

Sketching without Worrying: Noise-Tolerant Sketch-Based Image Retrieval

Ayan Kumar Bhunia¹ Subhadeep Koley^{1,2} Abdullah Faiz Ur Rahman Khilji* Aneeshan Sain^{1,2}
Pinaki Nath Chowdhury^{1,2} Tao Xiang^{1,2} Yi-Zhe Song^{1,2}
¹SketchX, CVSSP, University of Surrey, United Kingdom.
²iFlyTek-Surrey Joint Research Centre on Artificial Intelligence.

{a.bhunia, s.koley, a.sain, p.chowdhury, t.xiang, y.song}@surrey.ac.uk

Abstract

Sketching enables many exciting applications, notably, image retrieval. The fear-to-sketch problem (i.e., “I can’t sketch”) has however proven to be fatal for its widespread adoption. This paper tackles this “fear” head on, and for the first time, proposes an auxiliary module for existing retrieval models that predominantly lets the users sketch without having to worry. We first conducted a pilot study that revealed the secret lies in the existence of noisy strokes, but not so much of the “I can’t sketch”. We consequently design a stroke subset selector that detects noisy strokes, leaving only those which make a positive contribution towards successful retrieval. Our Reinforcement Learning based formulation quantifies the importance of each stroke present in a given subset, based on the extent to which that stroke contributes to retrieval. When combined with pre-trained retrieval models as a pre-processing module, we achieve a significant gain of 8%-10% over standard baselines and in turn report new state-of-the-art performance. Last but not least, we demonstrate the selector once trained, can also be used in a plug-and-play manner to empower various sketch applications in ways that were not previously possible.

1. Introduction

Thanks to the convenience of interactive touchscreen devices, sketch-based image retrieval (SBIR) [11, 12, 14, 38] has emerged as a practical means of image research that is complementary to the conventional text-based retrieval [25]. Although initially developed for a category-level setting [42, 36, 59], of late SBIR has undertaken a fine-grained shift to better reflect the inherent fine-grained characteristics (pose, appearance detail, etc) of sketches [46, 56, 7].

Despite great strides made [3, 33, 10], the fear-to-sketch has proven to be fatal for its omnipresence – a “I can’t sketch” reply is often the end of it. This “fear” is predominant for fine-grained SBIR (FG-SBIR), where the system dictates users to produce even more faithful and diligent

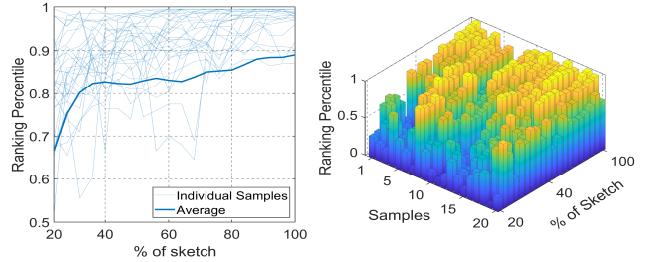


Figure 1: (a) While the *average* ranking percentile increases as the sketching proceeds from starting towards completion, *unwanted sudden drops* have been noticed for many individual sketches due to noisy/irrelevant strokes drawn. (b) The same thing is visualised with number of samples in the third axis to get an overall statistics on QMUL-Shoe-V2 dataset.

sketches than that required for category-level retrieval [11].

In this paper, we tackle this “fear” head-on and propose for the first time a *pre-processing* module for FG-SBIR that essentially let the users sketch without the worry of “I can’t”. We first experimentally show that, in most cases it is not about how bad a sketch is – most *can* sketch (even a rough outline) – the devil lies in the fact that users typically draw irrelevant (noisy) strokes that are detrimental to the overall retrieval performance (see Section 3). This observation has largely inspired us to alleviate the “can’t sketch” problem by *eliminating* the noisy strokes through selecting an optimal subset that *can* lead to effective retrieval.

This problem might sound trivial enough – e.g., how about considering all possible stroke subsets as training samples to gain model invariance against noisy strokes? Albeit theoretically possible, the highly complex nature of this process (i.e., $\mathcal{O}(2^N)$) quickly renders this naive solution infeasible, especially when the number of strokes in freehand sketches can range from an average of $N = 9$ to a max of $N = 15$ in fine-grained SBIR datasets (QMUL-ShoeV2/ChairV2 [56, 46]). Most importantly, augmenting the training data by random stroke dropping would lead to a noisy gradient during training. This is because out of all possible subsets, many of these augmented sketch subsets

*Interned with SketchX

are too coarse/incomplete to convey any meaningful information to represent the paired photo. Therefore, instead of naively learning the invariance, we advocate for finding meaningful subsets that can sustain efficient retrieval.

Our solution generally rests with detecting noisy strokes and leaving only those that positively contribute to successful retrieval. We achieve that by proposing a mechanism to quantify the *importance* of each stroke present in a given stroke-set, based on the extent to which that stroke is *worthy for retrieval* (i.e., makes a positive contribution). We work on vector sketches[4] in order to utilise stroke-level information, and propose a sketch stroke subset selector that learns to determine a binary action for every stroke – whether to include that particular stroke to the query stroke subset, or not. The stroke subset selector is designed via a hierarchical Recurrent Neural Network (RNN) that models the compositional relationship among the strokes. Once the stroke subset is obtained, it is first *rasterized* then passed through a pre-trained FG-SBIR model [56] to obtain a ranking of target photos against the ground-truth photo. The main objective is to select a particular subset that will rank the paired ground-truth photo towards the top of the ranking list. We use Reinforcement Learning (RL) based training due to the non-differentiability of rasterization operation. As explicit stroke-level ground-truth for the optimal subset is absent, we seek to train our stroke-subset selector with the help of pre-trained FG-SBIR for reward computation. In particular, we use the actor-critic version of proximal policy optimisation (PPO) to train the stroke subset selector.

Apart from the main objective of noisy stroke elimination, the proposed method also enables a few secondary sketch applications (Section 5) in a plug-and-play manner. First, we show that a pre-trained stroke selector can be used as a *stroke importance quantifier* to guide users to produce a sketch “just” enough for successful retrieval. Second, we demonstrate that it can significantly speed up existing works on interactive “on-the-fly” retrieval [7] removing the need for incomplete rasterized sketch to be unnecessarily passed for inference multiple times. Third, besides benefiting FG-SBIR, our subset selector module can also act as a faithful *sketch data augmente*r over random stroke dropping without much computational overhead. That is, instead of costly operation like sketch deformation [58] or unfaithful approximation like edge/contour-map as soft ground-truths [9], users can effortlessly generate n most representative subsets to augment training for many downstream tasks.

