

How to Handle *Sketch-Abstraction* in Sketch-Based Image Retrieval?

Subhadeep Koley^{1,2} Ayan Kumar Bhunia¹ Aneeshan Sain¹ Pinaki Nath Chowdhury¹
Tao Xiang^{1,2} Yi-Zhe Song^{1,2}

¹SketchX, CVSSP, University of Surrey, United Kingdom.

²iFlyTek-Surrey Joint Research Centre on Artificial Intelligence.

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

<https://subhadeepkoley.github.io/AbstractAway>

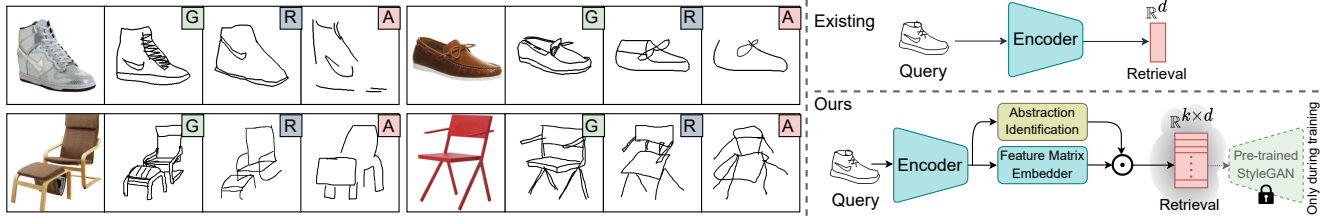


Figure 1. *Left*: Freehand sketches exhibit varied levels of abstraction – **G**: good, **R**: reasonable, **A**: abstract. *Right*: Unlike existing feature vector embedding, we learn a *feature matrix* representation in the joint embedding space, *regularised* by a *pre-trained StyleGAN*'s disentangled latent space, and an *abstraction-aware retrieval loss*. The *abstraction identification* head dynamically decides the row-dimension of the matrix embedding based on the query sketch abstraction/completeness.

Abstract

In this paper, we propose a novel *abstraction-aware sketch-based image retrieval framework* capable of handling sketch abstraction at varied levels. Prior works had mainly focused on tackling sub-factors such as drawing style and order, we instead attempt to model abstraction as a whole, and propose feature-level and retrieval granularity-level designs so that the system builds into its DNA the necessary means to interpret abstraction. On learning abstraction-aware features, we for the first-time harness the rich semantic embedding of pre-trained StyleGAN model, together with a novel abstraction-level mapper that deciphers the level of abstraction and dynamically selects appropriate dimensions in the feature matrix correspondingly, to construct a feature matrix embedding that can be freely traversed to accommodate different levels of abstraction. For granularity-level abstraction understanding, we dictate that the retrieval model should not treat all abstraction-levels equally and introduce a differentiable surrogate $\text{Acc.}@q$ loss to inject that understanding into the system. Different to the gold-standard triplet loss, our $\text{Acc.}@q$ loss uniquely allows a sketch to narrow/broaden its focus in terms of how stringent the evaluation should be – the more abstract a sketch, the less stringent (higher q). Extensive experiments depict our method to outperform existing state-of-the-arts in standard SBIR tasks along with challenging scenarios like early retrieval, forensic sketch-photo matching, and style-invariant retrieval.

1. Introduction

Unquestionably the biggest difference between sketches and photos lies with that of abstraction – sketches are ab-

stract depictions, photos being lifelike [39, 42, 69]. Abstraction in sketches is formed by many interlacing factors such as drawing skill, style, culture and subjective interpretation [13, 20, 40, 41, 47]. As a result, human sketches typically exhibit a large variation in terms of abstraction-levels – from good (art-trained), to reasonable (you), and those highly abstract (me!), as shown in Fig. 1(left).

The sketch community had been on a quest to tackle abstraction since inception [5, 47, 48, 56, 79], however mainly focusing on one sub-element at once (e.g., style [56]). This paper attempts for the first time to tackle sketch abstraction as a whole, for the problem of fine-grained sketch-based image retrieval (FG-SBIR) [8, 9, 12, 18, 19, 57, 58]. The end result is an abstraction-aware retrieval model that flexibly adapts to different levels of sketch abstraction while maintaining performance – it takes the abstraction away.

We operate under two guiding principles to tackle abstraction – on feature level, and on retrieval granularity – all to ensure our system has in its DNA means to accommodate all abstract forms of human sketches. On the former, we desire a flexible feature embedding that dynamically attends to the varying levels of abstraction. That is, under a highly abstract input sketch, we would like to return coarser features to conduct sketch-photo matching, and vice versa for those less abstract (more detailed). On the latter, we dictate that our system builds into its understanding, that retrieval granularity (i.e., how fine-grained the retrieval is) is *negatively* correlated with sketch abstraction-level. This is intuitive – one can not reasonably expect a rough contour sketch of a shoe, to magically retrieve a specific Jordan trainer.

To engineer the said abstraction-aware feature embed-

ding, we (i) make clever use of a pre-trained StyleGAN [36] model, which is shown via a pilot study (Sec. 4) to already exhibit a nicely disentangled coarse-to-fine latent feature embedding [78], to guide the construction of an abstraction-aware feature embedding for FG-SBIR such that it follows the same coarse-to-fine pattern, and (ii) design a novel abstraction-level mapper that first identifies the abstraction-level of a sketch, then dynamically selects dimensions in the feature embedding that correspond to the identified abstraction-level. These two designs elegantly interact with each other via a $k \times d$ feature *matrix* embedding, other than the conventional column *vector* [83]. It follows that by traversing across the rows of this matrix, one could easily vary the “fine-grainedness” of the sketch feature – *more* rows to encode *less* abstract sketches.

Embedding an understanding of retrieval granularity is largely analogous to that of the focus of a camera¹, but for a sketch – the more abstract the input sketch is, the less focused it should be. Translating to the retrieval setup, this means the system should be more tolerant given a highly abstract sketch, in terms of how well the sketch can perform retrieval, *i.e.*, the sketch becomes less “focused”. For that, we conduct another pilot study which shows the de facto triplet loss [83], the gold standard in the FG-SBIR community [9, 18, 57, 59, 64], is not fit for purpose. This is intuitive as well since triplet loss is solely concerned with bringing the best match to top-1, regardless of how good the sketch is. Instead, we propose a *differentiable* surrogate $\text{Acc} @ q$ loss mimicking actual test-time evaluation of FG-SBIR. It follows that q is our desired “focus” parameter – larger the q , lesser the “focus” (retrieval granularity).

In summary, (i) we for the first time tackle the abstraction problem as a whole for the task of FG-SBIR, (ii) we tailor abstraction understanding in our system’s DNA by specifically designing feature-level and retrieval granularity-level innovations, (iii) we utilise the information-rich StyleGAN [36] latent space to guide learning of abstraction-aware feature embedding, (iv) we propose a novel abstraction-level selector to dynamically assess levels of abstraction, and select appropriate levels in our feature matrix embedding, (v) we develop a differentiable $\text{Acc} @ q$ surrogate loss that allows the retrieval metric to accommodate different levels of abstraction. Comprehensive experiments show our method to surpass prior arts in a variety of FG-SBIR tasks such as early retrieval, forensic sketch-photo matching, and style-invariant retrieval.

2. Related Works

Sketch-Based Image Retrieval. Unlike category-level SBIR [21, 26, 45], *fine-grained*-SBIR (FG-SBIR) attempts at instance-specific sketch-photo correspondence [9, 11, 29, 56, 66, 76, 83]. The deep Siamese triplet network [83]

for FG-SBIR was further accelerated by attention-based higher-order loss [64], text tags [63], semi-supervised learning [9], or self-supervised pre-training strategies [50]. Although subsequent works have tackled partial [8] and noisy [11] sketches, hierarchical abstraction [55], sketch-style diversity [12, 56], the formal addressal of sketch abstraction in FG-SBIR remains unexplored till date.

