Deep Reinforced Attention Regression for Partial Sketch Based Image Retrieval

Dingrong Wang, Hitesh Sapkota, Xumin Liu, Qi Yu
Rochester Institute of Technology
{dw7445, hxs1943, xumin.liu, qi.yu}@rit.edu

2021 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-2398-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICDM51629.2021.00078

***Abstract*—Fine-Grained Sketch-Based Image Retrieval (FG-SBIR) aims at finding a specific image from a large gallery given a query sketch. Despite the widespread applicability of FG-SBIR in many critical domains (*e.g.,* crime activity tracking), existing approaches still suffer from a low accuracy while being sensitive to external noises such as unnecessary strokes in the sketch. The retrieval performance will further deteriorate under a more practical on-the-fly setting, where only a partially complete sketch with only a few (noisy) strokes are available to retrieve corresponding images. We propose a novel framework that leverages a uniquely designed deep reinforcement learning model that performs a dual-level exploration to deal with partial sketch training and attention region selection. By enforcing the model's attention on the important regions of the original sketches, it remains robust to unnecessary stroke noises and improve the retrieval accuracy by a large margin. To sufficiently explore partial sketches and locate the important regions to attend, the model performs bootstrapped policy gradient for global exploration while adjusting a standard deviation term that governs a locator network for local exploration. The training process is guided by a hybrid loss that integrates a reinforcement loss and a supervised loss. A dynamic ranking reward is developed to fit the on-the-fly image retrieval process using partial sketches. The extensive experimentation performed on three public datasets shows that our proposed approach achieves the state-of-the-art performance on partial sketch based image retrieval.**

***Index Terms*—Sketch-based image retrieval, partial sketch matching, reinforcement learning**

I. Introduction

Due to the large amounts of touch-screen devices and their broad usage, sketch-related computer vision applications have drawn increasing attention nowadays. Consequently, a series of interesting research problems have emerged, including sketch recognition, sketch representation/interpretation, sketch generation, and sketch-based image retrieval (SBIR) [1]–[6]. Among them, SBIR, especially fine-grained SBIR, is of particular interest [4]–[6] because of its important applications in both commercial and public safety domains. For commercial usage, a customer can buy items online by simply drawing sketches of items using a smartphone screen. For the public safety applications, it provides additional support to track criminals by drawing their sketches based on the description and searching them in the police criminal database.

Category level SBIR is well studied in existing works [1], [4], [5]. In this technique, based on a query sketch, it retrieves images of the same category as the query. However, the domain gap between sketches and images remains as a key technical challenge. This is because a sketch captures only object shape/contour information, which does not contain other relevant fine-grained details in an image, such as color and texture [2]. There are three common ways to deal with this problem. First is sketch-image generation combining with image recognition [7]. But this method's performance depends heavily on the performance of sketch-image generation, where uncertainty is extremely high. The second approach is sketch-image hashing [8], where sketch and image are encoded by a deep neural network into hashing codes, then by calculating the Hamming distance between query sketch and each category's image hashing cluster, we can decide which cluster's images should be retrieved based on the query sketch. The last approach is finding a common sketch-image embedding layer [4], which aims to find a common hidden space to align both sketch and image embedding vectors. Although these methods can reach high classification accuracy, category level SBIR is usually not suitable in many real applications, where people expect to see a more precise retrieval result instead of a category of images. FG-SBIR provides a promising direction to fulfill this demand.

Compared to the category level SBIR, FG-SBIR can explore more fine-grained details from both sketch and image sides and get a more precise retrieval result. FG-SBIR is initially addressed in [9], which employed a deformable part-based model (DPM) representation and graph matching. However, the model performs poorly (in terms of retrieval accuracy) whenever the situation gets complicated (*e.g.,* adding more abstract sketches or images from unknown categories). To deal with more complicated settings, deep learning has been leveraged to tackle FG-SBIR, which aims to learn both feature representation and a cross-domain matching function jointly [2], [3], [10]. In this research area, there are broadly two lines of work: Siamese CNN network with a verification loss [11] and triplet CNN with a triplet ranking loss [12] [2] [1]. The difference in those works lies in the network structure and the image branch. A Siamese network takes extracted edge maps as input in the image branch for alignment with the sketch branch, while a triplet network uses the original image as input, which contains more information. The problem with these models is that to reach a good retrieval performance, a complete sketch is needed. In practice, like an online shopping app, people like to retrieve what they want in just a few strokes, rather than the whole detailed sketch. However, these models can only predict on the full sketch and have poor performance when generalizing to partial sketch cases. Simply

978-1-6654-2398-4/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDM51629.2021.00078

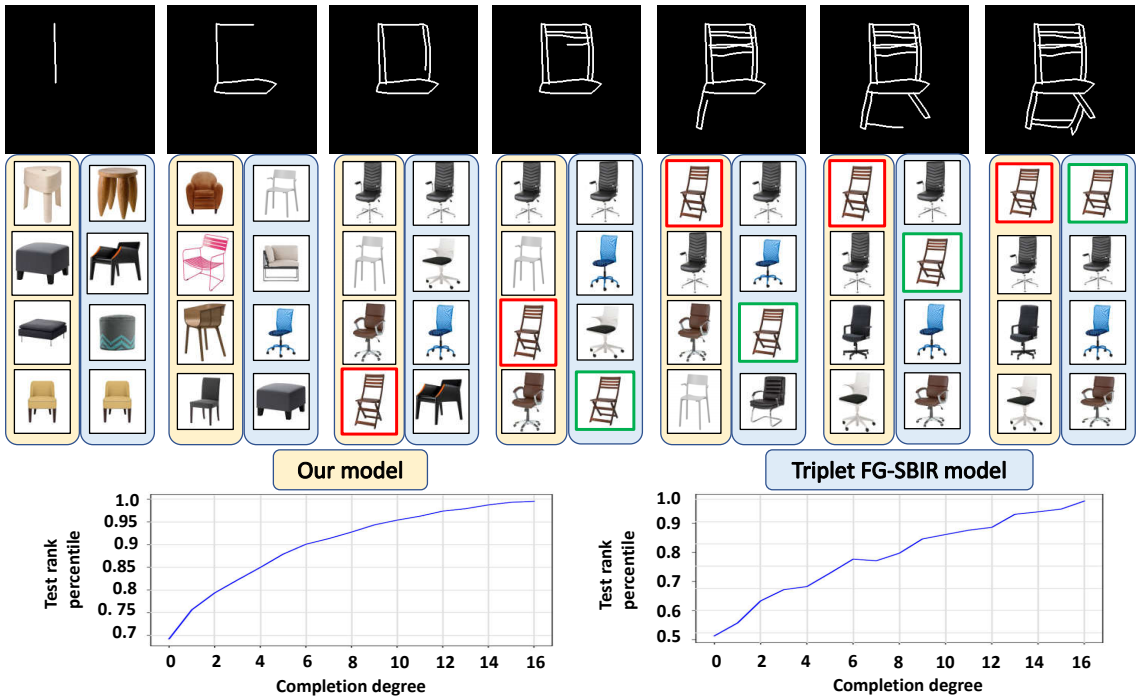


Fig. 1: Partial Sketch Retrieval Performance Comparison Between DARP-SBIR and a Triplet Network Model. The top row shows a sequence of partially complete sketches with increasing degree of completeness; the second row shows the corresponding retrieved list of images by DARP-SBIR (yellow) and Triplet network (blue); the last rows shows the ranking percentile of the matching image (*e.g.,* 0.9 means the matching image is ranked in the top 10% among all candidate images).

adding partial sketches into the training set is not applicable, as this jeopardizes the overall training performance.