In summary our contributions are, (a) We tackle the fear-to-sketch problem for sketch-based image retrieval for the first time, (b) We formulate the “can’t sketch” problem as stroke subset selection problem following detailed experimental analysis, (c) We propose a RL-based framework for stroke subset selection that learns through interacting with a pre-trained retrieval model. (d) We demonstrate our pre-

trained subset selector can empower other sketch applications in a plug-and-plug manner.

2. Related Works

Category-level SBIR: Category-level SBIR aims at retrieving category-specific photos from user given query sketches. Like any other retrieval system, Deep Neural Networks have become a de-facto choice for any recent SBIR frameworks [14, 12, 36, 59, 11, 6] over early hand-engineered feature descriptors [49]. Overall, category level SBIR makes use of Siamese networks based on either CNN [11, 12], RNN [53], Transformer [36] or their combinations [11] along with a triplet-ranking objective to learn a joint embedding space. A distance metric is used to rank the gallery photos against the learned embedding space for a given query sketch for retrieval. Further efforts have been made through zero-shot SBIR [12, 55] for cross-category generalisation, and employing binary hash-code embedding [28, 42] to reduce the computational complexity.

Fine-grained SBIR: Sketch holds a noteworthy advantage in its potential to depict fine-grained properties of the target image, which are hard to describe via other query mediums [45] like text or attribute. Consequently, interest surged in fine-grained SBIR [56], which aims at *instance-specific* matching for a user given query sketch. Initially starting with graph-matching models [33], FG-SBIR research gained traction with the advent of various deep-learning based approaches [56, 46, 7, 3]. Yu *et al.* [56] first pioneered deep triplet-ranking based *siamese networks* for learning a joint embedding space with instance-wise matching criteria. This was further augmented via attention with higher-order retrieval loss [46], cross-domain image generation [34], text tags [45], etc. Recent FG-SBIR works include advanced methods like hierarchical co-attention [39], reinforcement learning-based early retrieval [7], semi-supervised generation-retrieval joint training [3], etc.

While sketches are significantly subjective to user’s style [40] and vary considerably depending on the drawer’s drawing skill [7], these earlier works *assumed* the existing annotated fine-grained dataset to be *perfect*. In other words, a *rigid* assumption is made that every annotated paired sketch is a perfect depiction of the paired photo. In this work, we argue that ‘*all sketches are sketchy*’, which holds stronger significance for fine-grained SBIR, as every stroke of annotated sketch [57] represents a specific part of the paired photo, and the free-flow nature of amateur sketching is likely to introduce noise no matter how carefully it is drawn.

Modelling Partial Sketches: “Sketch” being an interactive medium, is drawn sequentially in a stroke-by-stroke manner. Moreover, due to its subjective nature, the same sketch might be perceived as partial or complete based on the user’s perception. Users can retrieve photos [7], create [50] imaginative visual-art, or edit existing photos [21]

through repeated interactions with the AI agent. Therefore, on-the-fly interaction with sketches requires sketch-based models to be capable of handling partial sketches. For instance, Sketch-RNN [16] can predict probable final sketch endings using a variational autoencoder trained on the vector sketch coordinates. Furthermore, attempts have been made to directly recognise partial sketches [27] and achieve sketch-to-photo generation [15] from incomplete sketch input, where both works involve a sketch-completion module based on image-to-image translation. Recently, on-the-fly FG-SBIR [7] has been introduced to retrieve even from a few elementary strokes as soon as the users start drawing. Overall, these works try to include random synthetic partial sketches during training to achieve their respective goals, but here we aim to answer “*whether a partial sketch has sufficient representative information/discriminative potential to retrieve photos faithfully*”. Furthermore, we aim to quantify the instant at which a sequentially drawn sketch would reach the optimum threshold point where it is representative enough for downstream tasks (e.g., retrieval). By doing so, we can faithfully train models with sufficiently representative partial sketches instead of randomly dropping strokes and ignoring instances where the synthetic partial sketch is too coarse to convey any meaning.

Reinforcement Learning in Vision: Reinforcement Learning (RL) [22] has been applied in different vision problems [26, 51]. RL becomes handy when there exists a non-differentiable way to quantify the *goodness* of the network’s state unlike differentiable loss function with hard-labels. Instead, learning progresses via interactions [13, 18] with the environment. Particularly in sketch community, RL has been leveraged for modelling sketch abstraction [31, 30], retrieval [7, 3], and designing competitive sketching agent [5]. Here, our objective is to engage an RL agent to get rid of noisy sketch strokes for better retrieval.

Learning from noisy labels: Despite significant progress from the community-generated labelled data, accurate labelling is challenging even for experienced domain experts [44]. Therefore, a separate topic of study [44, 60, 60] emerged, which aims at learning robust models even from the noisy data distribution. While the existing works [17, 48] mainly consider having access to a large, noisy dataset as well as a subset of carefully cleaned data for validation , our situation is even more difficult than usual. We assume that every annotated sketch is not an absolutely perfect matching sketch of the paired photo. Therefore, we aim to develop a noise-tolerant framework for FG-SBIR.

3. Pilot Study: What’s Wrong with FG-SBIR?

Baseline FG-SBIR: Instead of complicated pre-training [35] or joint-training [3], we use a three branch state-of-the-art Siamese network [3] as our baseline retrieval model, which is considered to be a strong baseline till

date. Each branch starts from ImageNet pre-trained VGG-16 [23], sharing equal weights. Given an input image $I \in \mathbb{R}^{H \times W \times 3}$, we extract the convolutional feature-map $\mathcal{F}(I)$, which upon global average pooling followed by l_2 normalisation generates a d dimensional feature embedding. This model has been trained with an anchor sketch (a), a positive (p) photo, and a negative (n) photo triplets $\{\bar{a}, \bar{p}, \bar{n}\}$ using *triplet-loss* [52]. Triplet-loss aims at increasing the distance between anchor sketch and negative photo $\delta^- = \|\mathcal{F}(\bar{a}) - \mathcal{F}(\bar{n})\|_2$, while simultaneously decreasing the same between anchor sketch and positive photo $\delta^+ = \|\mathcal{F}(\bar{a}) - \mathcal{F}(\bar{p})\|_2$. Therefore, the triplet-loss with margin $\mu > 0$ can be written as:

$$\mathcal{L}_{Triplet} = \max\{0, \delta^+ - \delta^- + \mu\} \quad (1)$$

Dual representation of sketch: Recent study has emphasised on the dual representation [4] of sketch for self-supervised feature learning. In rasterized pixel modality \mathcal{I} , sketch can be represented as spatially extended image of size $\mathbb{R}^{H \times W \times 3}$. On the other side, in vector modality \mathcal{V} , the same sketch can be characterised by a sequence of strokes (s_1, s_2, \dots, s_K) where each stroke is a sequence of successive points $s_i = (v_1^i, v_2^i, \dots, v_{N_i}^i)$, and each point is represented by an absolute 2D coordinate $v_n^i = (x_n^i, y_n^i)$ in a $H \times W$ canvas. Here, K is number of strokes and N_i is the number of points inside i^{th} stroke. Individual strokes arise due to pen up/down [16] movement. Although sketch vectors can easily be recorded through touch screen-devices, generation of the corresponding rasterized sketch image needs a costly [54] *rasterization* operation $\mathcal{R} : \mathcal{V} \rightarrow \mathcal{I}$. Either modality, raster or vector, has its own merits and demerits [4]. Apart from being more computationally efficient [54] than raster domain, vector modality also contains the stroke-by-stroke temporal information [16]. Nonetheless, sketch vectors lack the spatial information [4] which is critical to model the fine-grained details [3, 7]. Consequently, rasterized sketch image is the standard choice [35, 40, 39, 56] for FG-SBIR despite having a higher computational overhead and lacking temporal information.

Preliminary analysis: The performance barrier due to irrelevant strokes gets noticed under on-the-fly FG-SBIR [7] setup. Instead of only evaluating the complete sketch, we start rendering at the end of every new k^{th} stroke drawn as the rasterized sketch image $S_k^{\mathcal{I}} = \mathcal{R}([s_1, s_2, \dots, s_k])$ where $k = \{1, 2, \dots, K\}$, and pass it through the *pre-trained* baseline FG-SBIR model to get the feature representation $\mathcal{F}(S_k^{\mathcal{I}})$, followed by ranking the gallery images against it. We make these following observations on Shoe-V2 [56] dataset (*Linear Limit*): (i) As the sketch proceeds towards completion, the rank is supposed to be improved, however, we notice some unexpected dips in the performance in the later part of the drawing episode. This signifies that the later irrelevant strokes play a detrimental role, thereby degrading the retrieval performance (Fig. 1). (ii)

Compared to top@1(top@5) accuracy of 33.43%(67.81%) on using complete sketch for retrieval, if we consider best rank achieved at any of the instant during the sketch drawing episode as the retrieved result, top@1(top@5) accuracy extends to 42.54%(73.28%). (iii) Further, we note that the percentage of instances where subsequently added strokes drops the performance compared to the previous version $S_k^{\mathcal{I}}$ of the same sketch is 43.44%, which is a critical number.

Ablation on upper limit: Prior analysis unfolds the necessity of dealing with irrelevant stroke, and we *hypothesise that in many cases a subset of the strokes $K' \leq K$ could better retrieve the paired photo by excluding the irrelevant ones*. Different people follow varying stroke order for sketching. Therefore, in order to simulate different possible stroke orders and to estimate the upper limit that we can achieve through the smart stroke-subset selector, we do the following study. Given K strokes in a sketch, we form $(2^K - 1)$ stroke subsets taking *any* number of strokes at a time. Unlike the “on-the-fly” [7] protocol, this setting does not stick to a pre-recorded sequential order, rather it aims to find if there exists *any subset* that can retrieve the paired photo better than the entire sketch set. Under this setting, we achieve an exceptionally high top@1(top@5) accuracy of 66.37%(88.31%). However, evaluating with every possible stroke combination during real-time inference is *impractical*, and we do not have any explicit way to select one final result. Therefore, in this work, we seek to build a *smart stroke-subset selector* as a *pre-processing* module which when plugged in before any pre-trained FG-SBIR model [56, 45], will aim to construct the most representative subset to improve the overall accuracy.

4. Noisy Stroke Tolerant FG-SBIR

Overview: Our preliminary study motivates us to design a stroke-subset selector to eliminate the noisy strokes for FG-SBIR. While *raster sketch image* is essential [3] to model the fine-grained correspondence, the stroke-level sequential information is missing in raster modality. Therefore, taking advantage of the dual representation [4] of the sketch, we model the stroke subset selector on the sequential vector space. In summary, our noise-tolerant FG-SBIR consists of two following modules connected in cascade: (a) *stroke-subset selector* as pre-processing module working in vector space and (b) *pretrained FG-SBIR* $\mathcal{F}(\cdot)$ that uses rasterized version of predicted subset for final retrieval.

4.1. Stroke Subset Selector

Model: Given sketch-photo pair (S, P) , the sketch S can be represented as both raster image S_I and stroke-level sequential vector $S_V = (s_1, s_2, \dots, s_K)$. We design a stroke-subset selector $\mathcal{X}(\cdot)$ that takes S_V as input, and aim to predict an optimal subset $\bar{S}_V = \mathcal{X}(S_V)$ with K' strokes where $K' \leq K$. However, selecting the optimal subset of stroke

is an ill-posed problem. Firstly, there is *no explicit label* which represents the optimal stroke subset. In fact, there might be many sub-sets which can lead to successful retrieval. Furthermore, annotating the optimal stroke-subsets for the whole training dataset via brute-force iteration is computationally impractical [5].

In our framework, we treat stroke subset selector as a binary categorical classification problem. In other words, for a sketch of K strokes, we get an output of size $\mathbb{R}^{K \times 2}$, where every row is softmax normalised and it represents a probability distribution $p(a_i|s_i)$ over two classes: $a \in \{\text{select}, \text{ignore}\}$. However, we do not have any explicit one-hot labels for this binary classification task. Therefore, we let the stroke sub-set selector agent to interact with the pre-trained FG-SBIR model, and \mathcal{X} is learned using a pre-trained FG-SBIR model \mathcal{F} as a *critic* which provides the training signal to \mathcal{X} .