Handling Sketch Abstraction. *Freehand sketch abstraction* [47], refers to the level of detail at which a sketch is represented [6] based on user’s expertise, preference, time-limit, or the task in hand [47]. Recent approaches for handling sketch abstraction can be broadly categorised as – (a) *Subset-selection* [47, 48], (b) *Parametric sketch representation* [22–24] (c) *Primitive-based* [5], and (d) *Abstraction-guided deep encoder* [79]. Under selection-based methods, Muhammad *et al.* [47, 48] used the hard assumption that any freehand sketch can be crudely approximated from a subset containing the *most salient* contours of the corresponding edgemap. Compared to 2D coordinates [9, 10], recent works focus on learning a better representation of strokes through *parametric* curves via Bézier control points [22, 67], or differential equations [23]. These methods modulate the degree of abstraction (smoothness of the curve) by either varying the cardinality of Bézier control point [22] or the sinusoidal frequency parameters of a differential equation [23]. Contrarily, shape-based methods [5] handle abstraction by representing a sketch with a predefined set of geometric *primitives*. Finally, instead of a standard ImageNet pre-trained backbone, Yang *et al.* [79] designed a sketch-specific *deep encoder* for granular learning of shape and appearance abstraction at varying levels. While the existing methods are driven by some heuristic [5] or hard assumption [47, 48], we aim to model the abstraction for FG-SBIR utilising the smooth latent space [77] of a pre-trained StyleGAN [36] and a parametric surrogate loss.

GAN for Vision Task. Primarily aimed at image generation, GAN has now become the de facto choice for various downstream tasks [2, 28, 82, 87]. Delving deeper, it was revealed that the layer-wise representations [77, 78] learned by StyleGAN [36] or BigGAN [14] disentangle object semantics at different abstraction hierarchy (*i.e.*, fine, medium, and coarse). Built on this disentanglement, GAN-inversion has paved the way for several downstream tasks [3, 4, 54]. While GAN has various sketch-related applications in image synthesis [16, 68], image/video editing [44, 80, 85], sketch completion [43], we exploit the disentanglement property of a pre-trained StyleGAN [36] to model sketch abstraction for FG-SBIR.

Loss Functions for FG-SBIR. FG-SBIR practically aims at ranking all photos of a gallery by their proximity to the query-sketch [83], with its paired photo ideally topping the rank list. Although test-time evaluation metrics (*e.g.*,

¹<https://tinyurl.com/3tp3b732>

Recall@q, Precision@q, Accuracy@q) are usual for FG-SBIR, optimising them directly via gradient descent is difficult in practice, as they invoke non-differentiability [15, 52], owing to operations like sorting [52], counting, etc. Consequently, state-of-the-arts [56, 57, 83] mostly employ triplet-based contrastive proxy loss that minimises the distance between anchor-sketch and paired positive photo, compared to a random negative photo. While a plethora of deep metric losses [30, 49, 71, 72] exist, triplet loss [61] has become a standard baseline owing to its ease of training, and performance [83]. This was enhanced further with a higher-order loss [64], quadruplet loss [63], meta-learning its margin value [12], or moving to a reinforcement learning (RL) based pipeline [8, 11] that optimises rank (non-differentiable) of the paired photo. RL being allegedly unstable [8], we take to *surrogate loss* literature [15] that aims for a *differentiable approximation* of evaluation metrics [15], to directly train a model. We thus leverage latent space of a pre-trained StyleGAN along with a differentiable approximation of retrieval-rank via a parametric surrogate loss, to develop an abstraction-aware FG-SBIR framework.

3. Backgrounds

Baseline FG-SBIR Model. Given an input image $\mathcal{I} \in \mathbb{R}^{H \times W \times 3}$, we obtain a d -dimensional l_2 normalised feature $f_i = \mathcal{F}_b(\mathcal{I}) \in \mathbb{R}^d$ using an ImageNet pre-trained VGG-16 [62] backbone network \mathcal{F}_b , sharing weights between sketch and photo branches. It trains over a triplet loss [70] that aims to minimise the distance $\delta(\cdot, \cdot)$ of a sketch-feature (f_s) from its paired positive photo-feature (f_p), while increasing that from a random negative (f_n) photo-feature. With margin $\mu > 0$, the triplet loss is formulated as:

$$\mathcal{L}_{\text{triplet}} = \max\{0, \mu + \delta(f_s, f_p) - \delta(f_s, f_n)\} \quad (1)$$

StyleGAN. In traditional GAN [53], a noise vector $z \in \mathcal{Z}$ of size \mathbb{R}^d sampled from a Gaussian distribution is fed to the input layer of the generator to obtain an RGB image. Contrarily, StyleGAN [36] first transforms a sampled $z \in \mathcal{Z}$ of size \mathbb{R}^d into an intermediate latent code [1] $w^+ \in \mathcal{W}^+$ of size $\mathbb{R}^{k \times d}$ using a non-linear latent mapper $f: \mathcal{Z} \rightarrow \mathcal{W}^+$ (an 8-layer MLP), where k depends on resolution ($M \times M$) of output image as $k = 2 \log_2(M) - 2$ [36]. The synthesis network $\mathcal{G}(\cdot)$ starts with a learned constant tensor of size $\mathbb{R}^{4 \times 4 \times d}$ and contains several progressive resolution blocks, with each block having the sequence $\text{conv}3 \times 3 \rightarrow \text{AdaIN} \rightarrow \text{conv}3 \times 3 \rightarrow \text{AdaIN}$ [36]. Each latent vector w_i^+ , upon passing through a common affine transformation layer A , generates the style $y = (y_s, y_b) = A(w_i^+)$. Using adaptive instance normalisation (AdaIN), y controls specifics of the output image at the i^{th} level of synthesis network [36] by modulating the feature-map x as $\text{AdaIN}(x, y) = y_s \frac{x - \mu(x)}{\sigma(x)} + y_b$. Despite being trained in an unsupervised manner, StyleGAN’s latent space holds

disentangled object representation at multiple abstraction-levels [78]. Pertaining to FG-SBIR, as no hard ground-truth supervision exists to model varying abstraction-levels, we use this disentanglement potential in context of FG-SBIR.

4. Pilot Study: Problems and Solutions

How disentangled is the StyleGAN latent space? We here aim to assess the extent of semantic feature disentanglement achievable via the StyleGAN latent code $w^+ = \{w_1^+, w_2^+, \dots, w_k^+\}$ through a simple GAN inversion. In GAN inversion, given a real input image x and an initial random latent code w^+ , the aim is to find a new latent code w_*^+ , that can most accurately reconstruct the input using a well-trained generator $\mathcal{G}(\cdot)$ via a reconstruction objective: $w_*^+ = \text{argmin}_{w^+} \mathcal{L}(x, \mathcal{G}(w^+))$ [1]. To figure out how different latent vectors in w^+ control certain specific attributes of the image, we segregate w^+ of size $\mathbb{R}^{14 \times d}$ (for 256×256 real image) into three categories: $w_{\text{coarse}}^+ = \{w_1^+ \rightarrow w_5^+\}$, $w_{\text{mid}}^+ = \{w_6^+ \rightarrow w_9^+\}$ and $w_{\text{fine}}^+ = \{w_{10}^+ \rightarrow w_{14}^+\}$. Given three latent groups $\{w_{\text{coarse}}^+, w_{\text{mid}}^+, w_{\text{fine}}^+\}$, we optimise *any two* at once, while the third is sampled randomly from Gaussian distribution. We consider all such *exhaustive* latent group combinations (3C_2) by repeating this experiment thrice with different seed values, which yields multiple reconstructed images as shown in Fig. 2. Following existing literature [78], we hypothesise that *every group controls a specific set of attributes in the output image*, and not optimising a particular latent group (instead sampling from Gaussian distribution) will cause significant output variations corresponding to that set of attributes [78]. For instance, not optimising w_{coarse}^+ produces overall shape variation (e.g., shoe type), while doing so for w_{mid}^+ , and w_{fine}^+ latent groups yield mid (e.g., lace, bootstrap) and fine (e.g., colour, appearance) level attribute variations, respectively.