Existing work to provide image retrieval based on partial sketches is fairly scarce. One exception is the model that combines a triplet network with a reinforcement learning (RL) agent to predict the embedding of a partial sketch on the fly [13]. In this model, a partial sketch is regarded as a state of a particular time step from a full episode. The reward signal here does not correspond to the final optimization objective adopted by supervised training. Instead, it belongs to the total cumulative reward in a whole episode, which is more natural for the on-the-fly setting. As a result, RL training for partial sketches has the potential to get better results than the supervised training. While the model shows a promising trend to support partial sketch based retrieval, the performance is still much lower than the full sketch based result. The result gets even worse when the completion degree of a partial sketch is low as the model becomes more sensitive to the noisy stokes.

To address the fundamental challenges as outlined above, we propose a novel framework that employs Deep reinforced Attention Regression for Partial Sketch Based Image Retrieval (DARP-SBIR). The proposed framework aims to deal with partial sketches while being robust to noisy stokes to further advance the state of the art in FG-SBIR and support the practical on-the-fly image retrieval. While a RL agent adopted by existing works can capture the temporal dynamics among different stokes in a sketch to predict the embedding of a final sketch [13], the incomplete sketches containing noisy strokes along with a potentially large image repository make the reward signal very sparse and delayed. The proposed attention mechanism aims to identify the most important regions in the sketch while ignoring the unnecessary noisy strokes. To tackle the sparse and delayed reward signals, the framework employs a novel bootstrapped policy gradient with dual-level exploration that performs a global deep exploration with a multi-head locator network and local exploration by dynamically adjusting the covariance of the Gaussian policy. The state space of the RL agent is maintained by a RNN network that aggregates both the attended location and the currently available sketch, The hidden state from the RNN can also be used to predict the sketch embedding to formulate a supervised learning loss, which is used to training the RNN (and other component networks in the framework) to more accurately capture the hidden state space. Finally, a dynamic ranking reward function is developed to allow the agent to pick up the relatively weak reward signal in the early phase and then focus on the higher reward once it is better trained.

Figure 1 compares the partial sketch retrieval performance between the proposed DARP-SBIR framework and a classical triplet network based model [1]. From the second row, we can see that with just a few stokes (third column), our model can already rank the matching image in its top-4 list. DARP-

SBIR also puts the matching image on the very top of its retrieval list multiple steps ahead of the Triplet network, which requires the complete sketch to do so. It is worth to note that only a subset of stokes are included from a whole rendering episode so the actual gap between these two models is actually even bigger. The bottom row quantitatively evaluate the partial sketch retrieval performance using the ranking percentile vs. completion degree of sketches (defined in Section IV). Our main contributions are summarised as follows:

- We design a novel framework that employs deep reinforced attention regression to support partial sketch based on-the-fly image retrieval.
- To handle sparse and delayed reward signals due to incomplete sketches and noisy stokes, the RL agent performs dual-level exploration that combines deep global exploration through bootstrapped policy gradient and local exploration through dynamically adjusted Gaussian covariance.
- The state space of the attention RL agent is maintained through a RNN that dynamically aggregates both the current attended region provided by the agent and the current available sketch.
- A hybrid loss function is developed that combines a dynamic ranking reward for training the RL agent and a supervised learning loss for updating the RNN.

Extensive experiments have been conducted on three public sketch datasets to show that our proposed DARP-SBIR framework achieves the state-of-the-art performance on both complete and partial sketch based image retrieval.

II. Related Work

A. Category Level SBIR

Category-level sketch-image retrieval has been extensively studied by existing works [4], [14]–[16]. There are three mainstream methodologies in the field of category-level SBIR: (1) sketch-image generation and image recognition, (2) sketch-image hashing, and (3) sketch-image common embedding. For the first group of methods, representative works like [17] use a sketch-RNN to perform sketch-image generation and then use image recognition to find the matching images, which can be unreliable because the sketch-RNN may provide poor multi-class sketch-image generation. In the second group of work, Liu et al [18] apply a CNN-RNN two branch model to deal with images and sketches, respectively. In their approach, a CNN extracts static information from images, while a RNN extracts temporal information from strokes. Finally, both sketches and images can be encoded into hashing vectors and SBIR is achieved by comparing the hamming distance between the query sketch and different image clusters. The most representative work in the third group mainly leverages a triplet network, where a triplet input consisting of one sketch, one positive image, and one negative image is fed into the network to transform into hidden vectors in the common embedding layer. Then, through a triplet loss, the sketch embedding vector should be close to the positive image embedding vector while moving away from the negative image's embedding vector, just like clustering in the hidden space [3], [4]. In this way, by performing a simple l_2 distance search in the hidden space with the query sketch's embedding vector, we can pick the closest image embedding as the matching image. Since the target is a set of images instead of a single image, category level SBIR may not be suitable for certain applications because its retrieval result is not precise. However, it provides a good starting point for FG-SBIR.

B. Fine-grained SBIR

Fine-grained SBIR is a more recent research area, which is less investigated than category level SBIR in the sketch analysis field, partly due to the challenge in data collection. Initially, it was addressed by a deformable part-based model representation and graph matching [9]. After that, a number of deep learning approaches have been applied to find the embedding space for both sketches and images [1]–[3], [12]. Two branches training with edge map and verification loss has been replaced by a triplet network with a ranking loss [2], [3]. Nowadays, FG-SBIR is researched using more complicated network structures and carefully designed losses. For example, Yu et al. [1] proposed a deep triplet-ranking model for instance-level FG-SBIR. Then, this paradigm is replaced by hybrid generative-discriminative cross-domain image generation [12], combined with a more sophisticated triplet loss. Song et al. [2] propose a model, which is also a triplet CNN. But with the introduced multi-scale coarse-fine semantic fusion and HOLEF loss, their model is much more effective than previous ones. Although existing models can reach better accuracy for complete sketches, they cannot readily generalize to incomplete ones, which is the focus of our model.

C. Reinforcement Learning based SBIR

While there has been much research effort in leveraging reinforcement learning (RL) [19] in various computer vision problems, RL is not commonly used for SBIR with few exceptions. Ayan et al. propose a partial sketch training procedure, where RL is leveraged to trade-off between sketch recognisability and the number of strokes to observe [13]. It introduces a discount weight and negative rewards for the number of strokes. Although the model is able to handle partial sketches, the retrieval performance is still lower comparing to the full sketch FG-SBIR models mentioned before. Further, the model itself is not robust to disturbing strokes. In our model, we use a novel attention mechanism combined with carefully designed dual-level exploration to improve the retrieval accuracy and avoid disturbing strokes by selecting important regions to feed into the network.

