What Sketch Explainability *Really* Means for Downstream Tasks ?

Hmrishav Bandyopadhyay[1] Pinaki Nath Chowdhury[1] Ayan Kumar Bhunia[1]
Aneeshan Sain[1] Tao Xiang[1,2] Yi-Zhe Song[1,2]
[1]SketchX, CVSSP, University of Surrey, United Kingdom.
[2]iFlyTek-Surrey Joint Research Centre on Artificial Intelligence.
{h.bandyopadhyay, p.chowdhury, a.bhunia, a.sain, t.xiang, y.song}@surrey.ac.uk

Abstract

In this paper, we explore the unique modality of sketch for explainability, emphasising the profound impact of human strokes compared to conventional pixel-oriented studies. Beyond explanations of network behavior, we discern the genuine implications of explainability across diverse downstream sketch-related tasks. We propose a lightweight and portable explainability solution – a seamless plugin that integrates effortlessly with any pre-trained model, eliminating the need for re-training. Demonstrating its adaptability, we present four applications: highly studied retrieval and generation, and completely novel assisted drawing and sketch adversarial attacks. The centrepiece to our solution is a stroke-level attribution map that takes different forms when linked with downstream tasks. By addressing the inherent non-differentiability of rasterisation, we enable explanations at both coarse stroke level (SLA) and partial stroke level (P-SLA), each with its advantages for specific downstream tasks.

1. Introduction

Sketches, rooted in human expression [43], offer a distinctive modality for exploring explainability [61, 70]. In contrast to photos, where each pixel is independent and lacks inherent meaning, sketches are organised into strokes, with each stroke carrying subjective meaning assigned by the sketcher [44]. This paper explores sketch explainability, but with a unique perspective – aiming to provide explanations and unravel the true implications of explainability on various downstream sketch-related tasks.

With this perspective in mind, our approach champions an explainability solution that is (i) lightweight and portable – a plugin seamlessly integrating with multiple pre-trained models without necessitating re-training [98], and (ii) easily adaptable to a diverse array of downstream sketch-specific tasks, benefiting the broader community.

Our solution is exclusively centred on human strokes, aiming to attribute explanation on different stroke granularity: individual strokes (coarse) and their parts (fine). The

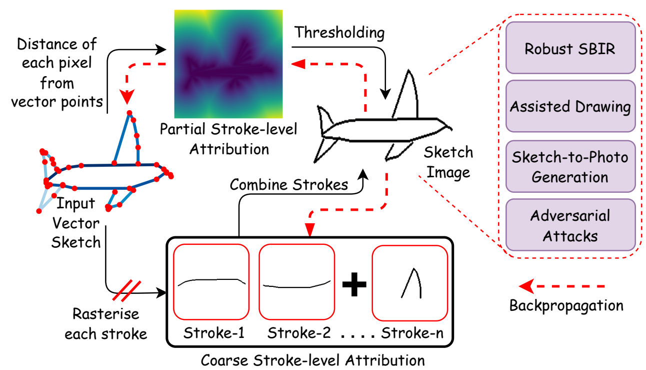

Figure 1. We attribute explanations for individual strokes (stroke-level attribution) and their vector coordinate points (point-level attribution). Stroke-level attribution rasterises individual strokes (non-differentiably) to produce n-stroke images. Next, we sum the stroke images to get the complete sketch image used for downstream tasks. Point-level Attribution computes distance transform from stroke coordinates and thresholds to get the sketch image. Our explainability solution works without re-training for existing tasks like SBIR and sketch-to-photo generation and novel tasks like filtering noisy strokes for assisted drawing and adversarial attack by removing a small stroke.

output of our model markedly differs from typical saliency maps [1] found in photo-based explainability models, where the emphasis is mostly on visualisation [90]. Ours is a task-driven attribution map that assigns *stroke-level* attributes capturing how altering stroke characteristics can impact model prediction. Depending on the downstream tasks, attributions can be grounded to, for example (i) importance of entire strokes, which is more suitable to filter noisy strokes [15] in assisted drawing, and remove small strokes for adversarial attacks on existing sketch encoders, and (ii) stroke shape and length, where a partial-stroke level attribution is beneficial for tasks like sketch-based image retrieval [77] and sketch-to-photo generation [55].

To showcase the adaptability of our model, we carefully devise four applications: two well-studied tasks from existing literature (retrieval [12, 26] and generation [55, 102]), and two entirely novel tasks (assisted drawing and sketch adversarial attack). In *retrieval*, we evaluate reliability of model predictions by comparing predicted stroke order with the order in which a human draws them. For *generation*, we pinpoint strokes with the least influence, offering explicit feedback to end-users regarding which strokes the model prioritised and which it overlooked. In *assisted drawing* [4], we assist novice artists in faithfully sketching a particular photo by identifying strokes that do not match the target photo. Lastly, in *adversarial attacks*, we unveil the vulnerability of state-of-the-art sketch encoders by removing a small imperceptible stroke in any sketch, resulting in significant changes to the model's prediction.