This pilot study aims to motivate *why* the disentangled latent representation of a pre-trained StyleGAN is a good choice for handling abstraction as a whole, whereas, later our proposed methodology (Sec. 5) describes *how* we use a pre-trained StyleGAN to learn an abstraction-aware feature matrix embedding in the context of FG-SBIR.

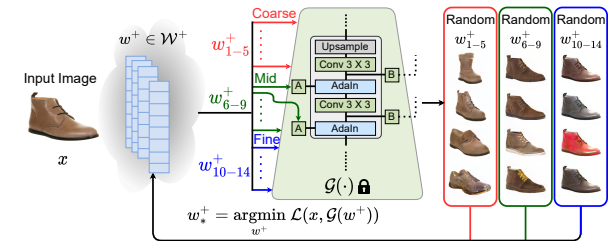


Figure 2. Pilot Study I: StyleGAN latent-disentanglement via optimising different groups of latent codes (coarse, medium, and fine).

Why model Abstraction for FG-SBIR? Highly abstract/partial early sketches may correspond randomly with multiple plausible photos from the gallery [8], and as the

drawing episode progresses towards completion, a sketch should start corresponding to a specific photo consistently [8]. We for the first time hypothesise this retrieval behaviour with a mathematical justification. Now, *entropy* being a faithful indicator of randomness [74], we use entropy of $\delta(\cdot, \cdot)$ in embedding space for a given query sketch s_t^q over successive ($t\% = 10\%, 20\%, \dots, 100\%$) levels of completion. Given a list of pre-computed photo features $\mathcal{P} = \{f_p^1, f_p^2, \dots, f_p^N\}$, and a sketch query feature $f_s^t = \mathcal{F}(s_t^q)$ at $t\%$, we compute a pairwise distance list as $\{\delta(f_s^t, f_p^1), \delta(f_s^t, f_p^2), \dots, \delta(f_s^t, f_p^N)\}$. Similarly, the distance from paired positive photo feature f_p^* is given as $\delta(f_s^t, f_p^*)$. We define the *similarity* between sketch query and positive photo as a probability distribution over all the gallery photos at $t\%$ as:

$$p_t = \exp(-\delta(f_s^t, f_p^*)) / \sum_{i=1}^N \exp(-\delta(f_s^t, f_p^i)) \quad (2)$$

Following our hypothesis, as the sketch approaches completion ($t \rightarrow 100\%$), the distance between sketch query and positive photo should ideally decrease with respect to other non-matching gallery photos. This would lead to an increase in p_t , thus decreasing entropy $\mathcal{H} (= -p_t \log p_t)$. However, Fig. 3 shows that the baseline FG-SBIR model [83] deviates from our ideal hypothesis, depicting an arbitrary entropy behaviour, hampering performance. This calls for explicit abstraction-modelling for sketches. Accordingly, when this is addressed in our model, it depicts a *consistent* entropy-decrease, thus following our hypothesis in guiding the sketch query towards a specific gallery instance as sketching reaches towards completion.

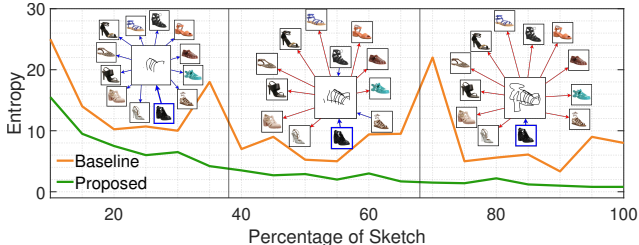


Figure 3. Pilot Study II: Compare retrieval consistency by comparing entropy of separation in the embedding space, evaluated over successive stages of sketch completion. Inset images show how our method directs the query to a single gallery image (blue) while pushing others away as sketching progresses.

5. Proposed Methodology

Overview. Our initial pilot study directs us towards two specific design elements: (i) leveraging a pre-trained StyleGAN [36] during training, and (ii) modelling the varied levels of sketch-abstraction via a novel loss function. Firstly, compared to the embedding feature *vector* of size \mathbb{R}^d of baseline model [83], we aim to use a feature *matrix* embedding of size $\mathbb{R}^{k \times d}$ for sketch/photo representation in the joint-embedding space. In particular, we distil the disentangled knowledge [78] residing inside a pre-trained StyleGAN’s [37] latent space into our matrix embedding. Here

specific rows of the matrix govern a few particular characteristics of the input, where moving down the rows leads from coarse to finer-level features [78]. Furthermore, our model automatically decides up to which row of the matrix embedding will be used for retrieval. Secondly, as triplet-based models [83] treat every sketch sample equally, ignoring their abstraction-level, we design a parametric surrogate loss that *accounts* for *individual* abstraction-level, thus leading towards a more practical training pipeline.

5.1. Model Architecture

The core of our model includes three major components: (i) discovering which of the $\mathbb{R}^{k \times d}$ StyleGAN [37] latent vectors are representative enough in context of FG-SBIR, (ii) modifying the feature embedding network to generate *matrix* instead of *vector* embedding, (iii) adaptively selecting the number of rows of the feature matrix embedding based on the input sketch abstraction for faithful retrieval.

Sketch-Specific Latent Groups. Representing 256×256 images in StyleGAN [36] latent space would require latent codes of size $\mathbb{R}^{14 \times d}$. As evident from our pilot study, initial latent vectors $\{w_1^+ \rightarrow w_9^+\}$ control the major *shape* and *structure*, while the later ones $\{w_{10}^+ \rightarrow w_{14}^+\}$ only govern the *colour* and *appearance* of the output. Likewise, *sketch* being sparse binary line drawings, can only provide *structural cues* in context of FG-SBIR [56]. Accordingly, during training, we exclusively use the structural latent vectors $\{w_1^+ \rightarrow w_9^+\}$, resulting in a feature matrix embedding of size $\mathbb{R}^{9 \times d}$. We further segregate the structural latent vectors into three sub-groups, each with a set of three latent vectors $\{w_{1 \rightarrow 3}^+, w_{4 \rightarrow 6}^+, w_{7 \rightarrow 9}^+\}$ to model a *knob* for varying the level of sketch *abstraction*. For the rest of the paper, we name the groups as $a = \{a_c, a_m, a_f\}$ for the brevity of description.

Architecture. A common backbone feature extractor $\mathcal{F}(\cdot)$ initialised from an ImageNet pre-trained VGG-16 [62] is used to obtain a feature-map of size $\mathcal{F}(I) \in \mathbb{R}^{h \times w \times d}$ for both sketch (s) and photo (p) branches (Fig. 4). To tackle the large domain gap between pixel-perfect photos and sparse sketches [55], we apply individual sketch and photo specific *feature matrix* embedding networks \mathcal{E}_s and \mathcal{E}_p on the output of $\mathcal{F}(I)$. Feature matrix embedding networks $\mathcal{E}_s, \mathcal{E}_p : \mathbb{R}^{h \times w \times d} \rightarrow \mathbb{R}^{9 \times d}$, each contains 9-individual stride-two convolution blocks with LeakyReLU [75], applied over $\mathcal{F}(I)$. Each of them predicts a d -dimensional feature, which upon concatenating across 9-blocks gives the sketch feature matrix as $m_s = \mathcal{E}_s(\mathcal{F}(s)) \in \mathbb{R}^{9 \times d}$ and photo feature matrix as $m_p = \mathcal{E}_p(\mathcal{F}(p)) \in \mathbb{R}^{9 \times d}$, respectively.