Effective deep exploration is crucial in RL tasks where the environment has a large action and state spaces. Various attempts have been made for effective deep exploration. Inspired from the idea of Thompson sampling [20], various posterior sampling based approaches have been proposed [21]–[23] to facilitate deep exploration. However, these techniques face the problem of inefficiency and intractability. To overcome the problems in previous approaches, randomized value function based approaches have been proposed. Based on this idea, both

linearly [24] as well as non-linearly [25] parameterized value functions are proposed. Relying on a non-linearly parameterized value function, Osband et al. [25] developed Bootstrapped DQN, which combines DQN and the bootstrapping technology. It performs well in multiple Atari games and boosts the performance significantly. However, the training time and computing resource cost pose a key bottleneck in a continuous action space, like attention regression in SBIR. To solve this problem while still leveraging the benefit of bootstrapping, we propose bootstrapped policy gradient training to alleviate the computation cost and further boost the performance through effective local exploration.

TABLE I: Notations with Descriptions

Notation	Description
$\mathbf{x}_i$	the i-th input partial sketch
$\mathbf{e}_i$	matching image embedding corresponding to $\mathbf{x}_i$
$\mathbf{g}_{i,t}$	glimpse vector at time step t for $\mathbf{x}_i$
$\mathbf{a}_{i,t}$	embedding vector of $\mathbf{x}_i$ at time step t
$\mathbf{v}_{i,t}$	glimpse location at time step t for $\mathbf{x}_i$
$\mathbf{h}_{i,t}$	hidden state vector at time step t for $\mathbf{x}_i$
$\boldsymbol{\omega}_{i,t}^k$	binary masks for $\mathbf{x}_i$ generated from locator head k at time step t
$R_{i,t}$	reward signal for $\mathbf{x}_i$ at time step t
η_t	threshold to adjust the dynamic ranking award
$\theta_g, \theta_h, \theta_a, \theta_v$	parameters of the glimpse, RNN, action, and locator networks
M	the number of total episodes
N	the number of partial sketches in one episode
T	the number of total time steps per episode
K	the number of locator heads

III. Methodology

We aim to develop a partial sketch based image retrieval framework that is capable of overcoming the well-known limitations for FG-SBIR: low retrieval accuracy and vulnerability to disturbing strokes. To fulfill these purposes and make our framework easy to adjust to partial sketches, we design a novel bootstrapped policy network, which enforces dynamic attention on the original sketch directly. Specifically, we divide the model training into two phases. In the first phase, we train the powerful backbone triplet Inception V3 model coupled with a triplet loss to get the target image embedding. In the second phase, we use the partial sketch based retrieval framework, which consists of a glimpse network, an action network, and a locator network to enforce dynamic visual attention on input sketches directly by selecting important attention regions. The selected attention location is processed by the glimpse network along with the current (partial) sketch, which will be sent to a RNN network to generate a hidden state vector. The state vector can be directly used by the action network to generate the sketch embedding. Meanwhile, the state is also used by the locator network to produce the updated attention location for the next time step. This dynamic attention mechanism is the core of our design, which could avoid noisy strokes from input sketches, as well as reduce the input dimensionality for more efficient training. Previously generated target image embedding is used to calculate the ranking reward and the reinforcement loss for bootstrapped policy gradient training to teach the model to look into the important regions. We further leverage a supervised loss to collectively train the glimpse, RNN, and action network to generate the sketch embedding as close to the ground truth image embedding as possible. Table I summarizes the major notations used throughout the paper.

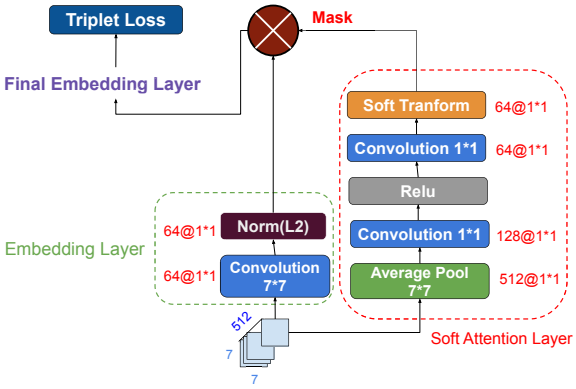


Fig. 2: Structure of the Embedding Network

A. Image Embedding Network

In order to get pre-trained ground-truth image embedding vectors, we first train a triplet network consisting of three branches with a shared set of parameters. The input is passed in the form of anchor, positive and negative samples. The goal is to minimize the distance between the anchor and the positive samples while maximizing its distance from the negative sample. In our case, the anchor is a sketch, the positive sample is an image corresponding to that sketch, called the ground truth image, while the negative sample is a random image not corresponding to that sketch. For the network structure, as Figure 2 shows, we add a fully connected embedding layer with 64 neurons on the top of the feature extractor (which is the backbone model of Resnet or InceptionV3) to get the embedding vector. We then normalize the output embedding vector in l_2 form. On the other side, we use a soft attention layer to extract effective part for that embedding vector to form a final embedding vector to be utilized by the main framework. For network training, we apply a triplet loss, which is formulated as follows. Define a triplet $\{a, p, n\}$, where a is anchor, p is a positive example, n is a negative example. Assume that a and p are from the same category, while n is not. Let $\psi(\cdot)$ and $\phi(\cdot)$ denote the embedding vectors for a sketch and image, respectively. Then, l_2 distance is used to calculate their difference, where δ_+ and δ_- denote the distances to the positive and negative instances, respectively. Finally, the triplet loss is defined as

$$\mathcal{L}_{tri} = \frac{1}{N}\sum_{n=1}^{N} \max\{0, \mu + \delta_+^n - \delta_-^n\} \tag{1}$$

where μ is a hyper-parameter that defines a margin.

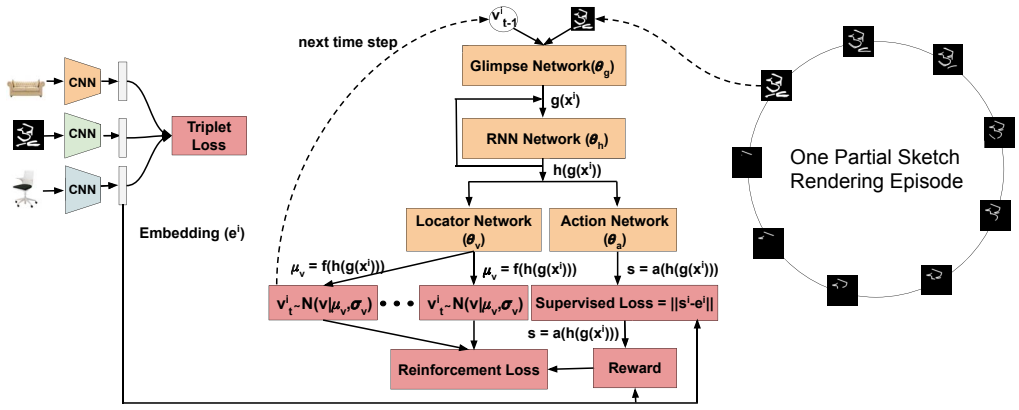


Fig. 3: Overall Architecture of DARP-SBIR

B. Deep Reinforced Attention Regression Network

In this section, we present the architecture (*see* Figure 3) of the proposed deep reinforced attention regression network for partial sketch based image retrieval (DARP-SBIR), which includes four key components: a glimpse network, a RNN network, an action network, and a locator network. We describe the first three components in this section and present the locator network next section that focuses on its unique dual-level exploration for attention regression.