The focal point connecting all downstream tasks is our proposed stroke-level attribution. The key question, therefore, is how to backpropagate information to strokes while addressing the inherent non-differentiability of rasterisation – strokes are most often represented as discrete coordinates and rasterised before feeding into downstream applications. We provide two solutions for non-differentiability: (i) coarse stroke level: we first rasterise individual strokes to produce raster stroke images. Then, we combine these stroke images to get the complete sketch image (see Fig. 1) – since this addition of stroke images is a differentiable operation, we can backpropagate information from the complete sketch to individual raster stroke images. (ii) fine partial-stroke level: we create a distance transform image from stroke coordinates ("red dots" in Fig. 1 vector sketch) by calculating the minimum distance of each pixel in the image from the coordinates. Then, we threshold the distance value to get the sketch image (white pixels for a high distance and black pixels for a low distance). Since the distance function and our threshold step are both differentiable, we can backpropagate information from the sketch images to stroke coordinates.

Our contributions can be summarised as follows: (i) We explore sketch explainability, emphasising the importance of strokes in human-drawn sketches. (ii) We highlight the profound impact of explainability on various sketch-related domains, presenting applications in retrieval, generation, assisted Drawing, and adversarial attacks. (iii) We solve for the non-differentiability problem of rasterisation, and provide both stroke-level and partial-stroke level attribution.

2. Related Works

Sketch for Visual Understanding: Having a high visual proximity [43] to real images and carrying human subjectivity [9, 76], amateur sketches or abstract line drawings [59] has been a popular modality for customised expression, thus driving extensive applications as a query for retrieval [26, 32, 76, 85] of object [29] or scene [24] images, 3D shapes [104], and even concepts like in 'pictionary-like' games [11]. As a canvas for creativity, sketch helps image-editing [62, 110, 114], or generation of objects [20, 21, 38, 99], scenes [109], and 3D shapes [10, 39, 105, 119]. Being easily editable, sketch enables interactive access to AI systems like image-segmentation [45, 122], object localisation [96], image-inpainting [110, 114], and incremental learning [14]. Being application-specific however, such works largely ignored *explaining* the '*how*' of sketch-correspondence. The few who did, customised training pipelines [61, 70] for niche tasks. In this work, we thus make the first attempt at visualising salient sketch-regions (strokes), as an explainability-tool (like GradCAM [81] in photos) for existing pre-trained sketch-based downstream networks.

Explaining CNN Predictions: CNN explanations visually highlight regions of input having the maximum influence on a model's predictions [87]. This visualisation of '*salient*' regions is either through an analysis [37] of a regular pre-trained network after it has completed training (*post-hoc*), or by designing and training explicitly interpretable (*i.e. explain-and-predict*) models [17, 19, 64]. Given a pre-trained CNN, a *post-hoc* algorithm either visualises (i) *model attributes* like feature and activation-maps [34, 65, 66, 82, 113, 120] that *imply* saliency of specific input (pixel) regions, or (ii) *input attributes* directly as pixels [91] or pixel-regions [51] coloured according to their relative importance. Visualisation of input attributes is facilitated through perturbation based algorithms [18, 36, 37] and gradient-based analysis [87, 89, 91]. Perturbation-based algorithms [27, 68, 113] detect saliency of pixel-regions by measuring impact of their absence on the prediction score. Whereas, gradient-based algorithms [8, 30, 86, 92, 117] measure the gradient of the prediction with respect to individual input-pixels (input-gradients [83]), attributing them based on this value. Unlike images however, sketch is a sparse-information modality [23] for pixels. As such, we explore explainability in sketches by computing stroke and point attribution for fine-grained sketch explanations.

Evaluation of CNN Explanations: The evaluation of CNN explanations has evolved over time – from naive qualitative analysis of visualisations [82, 87, 91, 113] to standardised theoretical [5, 92] and empirical [72, 107, 113] baselines. Analysing explanations theoretically helps evaluate them in a model-agnostic environment, where their mathematical form is checked against pre-defined properties *(axioms)*[5, 54, 92]. Empirical evaluations, on the other hand, involve experiments measuring *(i)* variance in explanations upon perturbations of inputs [107] and model weights [1, 2] (sanity checks), and *(ii)* accuracy of explanations in locating important features [37, 72, 82, 117]. These features, when perturbed, influence the CNN maximally, as measured by perturbation-based metrics [6, 51, 68, 78, 82, 83, 113]. Recently, however, these evaluation protocols have met criticism [35] due to their detachment from humans. Instead, human studies [53, 58, 82, 88], and human-centered metrics [35] have been proposed for evaluating explanation interpretability. In this work, we attribute strokes by back-propagation [86, 87, 91, 92], evaluating attributions through sanity checks [1], empirical metrics [51] and human studies [53] on downstream sketch-based applications [112].