Modelling Sketch Abstraction. In context of FG-SBIR, we model sketch abstraction by dynamically deciding the number of row vectors from m_s and m_p to be used for distance calculation, where traversing from top to bottom acts as a *tunable knob* controlling the degree of abstraction. Intuitively, for very coarse abstract sketches, we want to re-

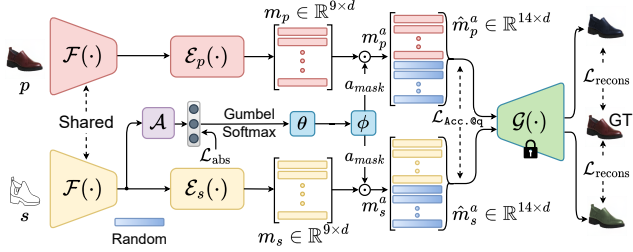


Figure 4. Our method learns a *feature matrix* representation in the joint embedding space, regularised by a pre-trained StyleGAN, trained with a weighted summation of reconstruction, abstraction identification, and Acc.@q losses. θ :flip(cumsum(flip(\cdot))), ϕ :repeat and flattening operation (more in text).

trieve using only the $\{a_c\}$ group of latent vectors, while for sketches with medium and finer-grained details, we aim to retrieve using $\{a_c, a_m\}$ and $\{a_c, a_m, a_f\}$ groups respectively. Therefore, we add an abstraction identification head $\mathcal{A} : \mathbb{R}^d \rightarrow \mathbb{R}^3$ that takes a global average pooled sketch feature from the backbone network as input, and predicts the probability distribution over three latent groups as $\hat{a} = p(a|s) = \mathcal{A}(\mathcal{F}(s))$ for a specific sketch input. After passing through a differentiable argmax (realised with Gumbel-Softmax [34]), \hat{a} gives a one-hot encoded vector depicting the abstraction-level. In order to generate the *latent selection mask* differentiably, we apply cumulative summation as follows $a_{\text{mask}} = \text{flip}(\text{cumsum}(\text{flip}(\hat{a})))$, thus obtaining three possible mask states as $(\hat{a}_c, \hat{a}_m, \hat{a}_f) \in \{(1, 0, 0), (1, 1, 0), (1, 1, 1)\}$. In order to get the equivalent 9-dimensional mask \tilde{a}_{mask} , we can simply use cascaded repeat and flattening operations² from any standard deep-learning library. Therefore given a sketch input, we can get the *abstraction-aware* feature matrix embeddings as $m_s^a = m_s * \tilde{a}_{\text{mask}}$ and $m_p^a = m_p * \tilde{a}_{\text{mask}}$, where based on the predicted \tilde{a}_{mask} , the later rows of m_p^a, m_s^a might get populated with zero values.

The question remains as to how we obtain the training signal to learn our abstraction identification head \mathcal{A} . Keeping the *sequentially* abstract nature of sketches [8] in mind, we thus make a logical assumption of associating the coarse, medium and fine levels with vector sketches rendered at 30%, 60% and 100% completion as one-hot encoded ground truth a_{gt} for $\{a_c, a_m, a_f\}$ respectively.

5.2. Loss Objectives

Reconstruction Loss. To distil the knowledge from a pre-trained StyleGAN [37], we use sketch-to-photo and photo-to-photo reconstruction losses. We pad the n non-zeros rows of m_s^a , and m_p^a with $(14 - n)$ d -dimensional vectors sampled from Gaussian distribution to obtain final matrix embeddings \hat{m}_s^a , and \hat{m}_p^a of size $\mathbb{R}^{14 \times d}$, as the pre-trained StyleGAN always expects a $\mathbb{R}^{14 \times d}$ latent code to generate an output. Upon passing \hat{m}_s^a , and \hat{m}_p^a through the pre-

²For example, $\{1, 1, 0\} \rightarrow \{1, 1, 1, 1, 1, 1, 0, 0, 0\}$

trained StyleGAN, we compute the reconstruction loss as:

$$\mathcal{L}_{\text{recons}} = \|p - \mathcal{G}(\hat{m}_s^a)\|_2 + \|p - \mathcal{G}(\hat{m}_p^a)\|_2 \quad (3)$$

Using this reconstruction objective, we aim to align the sketch/photo feature matrix embedding to that of the disentangled latent space of pre-trained StyleGAN [37], thus encouraging for a feature matrix where the abstraction-level decreases from *coarse* to *finer* across the rows.

Differentiable Acc.@q Retrieval Loss. Pre-trained StyleGAN [36]-based reconstruction loss alone, is not enough for discriminative learning in the feature matrix embedding space for cross-modal retrieval. As handling sketch abstraction is limited by triplet loss (Sec. 4), we introduce a novel surrogate proxy loss [15, 52] – Acc.@q. Mimicking a standard evaluation-metric [83], Acc.@q measures the percentage of query samples having true match gallery photos appearing in the top- q list. Varying q as 10, 5, and 1 helps to model coarse (30%), mid (60%) and fine (100%) levels of abstraction $\{a_c, a_m, a_f\}$ respectively. For instance, the initial *coarse* sketch may correspond to multiple gallery photos where having the true match in the top-10 ($q=10$) list might suffice, whereas for a complete (*fine*) sketch, we need to enforce retrieval at top-1 ($q=1$) position.

As computing rank across an entire dataset is impractical, we opt for a more reasonable choice of calculating Acc.@q across a batch. Given paired sketch-photo feature matrix embeddings across a batch with batch-size B as $\{(m_{s,i}^a, m_{p,i}^a)\}_{i=1}^B$, we first compute the pairwise-distances of a given sketch query $m_{s,i}^a$ with all the photos in a batch as $\mathcal{D} = \{\delta_{s,i}^{p,1}, \dots, \delta_{s,i}^{p,B}\}$, and true target distance with the paired photo as $\delta_{s,i}^{p,*}$. The relative distance vector now becomes $\mathcal{D}_{\text{rel}} = \delta_{s,i}^{p,*} - \mathcal{D}$. To calculate the rank of paired photo for i^{th} sketch query, we need to find the number of positive values in $\mathcal{D}_{\text{rel}}^i$, which requires a *non-differentiable* step function [15, 52] that we approximate via a temperature (τ) controlled Sigmoid function $S_\tau(x, \tau) = \frac{1}{1 + \exp(-x/\tau)}$. Once we find $\text{rank}_i = \sum S_{\tau_2}(\mathcal{D}_{\text{rel}}^i)$, Acc.@q for that sketch query (either +1 or 0) can easily be calculated using $S_{\tau_1}(q - \text{rank}_i)$. Thus the overall Acc.@q loss becomes:

$$\mathcal{L}_{\text{Acc.@q}} = -\frac{\sum_{i=1}^B S_{\tau_1}(q - \sum_{i=1}^B S_{\tau_2}(\mathcal{D}_{\text{rel}}^i))}{B} \quad (4)$$

A lower (*higher*) value of τ leads to a better (*worse*) approximation of the step function but results in a sparser (*denser*) gradient [15]. Empirically, we find $\tau_1 = 1$ and $\tau_2 = 0.01$ to be optimum. A PyTorch-like pseudocode has been provided in Algorithm 1 (for brevity, only \mathbb{R}^d case is shown).