Glimpse network. As Figure 4 shows, the glimpse network takes the original sketch $\mathbf{x}_i$ and a glimpse location vector $\mathbf{v}_{i,t-1}$ as input and extracts the information of the retina-like representation $\rho(\mathbf{x}_i, \mathbf{v}_{i,t-1})$ around location $\mathbf{v}_{i,t-1}$ from sketch $\mathbf{x}_i$. It encodes the region around $\mathbf{v}_{i,t-1}$ at a high resolution but uses a progressively lower resolution for pixels farther from $\mathbf{v}_{i,t-1}$, resulting in a vector of lower dimensionality than the original sketch $\mathbf{x}_i$. The term *glimpse* is to intuitively capture the combination of these high and low-resolution representations. The retina representation and glimpse location are then mapped into a hidden space using independent linear layers parameterized by $\theta_g = (\theta_g^0, \theta_g^1; \theta_g^2, \theta_g^3)$. With concatenation followed by rectified units, the glimpse network combines both representations to produce a final representation $\mathbf{g}_{i,t}$ to be used by the RNN network described next.

RNN network. The RNN network takes the current time step's glimpse vector $\mathbf{g}_t$ and the previous hidden state vector $\mathbf{h}_{t-1}$ as input, and outputs the current time step hidden state vector $\mathbf{h}_t$. Here, the hidden state vector encodes the model's knowledge of the environment and is instrumental in deciding how to act and where to deploy the glimpse sensor. The initial hidden vector is a random 256-dimensional embedding vector. The forward message of RNN is produced as follows: $\mathbf{h}_{i,t} = \text{RNN}(\mathbf{h}_{i,t-1}, \mathbf{g}_{i,t}; \theta_h)$.

Action Network. The action network takes the hidden state $\mathbf{h}_t$ of the current time step as input and predicts the predicted final sketch representation: $\mathbf{a}_{i,t} = \text{ACTN}(\mathbf{h}_{i,t}; \theta_a)$. The predicted sketch representation is then used to calculate the distance from the corresponding image embedding vector as the supervised loss function: $\mathcal{L}^s_{i,t} = \|\mathbf{a}_{i,t} - \mathbf{e}_i\|^2$. It is worth to note that sketch representation is predicted using the current hidden state $\mathbf{h}_{i,t}$ from the RNN network, which has aggregated the previous hidden state and the current glimpse vector $\mathbf{g}_{i,t}$. Recall that the glimpse vector is computed from the currently attended region predicted by the located network (which will be described next) and the already completed partial sketch.

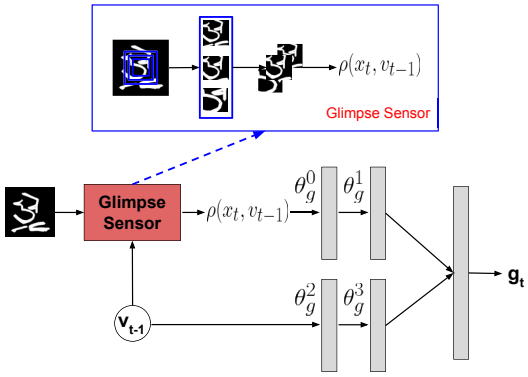


Fig. 4: The Glimpse Network

C. Dynamic Attention Regression via Dual-Level Exploration

Basic locator network. The basic locator network takes the current hidden state $\mathbf{h}_{i,t}$ and maps it to a location that indicates an important region of the sketch that the model should attend to. To accommodate the on-the-fly retrieval of matching images where stokes may continue to be collected to make a partial sketch more complete, we propose to achieve dynamic attention regression through deep reinforcement learning. In particular, the basic locator network can be modeled as a policy network that learns a stochastic policy $\pi(\mathbf{v}_{i,t}|\mathbf{h}_{i,t}; \theta_v)$, where $\mathbf{v}_{i,t} \in \mathbb{R}^2$ contains the (x, y) coordinates of the predicted location. The predicted location will then be used by the glimpse network to form the glimpse vector for the next time step along with the available partial sketch at that time step. Finally, the RNN will take the glimpse vector to

transition (the environment) to the next hidden state $\mathbf{h}_{i,t+1}$. Since the action space corresponds the locations in the sketch image, which is continuous, we adopt a Gaussian policy:

$$\mathbf{v}_{i,t} \sim \mathcal{N}(\text{LOCN}(\mathbf{h}_{i,t}; \theta_v), \Sigma) \tag{2}$$

where $\text{LOCN}(\mathbf{h}_{i,t}; \theta_v)$ is the mean location predicted based on the current hidden state and $\Sigma = \text{diag}(\sigma_x, \sigma_y)$. We will discuss how to set (σ_x, σ_y) dynamically to achieve effective local exploration.

Dynamic ranking reward. Since the ultimate goal of the sketch-based image retrieval is to rank the matching image higher than other irrelevant images, it is most appropriate to use a ranking based reward function. More intuitively, the reward should be large if the matching image is ranked high and small otherwise. Capturing this notion, we define the reward signal as follows:

$$R_{i,t} = \mathbb{1}(\text{score}_{i,t} \geq \eta_t) \tag{3}$$

where $\mathbb{1}(\text{condition})$ is an indicator function taking value 1 if condition is true and 0 otherwise and η_t is a dynamically adjusted threshold. The score of the matching image i is defined as the reciprocal of its ranking among all images:

$$\text{score}_{i,t} = \frac{1}{\sum_{j=1}^{N} \mathbb{1}(\|\mathbf{e}_j - \mathbf{a}_{i,t}\|_2^2 \leq \|\mathbf{e}_i - \mathbf{a}_{i,t}\|_2^2)} \tag{4}$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ are the outputs of the image embedding network for images i and j, respectively, $\mathbf{a}_{i,t}$ is the predicted final sketch representation by the action network at time step t, and N is the total number of images available in the training set. In the early stage of a training episode, it is desirable to keep the threshold η_t less strict, which allows the model to pick up the relatively weak reward signal. As the training progresses and the model gets better, the threshold is increased to encourage the model to seek for a higher reward. To capture this notion, we define an adaptive threshold that depends on an training steps as follows:

$$\eta_t = \frac{1}{1 + \exp(-\alpha \times t + \beta)} \tag{5}$$

where α and β are hyperparameters.