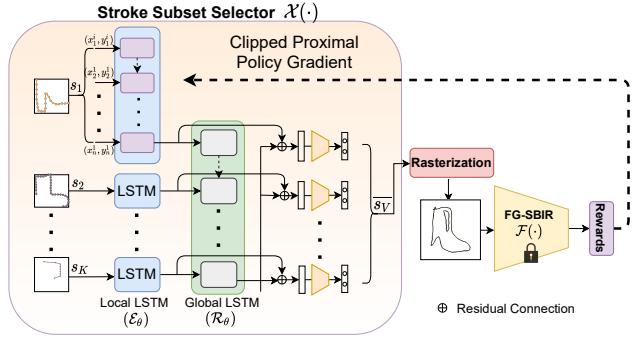


Figure 2: Illustration of Noise Tolerant FG-SBIR framework. Stroke Subset Selector $\mathcal{X}(\cdot)$ acts as a pre-processing module in the sketch vector space to eliminate the noisy strokes. Selected stroke subset is then rasterized and fed through an existing pre-trained FG-SBIR model for reward calculation, which is optimised by Proximal Policy Optimisation. For brevity, actor-only version is shown here.

Architecture: To design the architecture of stroke-level selector, we aim at preserving localised stroke-level information, as well as the compositional relationship [1] among the strokes, which together conveys the overall semantic meaning. Therefore, we employ a two-level hierarchical model comprising of a local stroke-embedding network (E_θ) and global relational network (R_θ) to enrich each stroke-level feature about the global semantics. In particular, we feed individual stroke of size $\mathbb{R}^{N_i \times 2}$ having N_i points through a local stroke-embedding network E_θ (e.g. RNN, LSTM or Transformer) whose weights of E_θ are shared across strokes. We take the final hidden-state feature as the localised representation $f_{s_i}^l \in \mathbb{R}^{d_s}$ for i^{th} stroke. Thereafter, feature representation of K such strokes having size of $\mathbb{R}^{K \times d_s}$ are further fed to a global relational network (R_θ) whose final hidden state $f^g \in \mathbb{R}^{d_s}$ captures the global semantic information of the whole sketch. Taking inspiration from residual learning [19], we fuse the global feature with

individual stroke-level feature through a residual connection with LayerNorm [2]. In concrete, every stroke feature enriched by global-local compositional hierarchy is represented by $\hat{f}_{s_i} = \text{LayerNorm}(f_{s_i}^l + f^g) \in \mathbb{R}^d$. We implement both \mathcal{E}_θ and \mathcal{R}_θ through a one layer LSTM with hidden state size 128. Further, we apply a shared linear layer (\mathcal{C}_θ) to get $p(a_i|s_i) = \text{softmax}(W_\mathcal{X}\hat{f}_{s_i} + b_\mathcal{X})$, where $W_\mathcal{X} \in \mathbb{R}^{d_s \times 2}$ and $b_\mathcal{X} \in \mathbb{R}^2$. We group three modules $\{\mathcal{R}_\theta, \mathcal{E}_\theta, \mathcal{C}_\theta\}$ of stroke subset selector as \mathcal{X}_θ . See Fig. 2.

4.2. Training Procedure

Necessity of RL: Due to the unavailability of ground-truth for optimum strokes, we rely on the pre-trained FG-SBIR model to learn the optimum stroke-subset selection strategy. In particular, given probability distribution $p(a_i|s_i) \in \mathbb{R}^2$ for every stroke over $\{\text{select}, \text{ignore}\}$, we can sample from categorical distribution as $a_i \sim \text{Categorical}([p(a_{\text{select}}|s_i), p(a_{\text{ignore}}|s_i)])$, and thereby we will be getting a stroke subset as \bar{S}_V with K' strokes, where $K' \leq K$. In order to get the training signal from pre-trained FG-SBIR model \mathcal{F} , we need to feed the subset sketch through \mathcal{F} . For that, we need to convert the sequential sketch vector to raster sketch image through rasterization $\bar{S}_I = \mathcal{R}(\bar{S}_V)$, as fine-grained SBIR model only [3, 7] works on raster image space. While subset sampling could be relaxed by Gumbel-Softmax [20] operation for differentiability, non-differentiable rasterization operation $\mathcal{R}(\cdot)$ squeeze us to use Policy-Gradient [47] from Reinforcement Learning (RL) literature [22].

MDP Formulation: In particular, given an input sketch S_V (initial state), the stroke-subset selector (\mathcal{X}_θ) acts a *policy network* which takes *action* on selecting every stroke, and we get an updated state as subset-sketch \bar{S}_V (next state). In order to train the policy network, we calculate *reward* using \mathcal{F} as a critic. Therefore, we can form the tuple of four elements (*initial_state*, *action*, *reward*, *next_state*) that is typically required to train any RL model. In order to model the existence of multiple possible successful subsets, we unroll this sequential Markov Decision Process (MDP) T times starting from the complete sketch vector. In other words, for each sketch data, we sequentially sample the subset strokes T times to learn the multi-modal nature of true stroke subsets. Empirically we keep *episode length* $T = 5$.

Reward Design: Our objective is to select the optimum set of stroke which can retrieve the paired photo with minimum rank (e.g. best scenario: rank 1). In other words, pairwise-distance between the query sketch and paired photo embeddings should be lower than that of query sketch and rest other photos of the gallery. As \mathcal{F} is fixed, we can pre-compute the features of all M gallery photos as $G \in \mathbb{R}^{M \times D}$ – thus eliminating the burden of repeatedly computing the photo features. During stroke sub-

set selector training, we just need to calculate the feature embedding $\mathcal{F}(\bar{S}_I)$ of rasterized version of predicted subset sketch, and we can calculate rank of paired photo using G and paired-photo index efficiently. We compute the reward both in the ranking space as well as in the feature embedding space using standard triplet loss on $\mathcal{F}(\bar{S}_I)$ following Eqn. 1, which is found to give better stability and faster training convergence. In particular, we want to minimise the rank of the paired photo and triplet loss simultaneously. Following the conventional norm of reward maximisation, we define the *reward* (R) as weighted summation of inverse of the rank and negative triplet loss as follows:

$$R = \omega_1 \cdot \frac{1}{\text{rank}} + \omega_2 \cdot (-\mathcal{L}_{\text{Triplet}}) \quad (2)$$

Actor Critic PPO: We make use of actor-critic version of Proximal Policy Optimisation (PPO) with clipped surrogate objective [41] to train our stroke-subset selector. In particular, the very basic policy gradient [47] objective that is to be minimised could be written as:

$$L^{PG}(\theta) = -\frac{1}{K} \sum_{i=1}^K \log p_\theta(a_i|s_i) \cdot R \quad (3)$$