Abstraction Identification Loss. Abstraction identification is a three-class classification problem that helps to *dynamically decide* the degree of abstraction of the input sketch, thus dictating the cardinality of effective rows in the feature matrix embedding. Using the vector sketch rendered at 30%, 60% or 100% completion-levels as ground truth a_{gt}

Algorithm 1 PyTorch-like code for differentiable $\text{Acc}@q$

```
import torch
import torch.nn.functional as F
import F.pairwise_distance as PD

def accuracy@q_loss(anchor, pos, q):
    # anchor: BXd; pos: BXd; q: 1x1(Acc.@q)
    # B←batch size; d←feature dimension
    acc.q = torch.tensor(0) # Accumulate accuracy
    dist = PD(anchor, pos) # BxB

    for i in range(len(anchor)):
        # Target distance for  $i^{th}$  anchor
        tgt_dist = PD(anchor[i], pos[i]) # 1x1
        # Relative distance to target
        rel_dist = tgt_dist - PD(anchor[i], pos) # Bx1

        signs = sigmoid(rel_dist, 0.01)
        rank_i = torch.sum(signs) # Get rank: 1x1
        acc.q += sigmoid(q - rank_i, 1)

    batch_acc = acc.q / len(anchor)
    loss = -1 * batch_acc
    return loss

# Approximate step function with Sigmoid
def sigmoid(tensor, tau):
    tensor = tensor / tau
    y = 1.0 / (1.0 + torch.exp(-tensor))
    return y
```

for coarse, medium and fine categories respectively, our abstraction identification loss becomes:

$$\mathcal{L}_{\text{abs}} = -\frac{1}{3} \sum_{i=1}^3 a_{gt}^i \log(\hat{a}^i) \quad (5)$$

In summary, our overall training objective can be defined as: $\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{recons}} + \lambda_2 \mathcal{L}_{\text{Acc}@q} + \lambda_3 \mathcal{L}_{\text{abs}}$. During inference, we *discard* the StyleGAN [37] generator module. The shared feature extractor (\mathcal{F}) followed by the feature matrix encoders ($\mathcal{E}_s, \mathcal{E}_p$) and the abstraction identification head (\mathcal{A}) is used to generate abstraction-aware feature matrix embeddings m_s^a and m_p^a . We calculate the distance between m_s^a and m_p^a to perform the retrieval during inference.

6. Experiments

Dataset. Following standard literature [8, 55], we use QMUL-ShoeV2 [8, 65] and QMUL-ChairV2 [8, 65] datasets, to evaluate FG-SBIR performance. They contain 6730 (2000) and 1800 (400) sketches (photos) respectively, with multiple sketches per photo with fine-grained association. While 6051 (1275) sketches and 1800 (300) photos from ShoeV2 (ChairV2) are used for training respectively, the rest are used for testing. We use photos from the UT Zappos50K [81] dataset for pre-training StyleGAN generator on the *Shoe* class. Due to the lack of large-scale chair image datasets, we use over 10,000 chair photos collected from shopping websites like IKEA, ARGOS, and Amazon to pre-train the StyleGAN generator on the *Chair* class.

Implementation Details. We use the Adam [38] optimiser to pre-train the StyleGAN generator for $8M$ iterations at a learning rate of 10^{-3} and a batch size of 8. Empirically, we disable path-length regularisation and reduce $R1$ regularisation’s weight to 2 for stabler training with diverse output.

We employ an ImageNet pre-trained VGG-16 [62] as the feature backbone and a StyleGAN2 [37] model, both with an embedding dimension $d = 512$. Our FG-SBIR model was trained for $3K$ epochs using the Adam [38] optimiser with a constant learning rate of 10^{-3} and batch size of 128. Values of $\lambda_{1,2,3}$ are set to 0.5, 1, and 0.5, empirically.

Evaluation. Aligning with recent literature [9, 18], we evaluate our work on the standard FG-SBIR evaluation metric – $\text{Acc}@q$, which measures the percentage of sketches having a true-paired photo in the top- q retrieved list.

Human Study. In this study, upon drawing an abstract sketch, human participants had to rate the top-ranked photo retrieved by every competing framework on a scale of 1 to 5 (bad→excellent) [32], based on their *opinion* of how closely the retrieved photo matched the *imagination* of their previously drawn sketch. Each of the 72 participants was asked to draw and rate 50 such complete sketches, resulting in a total of 3600 responses per method. For each method, we compute the final MOS value by taking the mean (μ) and variance (σ) of all its MOS responses.

Competitors. We evaluate the proposed method from three perspectives – (i) *Existing state-of-the-arts (SoTA): Triplet-SN* [83] uses Sketch-a-Net [84] backbone with standard triplet loss. **HOLEF-SN** [64] further extends [83] with spatial attention-based higher-order loss. (ii) *Baselines:* Owing to the lack of generative model-based FG-SBIR frameworks, we design three baselines. **B-pSp** first leverages a StyleGAN [37]-based pSp [54] model (pre-trained on ShoeV2 and ChairV2) to translate an input sketch to its photo domain. Next, it finds the matching feature in its nearest neighbourhood in the entire photo gallery using an ImageNet pre-trained VGG-16 [62] feature extractor. **B-pix2pix** and **B-CycleGAN** follows the same paradigm as B-pSp, except using pre-trained pix2pix [33] and CycleGAN [88] models respectively for sketch-photo translation. These baselines essentially transform *sketch-based* image retrieval into an *image-based* image retrieval problem. (iii) *Partial/Style-invariant/Early FG-SBIR methods:* Here we compare the proposed method with recent state-of-the-art FG-SBIR methods like **Partial-OT** [18], **Cross-Hier** [55], **StyleMeUP** [56], **On-the-fly** [8], and **SketchPVT** [59].

6.1. Quantitative Analysis

Comparative results on FG-SBIR is delineated in Tab. 1. Although HOLEF-SN [64] performs better than Triplet-SN [83] due to the adaptation of the attention module, both fare poorly when compared to ours, owing to their straightforward adaptation of a weaker backbone [84]. Generative baselines like B-pix2pix or B-CycleGAN fails to reach the performance of Triplet-SN [83] due to the inefficiency of standard image translation models like pix2pix [33] or CycleGAN [88] in sketch-to-photo synthesis task [46]. On the other hand, our method outperforms B-

pix2pix and B-CycleGAN with a significant Acc.@1 margin of 18.6 (27.9)% and 17.5 (27.0)% in ShoeV2 (ChairV2) dataset respectively. B-pSp depicts sub-optimal Acc.@1 of 47.8 (30.1)% in ShoeV2 (ChairV2) due to its naive adaptation of StyleGAN [37]. Later methods attempt to address multiple different traits of freehand sketch like partiality [18], hierarchical-abstraction [55], style-variation [56], early-retrieval [8] for better accuracy. Surprisingly, our abstraction-aware model shows improved performance without the complicated optimal-transport of Partial-OT [18], costly hierarchical co-attention of CrossHier [55], unstable meta-learning of StyleMeUp [56], or time-consuming RL-finetuning of On-the-fly [8]. SketchPVT [59], despite being the strongest replacement of ours, scores less than ours in every experimental setup. Furthermore, as discussed in the pilot study (Sec. 4), our method achieves an average entropy of 3.88 compared to 7.98, 8.71, and 10.11 of StyleMeUP [56], On-the-fly [8], and triplet-based baseline [83]. Most importantly, we evaluate FG-SBIR via a human study, where our method outperforms prior arts with a massive MOS value of 4.1 ± 0.5 on the ShoeV2 dataset.

Table 1. Results for standard FG-SBIR task.

Methods	ChairV2			ShoeV2		
	Acc.@1	Acc.@5	MOS $\mu \pm \sigma$	Acc.@1	Acc.@5	MOS $\mu \pm \sigma$
Triplet-SN [83]	47.4	71.4	2.6±0.4	28.7	63.5	2.2±0.2
HOLEF-SN [64]	50.7	73.6	2.7±0.6	31.2	66.6	2.5±0.1
B-pSp	47.8	70.2	2.9±0.2	30.1	63.9	2.8±0.9
B-pix2pix	44.2	66.9	2.8±0.7	26.7	60.2	2.3±0.1
B-CycleGAN	45.1	67.1	2.7±0.3	27.8	62.6	2.5±0.4
Partial-OT [18]	63.3	79.7	3.6±0.7	39.9	68.2	3.5±0.6
CrossHier [55]	62.4	79.1	3.5±0.3	36.2	67.8	3.3±0.1
StyleMeUp [56]	62.8	79.6	3.7±0.2	36.4	68.1	3.3±0.3
On-the-fly [8]	51.2	73.8	3.2±0.5	30.8	65.1	2.8±0.5
SketchPVT [59]	71.2	80.1	3.3±0.7	44.1	70.8	2.9±0.9
Proposed	72.1	80.9	4.4±0.5	45.3	77.3	4.1±0.5

6.2. Sketch Abstraction Analysis

As there exists no *hard ground truth* or *standard metric* of quantification [47, 48] for *sketch abstraction*, we verify the generalisability of our method against different forms of abstraction (*i.e.*, style variations [56], method variations [5, 47, 48], partial/early retrieval [8, 18]).