Policy gradient. Based on the reward signal defined above, the return at time-step t is computed as the total discounted reward from t:

$$G_{i,t} = \sum_{\hat{t}=1}^{T-\hat{t}} \gamma^{\hat{t}-1} R_{i,t+\hat{t}} \tag{6}$$

where $0 < \gamma \leq 1$ is a discount factor that assigns a higher reward at an earlier stage, which has the effect to encourage the discovery of the matching image with a less complete partial sketch. Finally, we define the objective function for the basic locator network as the expected return (which is the state value) at time step t:

$$\mathcal{L}_{i,t}(\theta_v) = \mathbb{E}_{p(\mathbf{h}_{i,t:T}; \theta_v)} \left[\sum_{\hat{t}=0}^{T-t} \gamma^{\hat{t}} R_{i,t+\hat{t}} \right] \tag{7}$$

where the hidden state distribution depends on the policy $\pi(\mathbf{v}_{i,t}|\mathbf{h}_{i,t}; \theta_v)$. Since the dynamics among the hidden environment states $\mathbf{h}_{i,t}$ is unknown, the expectation on the r.h.s. of (7) cannot be analytically computed. Thus, we perform Monte Carlo sampling to approximate the policy gradient:

$$\begin{aligned} \nabla_{\theta_v} \mathcal{L}_i(\theta_v) &= \sum_{t=1}^{T} \mathbb{E}_{p(\mathbf{h}_{i,t:T}; \theta_v)} \left[\nabla_{\theta_v} \log \pi(\mathbf{v}_{i,t}|\mathbf{h}_{i,t}) G_{i,t} \right] \\ &\approx \frac{1}{M} \sum_{m=1}^{M} \sum_{t=1}^{T} \nabla_{\theta_v} \log \pi(\mathbf{v}_{i,t}^{(m)}|\mathbf{h}_{i,t}^{(m)}) G_{i,t}^{(m)} \end{aligned} \tag{8}$$

where $\mathbf{v}_{i,t}^{(m)}, G_{i,t}^{(m)}$ are samples collected during episode m for partial sketch i at time step t where $m \in [M]$.

Dual-level exploration. The partial sketch coupled with a potentially large image repository could make the reward signal very sparse and delayed. As a result, effective exploration is critical to ensure fast and better convergence for training the locator network. The recently developed Bootstrapped DQN model has been demonstrated to be a promising exploration strategy in discrete action space scenarios [25]. In particular, Bootstrapped DQN approximates the distribution over Q values via a bootstrapping strategy to facilitate the deep and effective exploration. However, direct applying bootstrapped DQN to our SBIR framework faces two key challenges. First, it requires training two networks, including the Q network for value approximation and a policy network to update the policy. Second, our action space is continuous, choosing the best action also requires solving an optimization problem. As a result, these challenges increase both the computational time and the difficulty of the training process, making it less suitable for the on-the-fly image retrieval.

To deal with the key limitations presented above, we propose a dual level exploration technique that combines lightweight bootstrapped policy gradient (BPG) training for global deep exploration and adaptive local exploration through dynamically adjusting the variance of the Gaussian policy and the ranking reward. In particular, the BPG leverages a bootstrapped locator network with the K heads to predict the glimpse location. In each training episode, it randomly chooses one from K candidate heads to predict the next location for the entire episode. To achieve the bootstrapping behavior, the experience $\{\mathbf{h}_{i,t}, \mathbf{v}_{i,t}, R_{i,t+1}, \mathbf{h}_{i,t+1}\}$ is recorded along with a mask $\boldsymbol{\omega}_{i,t} \in \{0,1\}^K$ that is sampled from a mask distribution Ω. This experience is used to update the k-th locator network only when $\omega_{i,t}^k = 1$. As a result, different heads will be trained using sufficiently different experiences to ensure the bootstrapping property. The proposed BPG offers three key advantages. First, by maintaining K heads for the locator network and letting one head to execute one entire episode, it effectively achieves multi-step (or deep) exploration suitable for partial sketch matching with very sparse and delayed reward signals. Second, each head is trained using the bootstrapped experiences instead of from the same pool of experiences to effectively reduce the variance and improve the prediction accuracy. Last, by directly performing policy

gradient, it avoids solving two optimization problems as in deep Q-learning to improve the efficiency when dealing with a continuous action space.

In addition to the global deep exploration using the K heads, we equip each head with a local exploration capacity to ensure fast and better convergence of model training. In particular, the covariance matrix $\Sigma = \text{diag}(\sigma_x, \sigma_y)$ of the Gaussian policy in (2) plays an important to locate important regions in a sketch for the model to attend to. Within each head, if both σ_x and σ_y are set to very small, the head may be concentrated on a very small neighborhood and only performs exploitation without exploring other nearby regions. To perform effective local exploration, we propose to assign relatively large variances along with each coordinate in the early steps in the training and gradually shrink them along with the training process. Meanwhile, as shown in (5), the reward signal is also dynamically adjusted to allow the head to pick up a relatively weak signal in the early phase of training and then focus on higher rewards as the head becomes better trained. Our experimental results clearly justify the effectiveness of the dual-level exploration strategy.

D. The Training Process

For training, we use both supervised loss coupled with bootstrapped policy gradient loss. Given the input sketch $\mathbf{x}_i$ and glimpse location vector $\mathbf{v}_{i,t-1}$ from time step $t-1$, we use the glimpse network to determine the glimpse representation $\mathbf{g}_{i,t}$. Next, in the RNN network, we pass $\mathbf{g}_{i,t}$ along with the previous hidden state representation $\mathbf{h}_{t-1}$ to produce the current hidden state vector $\mathbf{h}_t$. The action network then takes $\mathbf{h}_t$ and produces the corresponding sketch representation $\mathbf{a}_{i,t}$. Using this representation, we compute the distance from the corresponding image embedding vector as the supervised loss.

On the other hand, given $\mathbf{h}_t$, the locator network with K heads produces the next location $\mathbf{v}_{i,t}$ to be attended. Assume that the k-th head is selected to execute the current episode. The experience $\{\mathbf{h}_{i,t}, \mathbf{v}_{i,t}, R_{i,t+1}, \mathbf{h}_{i,t+1}\}$ is recorded along with a mask $\boldsymbol{\omega}_{i,t}^k$ to ensure the bootstrapping behavior during model training. Again, the network finds the glimpse considering the new location and repeats the process until the episode complete. It should be noted that we randomly choose the one k head from K candidate heads to predict the next location and it is fixed for the entire episode. We repeat the process for M episodes. Finally, we update the locator network using (8). Next, we use the gradient of supervised loss to update the action network, RNN network, and glimpse network. More detailed training process is demonstrated in Algorithm 1.