For sampling efficiency, using the idea of Importance Sampling [32], PPO maintains an older policy $p'_\theta(a_i|s_i)$, and thus Conservative Policy Iteration (CPI) objective becomes $L^{CPI}(\theta) = -\frac{1}{K} \sum_{i=1}^K r_i(\theta) \cdot R$, where $r_i(\theta) = \log p_\theta(a_i|s_i)/\log p'_\theta(a_i|s_i)$. Further on, the clipped surrogate objective PPO can be written as $L^{CLIP}(\theta) = -\frac{1}{K} \sum_{i=1}^K \text{clip}(r_i(\theta), 1 - \varepsilon, 1 + \varepsilon)$, which aims to penalise too large policy update with hyperparameter $\varepsilon = 0.2$. We take a minimum of the clipped and unclipped objective, so the final objective is a lower bound (i.e., a pessimistic bound) on the unclipped objective. The final *actor only* version PPO objective becomes:

$$L^A(\theta) = -\frac{1}{K} \sum_{i=1}^K \min(L^{CPI}, L^{CLIP}) \quad (4)$$

To reduce the variance, the *actor-critic* version of PPO make use of a *learned state-value function* $V(S)$ where S is the sketch vector $S = (s_1, s_2, \dots, s_K)$. $V(S)$ shares parameter with actor network \mathcal{X}_θ , where only the last linear layer (\mathcal{C}_θ) is replaced by a new linear layer upon a single latent vector (accumulated stroke-wise features by averaging), predicting a scalar value that tries to *approximate the reward value*. Thus, the final loss function combines the policy surrogate and value function error time together with a entropy bonus (E_n) to ensure sufficient exploration is:

$$L^{AC}(\theta) = -\frac{1}{K} \sum_{i=1}^K (L^A - c_1(V_\theta(S) - R)^2 + c_2 E_n) \quad (5)$$

where, c_1 and c_2 are coefficients. As we unroll the sequential stroke-subset selection process for $T = 5$, for every sample the loss accumulated over the MDP episode is $\frac{1}{T} \sum_{t=1}^T L_t^{AC}(\theta)$.

5. Applications of Stroke-Subset Selector

Resistance against noisy strokes: Collected sketch labels, which are used to train the initial fine-grained SBIR model are also noisy. The proposed stroke-subset selector *not only assists during inference* by noisy-stroke elimination, but also *helps in cleaning training data*, which in turn can boost the performance to some extent. In particular, we train the FG-SBIR model and Stroke-Subset Selector in stage-wise alternative manner, with the FG-SBIR model using clean sketch labels produced by the trained stroke-subset selector. Our method thus offers a plausible way to alleviate the latent/hidden noises of a FG-SBIR dataset [56].

Modelling ability to retrieve: As the critic network tries to approximate the scalar reward value which is a measure of retrieval performance, we can use the *critic-network* to quantify the retrieval ability at any instant of a sketching episode. Higher scalar score from the critic signifies better retrieval ability. To wit, we ask the question whether a partial sketch is good enough for retrieval or not. Thus, instead of feeding rasterized partial sketch multiple times for on-the-fly [7] retrieval, we can save significant computation cost by feeding *only after* it gains a potential retrieval ability. Moreover, as both our actor and critic networks work in sketch vector modality, it adds less computational burden.

On-the-fly FG-SBIR: Training from Partial Sketches: State-of-the-art on-the-fly FG-SBIR [7] employs continuous RL for training using ranking objective. A supervised triplet-loss [58] based training, augmented with synthetic partial sketches obtained through random stroke-dropping is claimed to be sub-optimal, as randomly dropped strokes frequently banish crucial details, resulting in the augmented partial sketch containing insufficient information to depict the paired photo. In contrast, we use our stroke-subset selector to create several augmented partial versions of the same sketch, each with *sufficient retrievability*. While continuous RL is time intensive to train and allegedly unstable [22], we can use simple triplet-loss based supervised learning with multiple *meaningful augmented partial sketches*.

6. Experiments

Datasets: Two publicly available FG-SBIR datasets [56, 33, 7] namely QMUL-Shoe-V2 and QMUL-Chair-V2 are used in our experiments. Apart from having instance-wise paired sketch-photo, these datasets also contain the sketch coordinate information, and thus would enable us to train the stroke-subset selector using sketch vector modality. We use the standard training/testing split used by the existing state-of-the-arts. In particular, out of 6,730 (1,800) sketches and 2,000 (400) photos from Shoe-V2 (Chair-V2) dataset, 6,051 sketches (1,275) and 1,800 (300) photos are used for training respectively, and the rest are for testing [7].

Implementation: We have conducted all our experi-

ments on an 11-GB Nvidia RTX 2080-Ti GPU with PyTorch. For *fine-grained SBIR*, we have used ImageNet [37] pre-trained VGG-16 [43] backbone with feature embedding dimension $d = 512$. We train the FG-SBIR model using Adam optimiser [24] with a learning rate of 0.0001, batch size 16, and margin value of 0.2 for triplet loss. For *stroke subset selector*, we model local stroke embedding network and global relational network using one-layer LSTM with hidden state size 128 for each. The critic network shares the same weights with that of the actor, with only the last linear layer C_θ being replaced by a new one that predicts a single scalar value. We train it for 2000 epoch using Adam optimiser with initial learning rate 10^{-4} till 100 epochs, then reducing to 10^{-5} . We use a batch size of 16 and keep an old policy network for importance sampling [32] with episode length $T = 5$, and sampled instances are stored in a replay buffer. We update the current policy network at every 20 iteration using sampled instances from the replay buffer, and the old policy network's weights are copied from the current one for subsequent sampling. We empirically set both ω_1, ω_2 to 1, and keep $c_1 = 0.5, c_2 = 0.01, \epsilon = 0.2$.

Evaluation Metric: **(a) Standard FG-SBIR:** Aligning to the existing state-of-the-art FG-SBIR frameworks [35, 56], we use percentage of sketches having true-matched photo in the top-1 (acc.@1) and top-5 (acc.@5) lists to assess the FG-SBIR performance. **(b) On-the-fly FG-SBIR:** Furthermore, to showcase the early retrieval performance from partial sketch, adhering to prior early-retrieval work [7] we employ two plots namely, (i) *ranking-percentile* and (ii) $\frac{1}{rank}$ vs. *percentage of sketch*. Higher area under these curves indicate better early-retrieval potential. For the sake of simplicity, we call area under curves (i) and (ii) as r@A and r@B through the rest of the paper.