Sketching Style Variations. Differences in abstraction-level may arise from *dissimilar drawing styles* while sketching the same object [56] (See Fig. 5). The baseline Triplet-SN [83] fails for coarsely drawn abstract sketches due to the lack of explicit abstraction-handling. Nonetheless, our StyleGAN [37]-regularised abstraction-aware model dynamically attends to varying abstraction-levels and retrieves the true match photo irrespective of the input drawing style variations. This further underpins our motivation of being truly abstraction-aware. In Fig. 5, the number beside each sketch denotes the number of row vectors from m_s and m_p chosen dynamically by the abstraction identification head \mathcal{A} . Notably, for every input, the proposed \mathcal{A} module *correctly* identifies the abstraction-level and selects the appropriate

number of row vectors to encode the sketch.



Figure 5. Proposed (blue) method’s efficacy over Triplet-SN [83] (green) against different sketching styles of the same shoe (red bordered). Zoom in for the best view. (More in § Supplementary.)

Abstraction Method Variations. Following [5], we evaluate our method by abstracting the input sketches via different methods (*e.g.*, PMN [5], DSA [47], GDSA [48]) and budgets ($\{10, 20, 30, 100\}\%$), corresponding to different abstraction-levels. Accordingly, the proposed method outperforms (Tab. 2) other competing SoTAs [8, 83] even when sketches are abstracted at 10% budget, thus being in tune with our motivation of being agnostic of the abstraction methods. Please note, all competing methods in Tab. 2 are trained on the original ShoeV2 sketch-image pairs.

Table 2. Acc.@10 at various abstraction budgets (%) and methods on ShoeV2. (Qualitative results in § Supplementary.)

Abs. Methods	Triplet-SN [83]				On-the-fly [8]				Proposed			
	10	20	30	100	10	20	30	100	10	20	30	100
PMN [5]	9.4	19.3	33.8	68.4	10.1	21.6	40.8	71.9	51.8	66.1	68.1	82.3
DSA [47]	10.2	20.3	32.2	79.6	15.2	30.9	51.1	79.6	38.2	42.2	50.2	85.1
GDSA [48]	11.5	22.8	34.1		17.1	31.4	52.9		41.7	48.0	58.3	

Partial/Early Retrieval. To evaluate the abstraction-aware behaviour of our method in the form of *partial/early retrieval* setup, we use two curves namely (a) ranking-percentile (m@A) and (b) $\frac{1}{rank}$ (m@B) vs. percentage of sketch [8]. Standard methods for handling early-retrieval include Triplet-SN [83] with random stroke-dropping, RL fine-tuning of triplet model [8], or RL-based salient stroke-subset selection [11]. While high-performing RL-based methods [8, 11] are allegedly unstable, random stroke-dropping produces noisy gradients [11]. Contrarily, our Acc.@q models a similar ranking-based objective as the RL-based method [8] *without* extra overheads, by varying q to better model partial sketches. Thanks to Acc.@q, our model outperforms (Fig. 6) On-the-fly [8], and Subset [11]. Overall, our method achieves a higher m@A (m@B) of 86.22 (22.30) as compared to 85.38 (21.24) and 85.78 (21.1) claimed in [8] and [11] respectively.

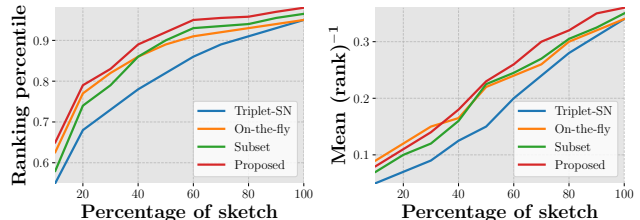


Figure 6. Quantitative results on ShoeV2 for early retrieval setup, visualised via the percentage of sketch. A higher area under the curve indicates better early retrieval performance.

6.3. Ablation

[i] Contribution of Acc. @q and Reconstruction Loss.

To estimate the efficacy of the proposed surrogate loss function, we replace Acc. @q with a standard triplet-based [83] objective. Replacing Acc. @q causes a sharp MOS value decrease of 1.7 ± 0.1 on ShoeV2 (Tab. 3). We posit that this drop is due to the absence of the test-time metric like ranking guidance provided by the surrogate loss. Additionally, replacing Acc. @q with SoTA surrogate losses like SoDeep [27] and SmoothAP [15] plummets MOS value by 1.1 ± 0.2 and 0.9 ± 0.1 respectively on ShoeV2. Moreover, as evident from the **w/o recons. loss** result, exclusion of the reconstruction loss destabilises the system, causing an additional Acc.@1 drop of 10.1% on the ShoeV2 dataset. Furthermore, fine-tuning the StyleGAN model (instead of frozen), results in lower Acc.@1 (30.2% on ShoeV2), likely because fine-tuning sacrifices the latent space disentanglement.

Table 3. Ablation on design.

Methods	ChairV2			ShoeV2		
	Acc.@1	Acc.@5	MOS $\mu \pm \sigma$	Acc.@1	Acc.@5	MOS $\mu \pm \sigma$
w/o abs. identification	59.2	71.9	3.0±0.1	38.1	69.2	2.5±0.3
w/o Acc. @q loss	58.5	70.1	2.8±0.3	37.1	68.1	2.4±0.4
w/o recons. loss	56.7	69.6	2.3±0.2	35.2	66.1	2.0±0.1
Ours+SoDeep [27]	60.3	74.1	2.7±0.4	39.6	71.3	3.0±0.3
Ours+SmoothAP [15]	61.7	74.9	3.1±0.2	40.3	73.9	3.2±0.4
Ours-full	72.1	80.9	4.2±0.3	45.3	77.3	4.1±0.5
Avg. Improvement	+11.1	+6.4	+1.3±0.0	+5.3	+4.7	+1.0±0.1

[ii] Importance of Abstraction Identification (Eq. 5).

To illustrate the relevance of our abstraction identification head, instead of three-stage rendering, we rasterise the full sketch vector at once, prior to training, thus using the entire feature matrix embedding of size $\mathbb{R}^{9 \times d}$ regardless of the input abstraction. Although less reflected in Acc.@1, a significant MOS drop of 1.6 ± 0.2 on ShoeV2 (Tab. 3) in case of **w/o abs. identification** signifies that sketch abstraction-modelling is incomplete without learning to predict the input abstraction-level via partial rendering.

[iii] Efficiency Analysis. As the gallery photo features are *pre-computed* during inference, the only difference in efficiency between a standard VGG-16 triplet model [83] and ours will arise from the additional *sketch feature matrix embedding network* \mathcal{E}_s . While the former takes 40.18G FLOPs to process a query-sketch of size 256×256 , we use 40.20G FLOPs, which is *only* 0.05% higher yet boosts accuracy by 16.6%. Secondly, unlike the \mathbb{R}^d feature vector of triplet-based models [83], we use a $\mathbb{R}^{9 \times d}$ embedding matrix for distance calculation during inference. Thus, *theoretically*, a triplet model should be $9 \times$ faster. However, in a practical FG-SBIR inference pipeline of *feature extraction* followed by *distance calculation* and *ranking*, our matrix distance calculation takes only a *negligible* 0.04% (0.03ms) out of a total of 77.1ms, on a 12 GB Nvidia 2080 Ti GPU. Besides, as the pre-trained StyleGAN is *used only during training*,

its architectural complexities *do not* affect the inference.

