (iii) In the context of on-the-fly FG-SBIR where we possess online sketch-stroke information, a reasonable alternative could be using a recurrent neural network like [4] for modelling the sketch branch instead of CNN. Following SketchRNN's [13] vector representation, a five-element vector is fed at every LSTM unit, and the hidden state vector is passed through a fully connected layer at any arbitrary instant, thus predicting the sketch feature representation. This alleviates the need of feeding a rendered rasterized sketch-image every time. However, replacing the CNN-sketch branch by RNN and keeping rest of the setup unchanged, performance drops significantly. As a result a top@5 accuracy of 19.62% (15.34%) is achieved compared to 76.34% (65.77%) in case of CNN for Chair-V2 (Shoe-V2) dataset. (iv) Different people have different stroke orders for sketching. Keeping this in mind we conducted an experiment by randomly shuffling stroke orders to check the consistency of our model. We obtain m@A and m@B values of 85.04% (34.84%) and 85.11% (20.92%) for Chair-V2 (Shoe-V2) datasets, respectively, demonstrating our robustness to such variations.

5. Conclusion

We have introduced a fine-grained sketch-based image retrieval framework designed to mitigate the practical barriers to FG-SBIR by analysing user sketches on-the-fly, and retrieve photos at the earliest instant. To this end, we have proposed a reinforcement-learning based pipeline with a set of novel rewards carefully designed to encode the 'early retrieval' scheme and stabilise the learning procedure against inconsistently drawn strokes. This provides considerable improvement on conventional baselines for on-the-fly FG-SBIR.

References

[1] Itamar Berger, Ariel Shamir, Moshe Mahler, Elizabeth Carter, and Jessica Hodgins. Style and abstraction in portrait sketching. *ACM TOG*, 2013. 1

[2] Tu Bui, Leonardo Ribeiro, Moacir Ponti, and John Collomosse. Deep manifold alignment for mid-grain sketch based image retrieval. In *ACCV*, 2018. 2

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

[4] John Collomosse, Tu Bui, and Hailin Jin. Livesketch: Query perturbations for guided sketch-based visual search. In *CVPR*, pages 2879–2887, 2019. 1, 2, 8

[5] John Collomosse, Tu Bui, Michael J Wilber, Chen Fang, and Hailin Jin. Sketching with style: Visual search with sketches and aesthetic context. In *ICCV*, 2017. 2

[6] 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, 2, 4

[7] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *ICML*, 2016. 4

[8] Chi Nhan Duong, Khoa Luu, Kha Gia Quach, Nghia Nguyen, Eric Patterson, Tien D Bui, and Ngan Le. Automatic face aging in videos via deep reinforcement learning. In *CVPR*, 2019. 3

[9] Anjan Dutta and Zeynep Akata. Semantically tied paired cycle consistency for zero-shot sketch-based image retrieval. In *CVPR*, 2019. 1, 2

[10] Martin Engilberge, Louis Chevallier, Patrick Pérez, and Matthieu Cord. Finding beans in burgers: Deep semantic-visual embedding with localization. In *CVPR*, 2018. 1

[11] Martin Engilberge, Louis Chevallier, Patrick Pérez, and Matthieu Cord. Sodeep: a sorting deep net to learn ranking loss surrogates. In *CVPR*, 2019. 6, 7

[12] 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. 2, 3, 7

[13] David Ha and Douglas Eck. A neural representation of sketch drawings. *ICLR*, 2017. 2, 3, 8

[14] Xiaoguang Han, Zhaoxuan Zhang, Dong Du, Mingdai Yang, Jingming Yu, Pan Pan, Xin Yang, Ligang Liu, Zixiang Xiong, and Shuguang Cui. Deep reinforcement learning of volume-guided progressive view inpainting for 3d point scene completion from a single depth image. In *CVPR*, 2019. 3

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 7

[16] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *JAIR*, 1996. 2, 3, 4

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[18] William R Knight. A computer method for calculating kendall's tau with ungrouped data. *Journal of the American Statistical Association*, 1966. 2, 5, 8

[19] Adriana Kovashka, Devi Parikh, and Kristen Grauman. Whittlesearch: Image search with relative attribute feedback. In *CVPR*, 2012. 6

[20] 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

[21] Xiaodan Liang, Lisa Lee, and Eric P Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *CVPR*, 2017. 3

[22] Fang Liu, Xiaoming Deng, Yu-Kun Lai, Yong-Jin Liu, Cuixia Ma, and Hongan Wang. Sketchgan: Joint sketch completion and recognition with generative adversarial network. In *CVPR*, 2019. 2, 3, 7

[23] Li Liu, Fumin Shen, Yuming Shen, Xianglong Liu, and Ling Shao. Deep sketch hashing: Fast free-hand sketch-based image retrieval. In *CVPR*, pages 2862–2871, 2017. 2

[24] Qing Liu, Lingxi Xie, Huiyu Wang, and Alan Yuille. Semantic-aware knowledge preservation for zero-shot sketch-based image retrieval. In *ICCV*, 2019. 2

[25] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016. 5, 7

[26] Pritish Mohapatra, Michal Rolinek, CV Jawahar, Vladimir Kolmogorov, and M Pawan Kumar. Efficient optimization for rank-based loss functions. In *CVPR*, 2018. 7

[27] Umar Riaz Muhammad, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Goal-driven sequential data abstraction. In *ICCV*, 2019. 3

[28] Radford M Neal. Annealed importance sampling. *Statistics and Computing*, 2001. 5

[29] 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. 1, 2, 6

[30] Kaiyue Pang, Yi-Zhe Song, Tony Xiang, and Timothy M Hospedales. Cross-domain generative learning for fine-grained sketch-based image retrieval. In *BMVC*, 2017. 2

[31] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NeurIPS Autodiff Workshop*, 2017. 6

[32] Daniel Carlos GuimarãEs Pedronette and Ricardo Da S Torres. Image re-ranking and rank aggregation based on similarity of ranked lists. *Pattern Recognition*, 2013. 5

[33] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017. 5

[34] Umar Riaz Muhammad, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning deep sketch abstraction. In *CVPR*, 2018. 3, 6

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy,

Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 6

[36] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *TOG*, 2016. 1, 2

[37] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, 2015. 5, 7

[38] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 5, 6, 7

[39] Yuming Shen, Li Liu, Fumin Shen, and Ling Shao. Zero-shot sketch-image hashing. In *CVPR*, 2018. 2

[40] Jifei Song, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning to sketch with shortcut cycle consistency. In *CVPR*, 2018. 6

[41] 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. 1, 2, 3, 4, 6

[42] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 4, 6

[43] Fang Wang, Le Kang, and Yi Li. Sketch-based 3d shape retrieval using convolutional neural networks. In *CVPR*, 2015. 2

[44] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *CVPR*, 2019. 3

[45] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 2009. 2, 4

[46] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 4

[47] Sasi Kiran Yelamarthi, Shiva Krishna Reddy, Ashish Mishra, and Anurag Mittal. A zero-shot framework for sketch based image retrieval. In *ECCV*, 2018. 2

[48] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018. 2

[49] 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, 3, 6

[50] Qian Yu, Yongxin Yang, Feng Liu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Sketch-a-net: A deep neural network that beats humans. *IJCV*, 2017. 1

[51] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *CVPR*, 2019. 2