Competitors: To the best of our knowledge, no earlier works have directly attempted to design a Noise-Tolerant FG-SBIR model in the SBIR literature. Therefore, we compare with the existing standard FG-SBIR works appeared in the literature, as well as, we develop some self-designed competitive baselines under the assumption of '*all sketches are sketchy*' – which explicitly intend to learn invariance against noisy strokes. **(a) State-of-the-arts (SOTA):** While **Triplet-SN** [56] uses Sketch-A-Net backbone along with triplet loss, **Triplet-Attn-HOLEF** extends [56] with spatial attention and higher order ranking loss. Recent works include: **Jigsaw-Pretrain** with self-supervised pre-training, **Triplet-RL** [7] employing RL-based fine-tuning, **Style-MeUP** involving MAML training, **Semi-Sup** [3] incorporating semi-supervised paradigm, and **Cross-Hier** [39] utilising cross-modal hierarchy with costly paired-embedding. **(b) Self-designed Baselines (BL):** We create multiple version of the same sketch by randomly dropping strokes (ensuring percentage of sketch vector length never drops below 80%) or by synthetically adding random noisy stroke

Table 1: Results under Standard FG-SBIR setup.

		Chair-V2		Shoe-V2	
		Acc@1	Acc@5	Acc@1	Acc@5
SOTA	Triplet-SN [56]	47.4%	71.4%	28.7%	63.5%
	Triplet-Attn-HOLEF [46]	50.7%	73.6%	31.2%	66.6%
	Triplet-RL [7]	51.2%	73.8%	30.8%	65.1%
	Mixed-Jigsaw [33]	56.1%	75.3%	36.5%	68.9%
	Semi-Sup [3]	60.2%	78.1%	39.1%	69.9%
	StyleMeUp [40]	62.8%	79.6%	36.4%	68.1%
BL	Cross-Hier [39]	62.4%	79.1%	36.2%	67.8%
	(B)aseline-Siamese	53.3%	74.3%	33.4%	67.8%
	Augmnt	54.1%	74.6%	33.9%	68.2%
	StyleMeUp+Augment	56.1%	76.9%	36.9%	69.9%
Limits	Contrastive+Augment	58.8%	77.1%	37.6%	70.1%
	Upper-Limit	78.6%	90.3%	66.3%	88.3%
	Linear-Limit	59.4%	77.3%	42.5%	73.2%
	Proposed	64.8%	79.1%	43.7%	74.9%

patches similar to [29]. **Augmnt** aims to learn the invariance against noisy stroke by adding them inside training. This is further advanced by **StyleMeUp+Augment** where synthetic noisy/augmented sketches are mixed in the inner-loop of [40] to learn invariance by optimising outer-loop on real sketches. **Contrastive+Augment** imposes an additional contrastive loss [8] such that the distance between two augmented versions of same sketch should be lower than that of with a random other sketch. Our pre-trained baseline FG-SBIR model is termed as **B-Siamese**.

6.1. Performance Analysis

The comparative analysis is shown in Table 1. Overall, we observe a significantly improved performance of our proposed Noise-Resistant fine-grained SBIR employing a stroke-subset selector as a pre-processing neural agent compared to the existing state-of-the-art. The early works tried to address different *architectural modifications* [45, 33], and later on the field of fine-grained SBIR witnessed successive improvements through adaptation of different paradigms like *self-supervised learning* [35], *meta-learning* [40], *semi-supervised learning* [3], etc. As opposed to these works, we underpin an important phenomenon of noisy strokes, which is inherent to FG-SBIR. Most interestingly, our simple stroke-subset selector can improve the performance of baseline B-Siamese model by an approximate margin of 10.31% without any complicated joint-training of *Semi-Sup* [3], costly hierarchical paired embedding of *Cross-Hier* [39], or meta-learning cumbersome feature transformation layer of *StyleMeUp* [40]. Furthermore, the performance of *Augmnt* baseline is slightly better than our baseline pre-trained FG-SBIR as it learns some invariance from augmented/partial sketch. While we experienced difficulty in stable training for *StyleMeUp+Augment*, *Contrastive+Augment* appears as a simple and straightforward way to learn the invariance against noisy strokes. Instead of modelling invariance, we aim to eliminate the noisy strokes, thus giving a freedom of explainability through visualisation. Despite using complicated architectures [39, 3], SOTA

fails even to beat the accuracy of Linear-Limit (refer to section 3), while we can. Nevertheless, we suppress it by keeping the simple baseline FG-SBIR untouched and prepending a simple stroke-selector agent – working on a cheaper vector modality for efficient deployment.

6.2. Further Analysis and Insights

Ability to retrieve/classify for partial sketches: The scalar value predicted by our learned state-value function (critic-network) [41] signifies the retrieval ability of partial sketch with the notion of higher being the better. We here train our model with a reward of $\frac{1}{rank}$ for easy interpretability. Once the stroke-subset selector with actor-critic version is trained, we feed the sketch to the critic network (in vector space) at a progressive step of 5% completion, and record the predicted scalar value at every instant. At the same time, we rasterize every partial instance and feed through pre-trained FG-SBIR to calculate the resultant ranking percentile of the paired photo. In Fig. 3, the high correlation demonstrates that the partial sketch with a higher scalar score by the critic network tends to have a higher average ranking percentile (ARP), while those with a lesser score result in lower ARP. Quantitatively, the top@5 accuracy for partial sketches is 80.1%, which have a higher predicted scalar score than a threshold of $\frac{1}{5}$. This validates the potential of our critic network in quantifying if a partial sketch is sufficient for retrieval. Suppose we repeat the same with the negative of the classification loss as a reward for a pre-trained classification network. In that case as well, we observe a similar consistent behaviour for partial sketch classification, indicating our approach to be generic for various sketch-related downstream tasks. See § supplementary.

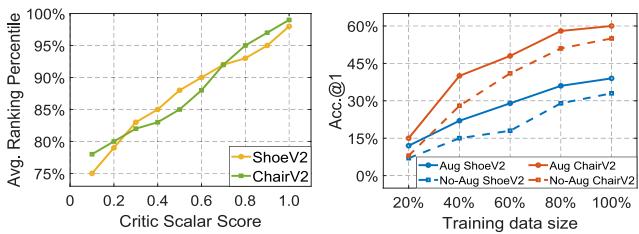


Figure 3: (a) Retrieval ability of partial sketch: correlation between critic network V(S) predicted score and ranking percentile (b) Performance at varying training data size with stroke-subset selector based data augmentation.