[iv] Latent Sub-grouping. We segregate the structural latent vectors into three sub-groups, each with a set of *three* latent vectors $\{w_{1 \rightarrow 3}^+, w_{4 \rightarrow 6}^+, w_{7 \rightarrow 9}^+\}$ to model varying the sketch abstraction-levels (Sec. 5). Testing with most such group combinations, we find that the $\{a_c, a_m, a_f\} = \{3, 3, 3\}$ combination is the most *generalised* and gives the *optimum* Acc.@1 (45.3%) compared to others (e.g., $\{2, 3, 4\}$ –39.8%, $\{2, 2, 5\}$ –33.6%, $\{1, 2, 6\}$ –28.9%, etc.).

6.4. Extension to Forensic Sketch-Photo Matching

Forensic sketch-photo matching is among the many practical applications of SBIR, where the system uses a forensic facial sketch as a query to recognise a person within a vast gallery (closed-set) of images [7, 17]. Forensic-sketching depicts varying levels of abstraction based on the *drawing skill* (or the lack of it) of the artist [7, 25]. Here we explore the potential of the information-rich latent space of StyleGAN [37] for solving the challenging problem of forensic sketch-photo recognition. Using the large-scale FFHQ [36] pre-trained StyleGAN model, we learn the backbone and the feature matrix encoder leveraging the IIIT-D viewed-sketch dataset (Fig. 7) [7], which includes *only one* sketch-photo pair for each of 238 different human subjects. Besides the standard Acc.@1, following the evaluation paradigm of [73], we also use the verification rate (VR) @ false acceptance rate (FAR) = 1% metric. Compared to other SoTA methods with complex architectures [17, 25, 73], handling data scarcity by training their models on the *much larger* CUFSF [86] dataset, our method despite being supervised with *only one* sketch-photo pair per subject, surpasses (Tab. 4) them with an average Acc.@1 gain of 10.77%.



Figure 7. Sketch-photo samples from the IIIT-D [7] dataset.

Table 4. Results for forensic sketch-photo recognition.

Methods	Acc.@1	VR @FAR=1%
RCN [25]	63.83	90.12
Deep-Face [51]	80.89	72.08
Center-Loss [71]	84.07	76.20
CDL [73]	85.35	82.52
RGM [17]	88.94	97.87
Proposed	91.39	97.38

7. Conclusion and Future Works

The proposed generation-guided FG-SBIR method along with a differentiable surrogate ranking loss explicitly handles freehand sketch abstraction, allowing it to retrieve accurate images even from highly *abstract* sketches. Extensive evaluations outperforming existing SoTAs depict the efficacy of StyleGAN’s disentangled latent space in abstraction-aware fine-grained retrieval tasks. Our method could further be extended to scene-level SBIR with scene-level GANs [35, 60]. Moreover, the synergy between freehand sketches and pre-trained GAN could pave the way for different weakly-supervised sketch-based applications like object co-part segmentation [31], image editing [54], etc.

References

- [1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space? In *CVPR*, 2019. 3
- [2] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Labels4Free: Unsupervised Segmentation Using StyleGAN. In *CVPR*, 2021. 2
- [3] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Only a Matter of Style: Age Transformation Using a Style-Based Regression Model. *ACM TOG*, 2021. 2
- [4] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement. In *CVPR*, 2021. 2
- [5] Stephan Alaniz, Massimiliano Mancini, Anjan Dutta, Diego Marcos, and Zeynep Akata. Abstracting Sketches through Simple Primitives. In *ECCV*, 2022. 1, 2, 7
- [6] Itamar Berger, Ariel Shamir, Moshe Mahler, Elizabeth Carter, and Jessica Hodgins. Style and Abstraction in Portrait Sketching. *ACM TOG*, 2013. 2
- [7] Himanshu S Bhatt, Samarth Bharadwaj, Richa Singh, and Mayank Vatsa. Memetically Optimized MCWLD for Matching Sketches With Digital Face Images. *IEEE TIFS*, 2012. 8
- [8] 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
- [9] 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, 6
- [10] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Yongxin Yang, Timothy Hospedales, Tao Xiang, and Yi-Zhe Song. Vectorization and Rasterization: Self-Supervised Learning for Sketch and Handwriting. In *CVPR*, 2021. 2
- [11] Ayan Kumar Bhunia, Subhadeep Koley, Abdullah Faiz Ur Rahman Khilji, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Sketching Without Worrying: Noise-Tolerant Sketch-Based Image Retrieval. In *CVPR*, 2022. 2, 3, 7
- [12] Ayan Kumar Bhunia, Aneeshan Sain, Parth Shah, Animesh Gupta, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Adaptive Fine-Grained Sketch-Based Image Retrieval. In *ECCV*, 2022. 1, 2, 3
- [13] Ayan Kumar Bhunia, Subhadeep Koley, Amandeep Kumar, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Sketch2Saliency: Learning to Detect Salient Objects from Human Drawings. In *CVPR*, 2023. 1
- [14] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *ICLR*, 2019. 2
- [15] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-AP: Smoothing the Path Towards Large-Scale Image Retrieval. In *ECCV*, 2020. 3, 5, 8
- [16] Wengling Chen and James Hays. SketchyGAN: Towards Diverse and Realistic Sketch to Image Synthesis. In *CVPR*, 2018. 2
- [17] MyeongAh Cho, Taeh Kim, Ig-Jae Kim, Kyungjae Lee, and Sangyoun Lee. Relational Deep Feature Learning for Heterogeneous Face Recognition. *IEEE TIFS*, 2020. 8
- [18] 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, 2, 6, 7
- [19] Pinaki Nath Chowdhury, Ayan Kumar Bhunia, Aneeshan Sain, Subhadeep Koley, Tao Xiang, and Yi-Zhe Song. SceneTrilogy: On Human Scene-Sketch and its Complementarity with Photo and Text. In *CVPR*, 2023. 1
- [20] Pinaki Nath Chowdhury, Ayan Kumar Bhunia, Aneeshan Sain, Subhadeep Koley, Tao Xiang, and Yi-Zhe Song. What Can Human Sketches Do for Object Detection? In *CVPR*, 2023. 1
- [21] John Collomosse, Tu Bui, and Hailin Jin. LiveSketch: Query perturbations for guided sketch-based visual search. In *CVPR*, 2019. 2
- [22] Ayan Das, Yongxin Yang, Timothy Hospedales, Tao Xiang, and Yi-Zhe Song. BézierSketch: A generative model for scalable vector sketches. In *ECCV*, 2020. 2
- [23] Ayan Das, Yongxin Yang, Timothy Hospedales, Tao Xiang, and Yi-Zhe Song. SketchODE: Learning Neural Sketch Representation in Continuous Time. In *ICLR*, 2021. 2
- [24] Ayan Das, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Cloud2Curve: Generation and Vectorization of Parametric Sketches. In *CVPR*, 2021. 2
- [25] Zhongying Deng, Xiaojiang Peng, and Yu Qiao. Residual Compensation Networks for Heterogeneous Face Recognition. In *AAAI*, 2019. 8
- [26] Sounak Dey, Pau Riba, Anjan Dutta, Josep Lladós, and Yi-Zhe Song. Doodle to Search: Practical Zero-Shot Sketch-based Image Retrieval. In *CVPR*, 2019. 2
- [27] Martin Engilberge, Louis Chevallier, Patrick Pérez, and Matthieu Cord. SoDeep: a Sorting Deep net to learn ranking loss surrogates. In *CVPR*, 2019. 8
- [28] Gereon Fox, Ayush Tewari, Mohamed Elgharib, and Christian Theobalt. StyleVideoGAN: A Temporal Generative Model using a Pretrained StyleGAN. In *BMVC*, 2021. 2
- [29] Longteng Guo, Jing Liu, Yuhang Wang, Zhonghua Luo, Wei Wen, and Hanqing Lu. Sketch-based Image Retrieval using Generative Adversarial Networks. In *ACM MM*, 2017. 2
- [30] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality Reduction by Learning an Invariant Mapping. In *CVPR*, 2006. 3
- [31] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *CVPR*, 2019. 8
- [32] Quan Huynh-Thu, Marie-Neige Garcia, Filippo Speranza, Philip Corriveau, and Alexander Raake. Study of Rating Scales for Subjective Quality Assessment of High-Definition Video. *IEEE TBC*, 2010. 6
- [33] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *CVPR*, 2017. 6
- [34] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2016. 5