Algorithm 1 Training Process for DARP-SBIR Framework

Input: Training sketch set $\mathbf{X}$, initial hidden state vector $\mathbf{H}_0$, initial glimpse location $\mathbf{V}_0$, pre-trained image embedding $\mathbf{E}$, total episodes M, Time step per episode T
Output: Updated network parameterrs $\theta_g, \theta_h, \theta_v, \theta_a$

initialize supervised loss $\mathcal{L}^s$
for $m = 0; m \leq M; m \leftarrow m + 1$ **do**
  Sample $k \sim [1, .., K]$
  for $i = 0; i \leq N; i \leftarrow i + 1$ **do**
    Sample $\mathbf{x}_i \sim \mathbf{X}$
    Initialize: $\mathbf{v}_{i,0}^{(m)}$, $\mathbf{h}_{i,0}^{(m)}$
    for $t = 0; t \leq T; t \leftarrow t + 1$ **do**
      Obtain retina representation $\rho(\mathbf{x}_i, \mathbf{v}_{i,t}^{(m)})$
      Obtain $\mathbf{g}_{i,t}^{(m)}$ from Glimpse Network $(\rho(\mathbf{x}_i, \mathbf{v}_{i,t}^{(m)}); \theta_g)$
      Obtain $\mathbf{h}_{i,t+1}^{(m)}$ from $RNN(\mathbf{g}_{i,t}^{(m)}, \mathbf{h}_{i,t}^{(m)}; \theta_h)$
      Obtain $\mathbf{a}_i^{(m)}$ from $ACTN(\mathbf{h}_{i,t+1}^{(m)}; \theta_a)$
      Compute supervised loss and add to $\mathcal{L}^s$
      Compute $\mathcal{R}_{i,t+1}^{(m)}$ using (3)
      Compute $\mathbf{v}_{i,t}^{(m)}$ using locator network θ_v
      Store experience $\{\mathbf{h}_{i,t}^{(m)}, \mathbf{v}_{i,t}^{(m)}, \mathcal{R}_{i,t+1}^{(m)}, \mathbf{h}_{i,t+1}^{(m)}\}$
    end for
  end for
end for
Update θ_v using (8)
Update θ_a, θ_h, and θ_g using the supervised loss $\mathcal{L}^s$
Repeat step 1∼20

IV. Experiments

In this section, we present our experimental results on three real-world sketch datasets to demonstrate: (1) the state-of-the-art performance in partial sketch based image retrieval, (2) effectiveness of attention regression through deep reinforcement learning, and (3) how the novel dual-level exploration helps to achieve better and faster convergence of the locator network that boosts the entire image retrieval performance.

A. Datasets

We include three commonly used sketch datasets in our evaluation. The first two datasets, including QMUL-Shoe-V2[1] [3], [26], [27] and QMUL-Chair-V2[2] [26], are specifically designed for FG-SBIR. Both datasets contain coordinate stroke information, enabling us to render the rasterized sketch images at intervals for training our DARP-SBIR framework and evaluate its retrieval performance over different stages of a complete sketch drawing episode. In our pre-processing step, we split the whole sketch drawing process into 17 partial sketches, using openCV's image dilation function to thicken the strokes, and leveraging the flipping and rotating function to perform data augmentation. We further include another publicly available fine-grained sketch-image dataset, Sketchy[3] [11]. We select 5 categories from this dataset to test our framework's retrieval performance. Table II summarizes the key properties of all three datasets.

B. Comparison Baselines

We include four competitive comparison baselines, where the first model primarily relies on a triplet network for both images and sketches, two other models leverage relatively simple reinforcement leaning models to support partial sketch matching, and the last model that adapts the bootstrapped

[1] http://sketchx.eecs.qmul.ac.uk/downloads/
[2] http://sketchx.eecs.qmul.ac.uk/downloads/
[3] https://sketchy.eye.gatech.edu/

TABLE II: Description of Datasets

Dataset	Sketch			Image		
Description	*Train*	*Test*	*Total*	*Train*	*Test*	*Total*
ChairV2	725	275	1000	300	100	400
ShoeV2	6051	679	6730	1800	200	2000
Sketchy	2500	500	3000	415	85	500

DQN into our proposed framework. We present the details of these models below:

- *Triplet network:* The triplet network outputs embedding vectors for both images and sketches [1], [2]. By projecting both images and sketches onto the embedding space, they can be directly matched through commonly used similarity or distance metrics.
- *Triplet + Vanilla RL:* To support partial sketch matching, this model combines the triplet network for image embedding with a simple policy network [28] to generate the final sketch representation , which can then be matched with the image embedding for retrieval.
- *Triplet + PPO:* In this model [13], the simple policy network is replaced by an improved version of the PPO network [29]. The PPO network is trained with the replay memory. The value network is trained by Monte-Carlo method and the policy network is trained by policy gradient.
- *Bootstrapped DQN:* To demonstrate the effectiveness of the proposed deep attention model with dual-level exploration, we implement a Bootstrapped DQN [25] version of the proposed framework by keeping all other network fixed but replacing the locator network with a Bootstrapped DQN for location regression.

C. Experimental Settings

We conduct a grid search to set the value of some related parameters in the proposed framework. In particular, we choose glimpse patches with a size of 8×8 from the original $64 \times 64 \times 1$ input sketch and the number of glimpses per sketch is 12. The dimension for hidden glimpse vector $\mathbf{g}_{i,t}$ is 128, the dimension for hidden state vector $\mathbf{h}_{i,t}$ is 256, the sketch embedding dimension $\mathbf{a}_{i,t}$ is 64, same as the dimension of pre-trained image embedding $\mathbf{e}_i$. For the dynamic threshold given in (5), α is set as 0.02 and β as -2 (other values in a similar range also work well). The number of candidate heads in the bootstrapped policy network is set as $K = 6$. The number of partial sketches in one episode is typically 17. We train the model for 1,000 epochs to get desired results. Early stop is possible if the model doesn't improve the validation accuracy for a long time. The learning rate is set to be 3×10^{-4} and the resolution scale parameter is set to 1. All the experiments are conducted on a 4-core Intel i7 CPU and one V100 GPU card with CUDA 10.2 and PyTorch 1.6.0. Our implementation source code can be found using this link[4].

[4]https://github.com/wdr123/DARP-SBIR

D. Evaluation Metrics

To properly evaluate the FG-SBIR performance under the on-the-fly setting, we consider two categories of evaluation metrics by following existing works in this area. In particular, to evaluate the retrieval performance based on complete sketches, we consider the images that appear at the top of the retrieved list as those that matter more. Therefore, our first metric evaluates the percentage of complete sketches with their true matching images appearing in the top-q list, which is referred to as $acc@q$. To assess the retrieval performance on partial sketches, the metric should encourage early retrieval [13]. Thus, we compute the mean area under the plot of $1/\text{rank}$ versus the completion degree of a sketch, referred to as area under inverse rank or short for AUIR. We further consider the ranking percentile: $(N - \sum_{j=1}^{N} \mathbb{1}(||\mathbf{e}_j - \mathbf{a}_{i,t}||_2^2 \leq ||\mathbf{e}_i - \mathbf{a}_{i,t}||_2^2))/N$, versus the completion degree of a sketch as another metric for partial sketch performance evaluation.