Data Augmentation: Our elementary study reveals that there exists multiple possible subsets which can retrieve the paired photo faithfully. In particular, we use our policy network to get stroke wise importance measure using $p(a_i|s_i)$ towards the retrieval objectives. Through categorical sampling of $p(a_i|s_i)$, we can create multiple augmented versions of the same sketch to increase the training data size. To validate this, we compute the performance of baseline retrieval model at varying training data size with our sketch augmentation strategy in Fig. 3. While accuracy remains

marginally better towards the high data regime, stroke subset selection based strategy excels the standard supervised counter-part by a significant margin, thus proving the efficacy of our smart data-augmentation approach.

On-the-Fly Retrieval: Training a model with partial sketches generated by *random stroke-dropping* gives rise to noisy gradient, and thus this naive baseline falls short compared to RL-based fine-tuning that consider the complete sketch drawing episode for training. In lieu of RL-based fine-tuning [7], we train an on-the-fly retrieval model from meaningful (holds ability to retrieve) partial sketches augmented through our critic network that have a higher scalar score than $\frac{1}{20}$. While training a continuous RL pipeline [7] is unstable and time-consuming, we achieve a competitive on-the-fly $r@A(r@B)$ performance of 85.78(21.1) with *basic* triplet-loss based model trained with *smartly* augmented partial sketches compared to 85.38 (21.24) as claimed in [7] on ShoeV2. From Fig. 4, we can see that at very early few instances, RL-Based fine-tuning [7] performs better, while ours achieve a significantly better performance as the drawing episode proceeds towards completion. While early sketch drawing episode is too coarse that hardly it can retrieve, through modelling the retrieval ability (with threshold of $\frac{1}{10}$) of partial sketches, we can reduce the number of time we need to feed the rasterized sketch by 42.2% with very little drop in performance ($r@A(r@B)$: 85.07 (20.98)). Thus modelling partial sketches lead to significant computational edge under on-the-fly setting.

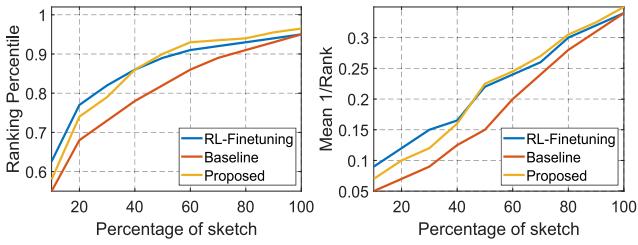


Figure 4: Comparative results under on-the-fly setup (ShoeV2), visualised through percentage of sketch. Higher area under the plots indicates better early retrieval performance.

Resistance to Noisy Stroke: The significance of stroke subset selector is quantitatively shown in Table 1. While it validates our potential under inherent low-magnitude noise existed in the dataset (shown in Fig. 5), we further aim to see how our method works on extreme noisy situation. In particular, we augment the training sketches by synthetic noisy patches, and train our subset selector with a pre-trained retrieval model. During testing, we synthetically add noisy strokes [29], and pass it through stroke-subset selector (pre-processing module) before feeding it to the retrieval model. While excluding the selector, the top@1 (top@5) drops to 13.4%(44.9%) in presence of synthetic noises, our stroke subset selector can improve them to 37.2%(68.2%) by eliminating the synthetic noisy strokes (see Fig. 6).

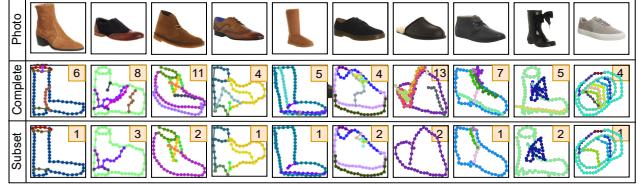


Figure 5: Examples showing selected subset performing better (rank in box) than complete sketch from ShoeV2.

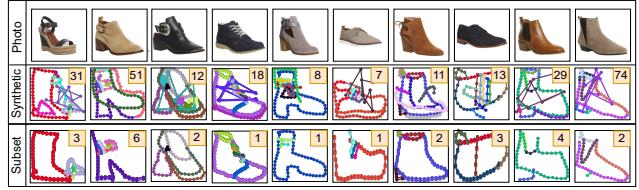


Figure 6: Examples showing ability to perform (rank in box) under synthetic noisy sketch input on ShoeV2.

Ablation on Design: (i) Instead of designing the stroke subset selector through hierarchical LSTM, another straight forward way is to use one layer bidirectional LSTM, where every coordinate point is being fed to each time step. However, the top@1(top@5) lags behind by 4.9%(6.7%) than ours, which verifies the necessity of hierarchical modelling of sketch vectors to consider the compositional relationship in our problem. Replacing LSTM by Transformer leads to no meaningful improvement in our case. (ii) Being a pre-processing step, we compare the extra time required for selecting the optimal stroke set. In particular, it adds extra 22.4% multiply-add operations and 18.3% extra CPU time compared standard baseline FG-SBIR. (iii) Compared to different RL methods [41], we get best results with PPO actor-critic version with clipped surrogate objective that beats its actor-only alternative by 1.7% top@1 accuracy(ShoeV2). Importantly, training with critic network leads to one important byproduct of modelling retrieval ability of partial sketches. (iv) Exploring different possible reward functions, we conclude that combining rewards from both ranking and feature embedding space through triplet loss gives most optimum performance than ranking only counterpart by extra 1.2% top@1 accuracy (ShoeV2). Please refer to supplementary for more details.

7. Conclusion

In this paper, we tackle the “fear to sketch” issue by proposing an intelligent stroke subset selector that automatically selects the most representative stroke subset from the entire query stroke set. Our stroke subset selector can detect and eliminate irrelevant (noisy) strokes, thus boosting performance of any off-the-shelf FG-SBIR framework. To this end, we designed an RL-based framework, which learns to form an optimal stroke subset by interacting with a pre-trained FG-SBIR model. We also show how the proposed selector can augment other sketch applications in a plug-and-play manner.