- [35] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up GANs for Text-to-Image Synthesis. In *CVPR*, 2023. 8
- [36] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *CVPR*, 2019. 2, 3, 4, 5, 8
- [37] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. In *CVPR*, 2020. 4, 5, 6, 7, 8
- [38] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015. 6
- [39] Subhadeep Koley, Ayan Kumar Bhunia, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Picture that Sketch: Photorealistic Image Generation from Abstract Sketches. In *CVPR*, 2023. 1
- [40] Subhadeep Koley, Ayan Kumar Bhunia, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. You'll Never Walk Alone: A Sketch and Text Duet for Fine-Grained Image Retrieval. In *CVPR*, 2024. 1
- [41] Subhadeep Koley, Ayan Kumar Bhunia, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Text-to-Image Diffusion Models are Great Sketch-Photo Matchmakers. In *CVPR*, 2024. 1
- [42] Subhadeep Koley, Ayan Kumar Bhunia, Deeptanshu Sekhri, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. It's All About Your Sketch: Democratizing Sketch Control in Diffusion Models. In *CVPR*, 2024. 1
- [43] Fang Liu, Xiaoming Deng, Yu-Kun Lai, Yong-Jin Liu, Cui Xia Ma, and Hongan Wang. SketchGAN: Joint Sketch Completion and Recognition with Generative Adversarial Network. In *CVPR*, 2019. 2
- [44] Feng-Lin Liu, Shu-Yu Chen, Yu-Kun Lai, Chunpeng Li, Yue-Ren Jiang, Hongbo Fu, and Lin Gao. DeepFaceVideoEditing: Sketch-based Deep Editing of Face Videos. *ACM TOG*, 2022. 2
- [45] 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
- [46] Runtao Liu, Qian Yu, and Stella X Yu. Unsupervised Sketch-to-Photo Synthesis. In *ECCV*, 2020. 6
- [47] Umar Riaz Muhammad, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning Deep Sketch Abstraction. In *CVPR*, 2018. 1, 2, 7
- [48] Umar Riaz Muhammad, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Goal-Driven Sequential Data Abstraction. In *ICCV*, 2019. 1, 2, 7
- [49] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep Metric Learning via Lifted Structured Feature Embedding. In *CVPR*, 2016. 3
- [50] 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. 2
- [51] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep Face Recognition. In *BMVC*, 2015. 8
- [52] Yash Patel, Giorgos Tolias, and Jiří Matas. Recall@k Surrogate Loss With Large Batches and Similarity Mixup. In *CVPR*, 2022. 3, 5
- [53] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *ICLR*, 2016. 3
- [54] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. In *CVPR*, 2021. 2, 6, 8
- [55] 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, 4, 6, 7
- [56] Aneeshan Sain, Ayan Kumar Bhunia, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. StyleMeUp: Towards Style-Agnostic Sketch-Based Image Retrieval. In *CVPR*, 2021. 1, 2, 3, 4, 6, 7
- [57] 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, 2, 3
- [58] Aneeshan Sain, Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Subhadeep Koley, Tao Xiang, and Yi-Zhe Song. CLIP for All Things Zero-Shot Sketch-Based Image Retrieval, Fine-Grained or Not. In *CVPR*, 2023. 1
- [59] Aneeshan Sain, Ayan Kumar Bhunia, Subhadeep Koley, Pinaki Nath Chowdhury, Soumitri Chattopadhyay, Tao Xiang, and Yi-Zhe Song. Exploiting Unlabelled Photos for Stronger Fine-Grained SBIR. In *CVPR*, 2023. 2, 6, 7
- [60] Axel Sauer, Katja Schwarz, and Andreas Geiger. StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets. In *SIGGRAPH*, 2022. 8
- [61] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *CVPR*, 2015. 3
- [62] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015. 3, 4, 6
- [63] Jifei Song, Yi-Zhe Song, Tony Xiang, and Timothy M Hospedales. Fine-Grained Image Retrieval: the Text/Sketch Input Dilemma. In *BMVC*, 2017. 2, 3
- [64] Jifei Song, Qian Yu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep Spatial-Semantic Attention for Fine-Grained Sketch-Based Image Retrieval. In *ICCV*, 2017. 2, 3, 6, 7
- [65] Jifei Song, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning to Sketch with Shortcut Cycle Consistency. In *CVPR*, 2018. 6
- [66] Himanshu Thakur and Soumitri Chattopadhyay. Active Learning for Fine-Grained Sketch-Based Image Retrieval. In *BMVC*, 2023. 2
- [67] Yael Vinker, Ehsan Pajouheshgar, Jessica Y Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. CLIPasso: Semantically-Aware Object Sketching. *ACM TOG*, 2022. 2
- [68] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch Your Own GAN. In *ICCV*, 2021. 2

- [69] Xiu-Shen Wei, Yi-Zhe Song, Oisín Mac Aodha, Jianxin Wu, Yuxin Peng, Jinhui Tang, Jian Yang, and Serge Belongie. Fine-grained image analysis with deep learning: A survey. *IEEE TPAMI*, 2021. 1
- [70] Kilian Q Weinberger and Lawrence K Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *JMLR*, 2009. 3
- [71] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A Discriminative Feature Learning Approach for Deep Face Recognition. In *ECCV*, 2016. 3, 8
- [72] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling Matters in Deep Embedding Learning. In *ICCV*, 2017. 3
- [73] Xiang Wu, Lingxiao Song, Ran He, and Tieniu Tan. Coupled Deep Learning for Heterogeneous Face Recognition. In *AAAI*, 2018. 8
- [74] Yue Wu, Yicong Zhou, George Saveriades, Sos Aгаian, Joseph P Noonan, and Premkumar Natarajan. Local Shannon entropy measure with statistical tests for image randomness. *Information Sciences*, 2013. 4
- [75] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv preprint arXiv:1505.00853*, 2015. 4
- [76] Rui Xu, Zongyan Han, Le Hui, Jianjun Qian, and Jin Xie. Domain Disentangled Generative Adversarial Network for Zero-Shot Sketch-Based 3D Shape Retrieval. In *AAAI*, 2022. 2
- [77] Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. Generative Hierarchical Features from Synthesizing Images. In *CVPR*, 2021. 2
- [78] Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic Hierarchy Emerges in Deep Generative Representations for Scene Synthesis. *IJCV*, 2021. 2, 3, 4
- [79] Lan Yang, Kaiyue Pang, Honggang Zhang, and Yi-Zhe Song. SketchAA: Abstract Representation for Abstract Sketches. In *ICCV*, 2021. 1, 2
- [80] Shuai Yang, Zhangyang Wang, Jiaying Liu, and Zongming Guo. Deep Plastic Surgery: Robust and Controllable Image Editing with Human-Drawn Sketches. In *ECCV*, 2020. 2
- [81] Aron Yu and Kristen Grauman. Fine-Grained Visual Comparisons with Local Learning. In *CVPR*, 2014. 6
- [82] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative Image Inpainting with Contextual Attention. In *CVPR*, 2018. 2
- [83] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch Me That Shoe. In *CVPR*, 2016. 2, 3, 4, 5, 6, 7, 8
- [84] 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. 6
- [85] Yu Zeng, Zhe Lin, and Vishal M Patel. SketchEdit: Mask-Free Local Image Manipulation with Partial Sketches. In *CVPR*, 2022. 2
- [86] Wei Zhang, Xiaogang Wang, and Xiaoou Tang. Coupled information-theoretic encoding for face photo-sketch recognition. In *CVPR*, 2011. 8
- [87] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative Visual Manipulation on the Natural Image Manifold. In *ECCV*, 2016. 2
- [88] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *ICCV*, 2017. 6