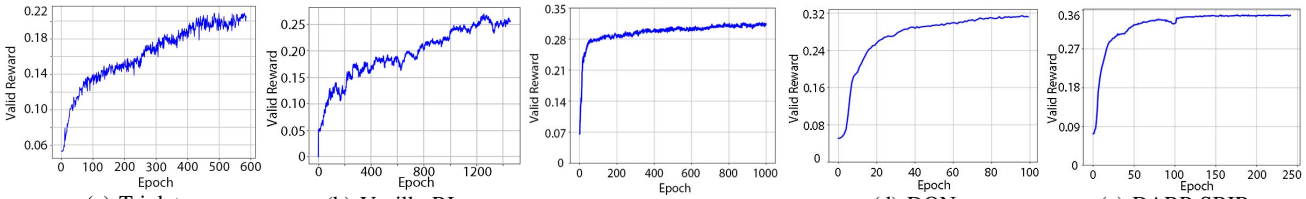
E. Comparison Results

For the comparison within different baselines, we report the performance on both partial and complete sketch based retrieval using AUIR and $acc@5$ in Table III.

Complete sketch result. First, for the complete sketch retrieval, the training curves of our model and other baselines on the ChairV2 dataset are shown in Figure 5. The proposed model outperforms all the baselines with a clear margin. The second best model is the Triplet + PPO model that benefits from an advanced PPO network [29]. However, no attention mechanism is provided, making it less robust to the noisy stokes. The Bootstrapped DQN based SBIR also performed worse than our framework. This could be due to the sub-optimal results obtained when optimizing over a continuous action space. For ChairV2 dataset, we leverage a pre-trained sketch embedding vector from [29] to initialize the RNN hidden state vector $\mathbf{h}_0$. For the other two datasets, we use a self-generated sketch embedding to initialize $\mathbf{h}_0$. Thus, all the baseline models are compared under the same condition in each dataset.

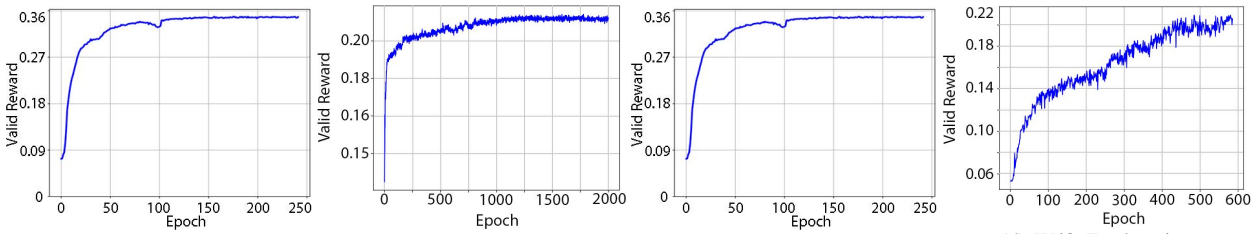
Partial sketch result. For partial sketch retrieval, we use AUIR, which encourage early retrieval using less complete partial sketches. Benefiting from the deep reinforced attention regression mechanism, the proposed framework again outperforms all the baselines on all three datasets. The Triplet network, which is not designed for the on-the-fly retrieval, performs the worst in most cases.

Figure 7 further compares DARP-SBIR with the second best model, Triplet + PPO, on partial sketch retrieval through ranking percentile vs. complete degree of a sketch by using ChairV2 as an example. As can be seen, our model only needs six out of sixteen strokes to reach the top-10 ranking percentile while the Triplet + PPO needs more than 12 strokes to reach the same ranking. Performance on the other two datasets shows similar trends, which are omitted due to lack of space.



(a) Triplet (b) Vanilla RL (c) PPO (d) DQN (e) DARP-SBIR

Fig. 5: Comparison Experiment

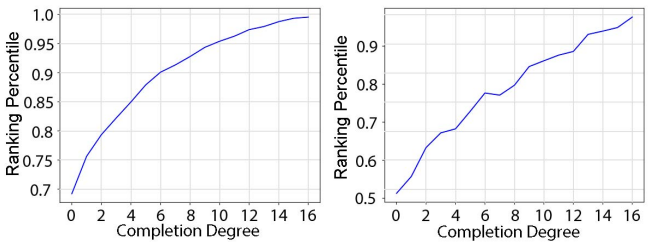


(a) With Attention (b) W/O Attention (c) With Exploration (d) W/O Exploration

Fig. 6: Effectiveness of Attention Regression and Dual-Level Exploration

TABLE III: Performance Comparison

Comparison	Chair V2		Shoe V2		Sketchy	
Model	*AUIR*	*acc@5*	*AUIR*	*acc@5*	*AUIR*	*acc@5*
Triplet network	21.05	64.47	12.05	52.34	25.00	69.50
Triplet + Vanilla RL	25.34	69.67	13.60	54.80	24.30	68.70
Triplet + PPO	33.35	75.44	15.50	57.10	27.88	71.24
Bootstrapped DQN	30.05	71.45	13.50	54.50	24.50	69.00
DARP-SBIR	**35.65**	**79.12**	**18.12**	**60.02**	**32.32**	**73.82**



(a) DARP-SBIR (b) Triplet + PPO

Fig. 7: Ranking Percentile Vs. Complete Degree (ChairV2)

F. Ablation Study

We conduct an ablation study to evaluate the effectiveness of attention regression and the dual-level exploration used by the bootstrapped policy network.

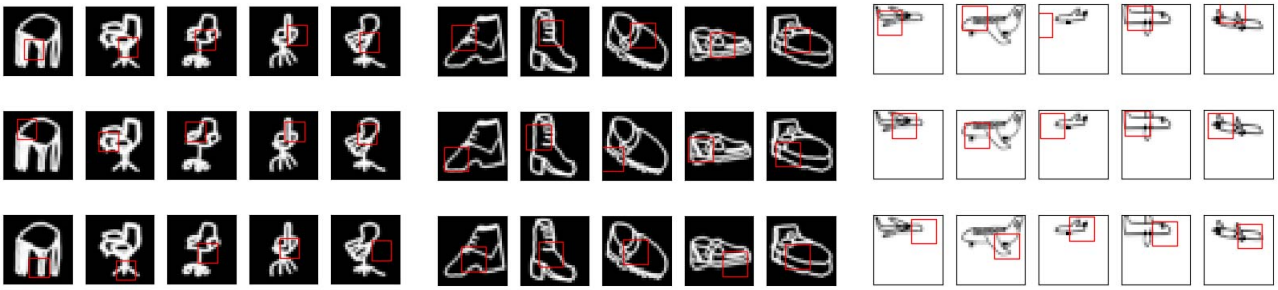
First, we compare the validation rewards of our model using two variants, with attention regression and without attention regression by using ChairV2 as an example. It is worth to note that without attention regression, both the glimpse network and the locator network will be removed from the framework. The RNN network is used to capture the temporal dynamics in the on-the-fly setting and the action network is to predict the final sketch representation. Figure 6 (a-b) shows that the attention mechanism plays an important role in the overall retrieval performance by helping the framework identify the most important regions to attend to while ignoring the noisy stokes. Comparing to the model without attention selection and only using supervised loss, the model with attention mechanism gains around 75% performance boost. Also, the model with attention mechanism could reach their maximum performance in relatively shorter time.