References

- [1] Emre Aksan, Thomas Deselaers, Andrea Tagliasacchi, and Otmar Hilliges. Cose: Compositional stroke embeddings. In *NeurIPS*, 2021. [4](#)
- [2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. [5](#)
- [3] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Aneeshan Sain, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. More photos are all you need: Semi-supervised learning for fine-grained sketch based image retrieval. In *CVPR*, 2021. [1, 2, 3, 4, 5, 6, 7](#)
- [4] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Vectorization and rasterization: Self-supervised learning for sketch and handwriting. In *CVPR*, 2021. [2, 3, 4](#)
- [5] Ayan Kumar Bhunia, Ayan Das, Umar Riaz Muhammad, Yongxin Yang, Timothy M Hospedales, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. Pixelor: A competitive sketching ai agent. so you think you can sketch? *ACM TOG*, 2020. [3, 4](#)
- [6] Ayan Kumar Bhunia, Viswanatha Reddy Gajjala, Subhadeep Koley, Rohit Kundu, Aneeshan Sain, Tao Xiang, and Yi-Zhe Song. Doodle it yourself: Class incremental learning by drawing a few sketches. In *CVPR*, 2022. [2](#)
- [7] Ayan Kumar Bhunia, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Sketch less for more: On-the-fly fine-grained sketch-based image retrieval. In *CVPR*, 2020. [1, 2, 3, 4, 5, 6, 7, 8](#)
- [8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. [7](#)
- [9] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *CVPR*, 2018. [2](#)
- [10] Pinaki Nath Chowdhury, Ayan Kumar Bhunia, Viswanatha Reddy Gajjala, Aneeshan Sain, Tao Xiang, and Yi-Zhe Song. Partially does it: Towards scene-level fg-sbir with partial input. In *CVPR*, 2022. [1](#)
- [11] John Collomosse, Tu Bui, and Hailin Jin. Livesketch: Query perturbations for guided sketch-based visual search. In *CVPR*, 2019. [1, 2](#)
- [12] Sounak Dey, Pau Riba, Anjan Dutta, Josep Llados, and Yi-Zhe Song. Doodle to search: Practical zero-shot sketch-based image retrieval. In *CVPR*, 2019. [1, 2](#)
- [13] 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](#)
- [14] Anjan Dutta and Zeynep Akata. Semantically tied paired cycle consistency for zero-shot sketch-based image retrieval. In *CVPR*, 2019. [1, 2](#)
- [15] 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. [3](#)
- [16] David Ha and Douglas Eck. A neural representation of sketch drawings. In *ICLR*, 2018. [3](#)
- [17] Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. In *NeurIPS*, 2018. [3](#)
- [18] 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](#)
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. [4](#)
- [20] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. [5](#)
- [21] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *ICCV*, 2019. [2](#)
- [22] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *JAIR*, 1996. [3, 5, 6](#)
- [23] Andrew Zisserman Karen Simonyan. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. [3](#)
- [24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. [6](#)
- [25] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *ECCV*, 2018. [1](#)
- [26] Xiaodan Liang, Lisa Lee, and Eric P Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *CVPR*, 2017. [3](#)
- [27] 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. [3](#)
- [28] Li Liu, Fumin Shen, Yuming Shen, Xianglong Liu, and Ling Shao. Deep sketch hashing: Fast free-hand sketch-based image retrieval. In *CVPR*, 2017. [2](#)
- [29] Runtao Liu, Qian Yu, and Stella X Yu. Unsupervised sketch to photo synthesis. In *ECCV*, 2020. [7, 8](#)
- [30] Umar Riaz Muhammad, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Goal-driven sequential data abstraction. In *ICCV*, 2019. [3](#)
- [31] Umar Riaz Muhammad, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning deep sketch abstraction. In *CVPR*, 2018. [3](#)
- [32] Radford M Neal. Annealed importance sampling. *Statistics and Computing*, 2001. [5, 6](#)
- [33] 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, 7](#)
- [34] 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](#)

- [35] Kaiyue Pang, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Solving mixed-modal jigsaw puzzle for fine-grained sketch-based image retrieval. In *CVPR*, 2020. 3, 6, 7
- [36] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. Sketchformer: Transformer-based representation for sketched structure. In *CVPR*, 2020. 1, 2
- [37] 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
- [38] Aneeshan Sain, Ayan Kumar Bhunia, Vaishnav Potlapalli, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Sketch3t: Test-time training for zero-shot sbir. In *CVPR*, 2022. 1
- [39] Aneeshan Sain, Ayan Kumar Bhunia, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. Cross-modal hierarchical modelling for fine-grained sketch based image retrieval. In *BMVC*, 2020. 2, 3, 6, 7
- [40] Aneeshan Sain, Ayan Kumar Bhunia, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. Stylemeup: Towards style-agnostic sketch-based image retrieval. In *CVPR*, 2021. 2, 3, 7
- [41] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 5, 7, 8
- [42] Yuming Shen, Li Liu, Fumin Shen, and Ling Shao. Zero-shot sketch-image hashing. In *CVPR*, 2018. 1, 2
- [43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6
- [44] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*, 2021. 3
- [45] Jifei Song, Yi-Zhe Song, Tony Xiang, and Timothy M Hospedales. Fine-grained image retrieval: the text/sketch input dilemma. In *BMVC*, 2017. 2, 4, 7
- [46] 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 *CVPR*, 2017. 1, 2, 7
- [47] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NeurIPS*, 2000. 5
- [48] Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. In *CVPR*, 2019. 3
- [49] Giorgos Tolias and Ondrej Chum. Asymmetric feature maps with application to sketch based retrieval. In *CVPR*, 2017. 2
- [50] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch your own gan. In *ICCV*, 2021. 2
- [51] 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
- [52] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 2009. 3
- [53] Peng Xu, Yongye Huang, Tongtong Yuan, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, Zhanyu Ma, and Jun Guo. Sketchmate: Deep hashing for million-scale human sketch retrieval. In *CVPR*, 2018. 2
- [54] Peng Xu, Chaitanya K Joshi, and Xavier Bresson. Multi-graph transformer for free-hand sketch recognition. *IEEE T-NNLS*, 2021. 3
- [55] Sasi Kiran Yelamarthi, Shiva Krishna Reddy, Ashish Mishra, and Anurag Mittal. A zero-shot framework for sketch based image retrieval. In *ECCV*, 2018. 2
- [56] 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, 4, 6, 7
- [57] Qian Yu, Jifei Song, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Fine-grained instance-level sketch-based image retrieval. *IJCV*, 2021. 2
- [58] 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. 2, 6
- [59] Jingyi Zhang, Fumin Shen, Li Liu, Fan Zhu, Mengyang Yu, Ling Shao, Heng Tao Shen, and Luc Van Gool. Generative domain-migration hashing for sketch-to-image retrieval. In *ECCV*, 2018. 1, 2
- [60] Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Robust curriculum learning: from clean label detection to noisy label self-correction. In *ICLR*, 2021. 3