We further compare our model's performance with and without the dual-level exploration. Figure 6 (c-d) shows that exploration is very important in performance boosting and reducing the training time. One thing to note, if a model only uses attention region selection while not applying sufficient exploration, then its performance is similar to not using attention region selection at all. This clearly demonstrate importance of exploration in partial sketch based image retrieval that involves very sparse and delayed reward signals.

Finally, Figure 8 shows the corresponding glimpse results for the three datasets to highlight the importance of local exploration (in combination with the global deep exploration). As shown, after proper training, our model is able to focus on the important regions of the sketches and therefore, ignoring the unnecessary strokes drawn. The attended regions usually cover a wide area in the sketches, partly due to the effect of dynamically adjusting the covariance in the Gaussian policy and the reward signal.

V. Conclusions

In this work, we propose a novel on-the-fly FG-SBIR framework that takes attention regions directly from an original sketch to achieve high sketch-based image retrieval performance. To deal with partial sketches and provide correct predictions as early as possible, it is trained with a



(a) ChairV2 (b) ShoeV2 (c) Sketchy

Fig. 8: Glimpse Results

bootstrapped policy network that performs novel dual-level exploration. To deal with the sparse and delayed reward signals in partial sketch based retrieval, we combine bootstrapping with a K-head locator network with effective local exploration strategies, including using an adaptive Gaussian covariance and dynamically adjusting the reward threshold. As a result, our framework can enforce an attention mechanism on the original sketch directly, reaching a high accuracy while not demanding a high computation cost. Experiments conducted on three real-world sketch datasets clearly demonstrate the state-of-the-art retrieval performance on both complete and partial sketch settings.

ACKNOWLEDGEMENT

This research was supported in part by an NSF IIS award IIS-1814450 and an ONR award N00014-18-1-2875. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agency.

REFERENCES

[1] Q. Yu, F. Liu, Y.-Z. Song, T. Xiang, T. M. Hospedales, and C. C. Loy, "Sketch me that shoe," in *CVPR*, 2016, pp. 799–807.

[2] J. Song, Q. Yu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Deep spatial-semantic attention for fine-grained sketch-based image retrieval," in *ICCV*, 2017, pp. 5552–5561.

[3] K. Pang, K. Li, Y. Yang, H. Zhang, T. M. Hospedales, T. Xiang, and Y.-Z. Song, "Generalising fine-grained sketch-based image retrieval," in *CVPR*, 2019, pp. 677–686.

[4] S. Dey, P. Riba, A. Dutta, J. Lladós, and Y.-Z. Song, "Doodle to search: Practical zero-shot sketch-based image retrieval," *CVPR*, pp. 2174–2183, 2019.

[5] A. Dutta and Z. Akata, "Semantically tied paired cycle consistency for zero-shot sketch-based image retrieval," *CVPR*, pp. 5084–5093, 2019.

[6] J. Collomosse, T. Bui, and H. Jin, "Livesketch: Query perturbations for guided sketch-based visual search," in *CVPR*, June 2019.

[7] L. Guo, J. Liu, Y. Wang, Z. Luo, W. Wen, and H. Lu, "Sketch-based image retrieval using generative adversarial networks," in *ACM MM*, 2017, p. 1267–1268.

[8] P. Xu, Y. Huang, T. Yuan, K. Pang, Y.-Z. Song, T. Xiang, T. M. Hospedales, Z. Ma, and J. Guo, "Sketchmate: Deep hashing for million-scale human sketch retrieval," *CVPR*, pp. 8090–8098, 2018.

[9] Y. Li, T. M. Hospedales, Y. zhe Song, and S. Gong, "Fine-grained sketch-based image retrieval by matching deformable part models," *BMVC*, 2014.

[10] J. Lei, Y. Song, B. Peng, Z. Ma, L. Shao, and Y.-Z. Song, "Semi-heterogeneous three-way joint embedding network for sketch-based image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 3226–3237, 2020.

[11] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, "The sketchy database: Learning to retrieve badly drawn bunnies," *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, 2016.

[12] T. X. Kaiyue Pang, Yi-zhe Song and T. Hospedales, "Cross-domain generative learning for fine-grained sketch-based image retrieval," in *BMVC*, G. B. Tae-Kyun Kim, Stefanos Zafeiriou and K. Mikolajczyk, Eds. BMVA Press, September 2017, pp. 46.1–46.12.

[13] A. Bhunia, Y. Yang, T. M. Hospedales, T. Xiang, and Y.-Z. Song, "Sketch less for more: On-the-fly fine-grained sketch-based image retrieval," *CVPR*, pp. 9776–9785, 2020.

[14] T. Bui, L. Ribeiro, M. Ponti, and J. Collomosse, "Deep manifold alignment for mid-grain sketch based image retrieval," *ACCV*, vol. 11363, pp. 314 – 329, 2019.

[15] Y. Cao, C. Wang, L. Zhang, and L. Zhang, "Edgel index for large-scale sketch-based image search," in *CVPR*, 2011, pp. 761–768.

[16] J. Collomosse, T. Bui, M. J. Wilber, C. Fang, and H. Jin, "Sketching with style: Visual search with sketches and aesthetic context," in *ICCV*, Oct 2017.

[17] D. R. Ha and D. Eck, "A neural representation of sketch drawings," *ArXiv*, vol. abs/1704.03477, 2017.

[18] L. Liu, F. Shen, Y. Shen, X. Liu, and L. Shao, "Deep sketch hashing: Fast free-hand sketch-based image retrieval," *CVPR*, pp. 2298–2307, 2017.

[19] L. Kaelbling, M. Littman, and A. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, 1996.

[20] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

[21] A. Guez, D. Silver, and P. Dayan, "Efficient bayes-adaptive reinforcement learning using sample-based search," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[22] I. Osband, D. Russo, and B. Van Roy, "(more) efficient reinforcement learning via posterior sampling," in *Advances in Neural Information Processing Systems*, vol. 26, 2013, p. 3003–3011.

[23] M. Strens, "A bayesian framework for reinforcement learning," in *Proceedings of The 17th International Conference on Machine Learning*. ICML, 2000, pp. 943–950.

[24] I. Osband, B. V. Roy, and Z. Wen, "Generalization and exploration via randomized value functions," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2377–2386.

[25] I. Osband, C. Blundell, A. Pritzel, and B. V. Roy, "Deep exploration via bootstrapped dqn," in *NIPS*, 2016.

[26] J. Song, K. Pang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Learning to sketch with shortcut cycle consistency," *CVPR*, pp. 801–810, 2018.

[27] U. R. Muhammad, Y. Yang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Learning deep sketch abstraction," in *CVPR*, June 2018.

[28] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," *ArXiv*, vol. abs/1602.01783, 2016.

[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *ArXiv*, vol. abs/1707.06347, 2017.