3. Background

Here, we provide a brief overview of some standard concepts, ubiquitous in explainability literature [1] to help formalise the question: *"what entails a good explanation?"*

Attribution Algorithms: It highlights relevant regions (e.g., pixels in an image, see Fig. 1) that are responsible for the model's prediction. Despite its importance for safety-critical applications [49, 75, 101], making an attribution algorithm interpretable to humans remains an open problem. Surprisingly, a more faithful attribution is usually less interpretable and vice-versa [91, 113]. Prior works [16, 81] study this trade-off between faithfulness vs. interpretability as an answer to: *"What makes a good visual explanation?"*.

The attribution map $\mathbb{A} \in \mathbb{R}^{H \times W}$ is typically calculated using gradients for an input $\mathrm{X} \in \mathbb{R}^{H \times W \times 3}$ for a classification model $\hat{y} = F_\theta(\mathrm{X}) \in \mathbb{R}^C$, pre-trained on C categories. The gradients are a simple and good indicator of how much the model prediction changes for input X as,

$$\mathbb{A} = \partial F_\theta(\mathrm{X}) / \partial \mathrm{X} \tag{1}$$

Interpretability: It is the ability of an attribution algorithm to provide a qualitative "understanding" for a model [73]. This "understanding" depends on the target audience, e.g., a human expert may interpret a small Bayesian network [56], but a layman is more comfortable with a weighted attention (or feature) map that highlights salient regions [1]. To evaluate the interpretability of attribution maps, prior works either (i) perform downstream tasks [31, 81] that depend on the interpretation (e.g., object localisation – predict the bounding box and semantic segmentation for image region with the highest attribution) or (ii) human studies [53], typically conducted in two setups – class discriminative (given an attribution map, ask users to identify the category predicted by a model), and trustworthiness (compare attribution maps from a strong and a weak model, and ask users to identify the stronger model).

Faithfulness: It is the ability of attribution algorithms to accurately "explain" the computation learned by a model. For example, in theory, a fully faithful attribution is the entire model (e.g., ResNet-18 [41]) but is not interpretable by a human. In practice, for an attribution to be meaningful, it is often impossible to be completely faithful. To balance this trade-off, prior works explore human-interpretable attributions that are locally faithful for a given model prediction. One approach is image occlusion [73], where the difference in the model scores is measured when masking different patches in an input image. Image patches that significantly change the model score are deemed important by the attribution algorithm.

4. Proposed Method

The attribution map $\mathbb{A}$ in Eq. (1) gives a faithful fine-grained explanation for each pixel in X. However, such pixel attribution does not provide meaningful explanations when interpreting sparse human-drawn sketches (many empty white pixels). Additionally, pixel attribution does not consider the *sketch construction process* – humans sketch a sequence of strokes, not pixels. Hence, it is most appropriate that sketch predictions should be attributed to strokes and not pixels. However, the key challenge to stroke attributions is that most sketch applications [77, 118] use raster sketches – a non-differentiable process to convert a sequence of strokes into pixels. In the following sections, we propose two stroke attribution algorithms that consider the sketch construction process by designing a differentiable rasterisation pipeline for a vector sequence of strokes.

Algorithm 1: Non-differentiable Rasterisation

```
Data: V ← Vector Sketch of size ℝ^{T×5}
Result: X ← Blank (Zero) Canvas of size ℝ^{h×w×3} ;
𝔹(·,·) ← Bresenham Function ;
v_0 = (x_0, y_0, q_0^1, q_0^2, q_0^3) ← V[0] ;
v_prev ← v_0 ;
q_prev ← q_0^1 ;
for v_t = (x_t, y_t, q_t^1, q_t^2, q_t^3) ← V[1 ... T] do
    if q_prev = 1 and q_t^1 = 1 then
        for (p_x^i, p_y^i) ← 𝔹(v_prev, v_t) do
            X (p_x^i, p_y^i) ← 255;
        end
    end
    v_prev ← v_t;
    q_prev ← q_t^1;
    if q_t^3 = 1 then
        exit() ;        /* End of Drawing */
    end
end
```

4.1. Sketch Representations

While sketches are used in several formats (or representations) like Raster [111], Vector [40], or Bézier [28], digital sketches are primarily captured in vector form, as a list of points traced on a drawing pad. These points are usually a five-element vector $v_t = (x_t, y_t, q_t^1, q_t^2, q_t^3)$ where, (x_t, y_t) are absolute coordinates in a $(H \times W)$ drawing canvas and the last three elements are one-hot encoding of pen-states: pen touching the paper $(1, 0, 0)$, pen is lifted $(0, 1, 0)$, and end of drawing $(0, 0, 1)$. Hence, a vector sketch with T points is represented as $\mathrm{V} \in \mathbb{R}^{T \times 5}$.

As most downstream sketch applications work on rasterised sketches $\mathrm{X} \in \mathbb{R}^{H \times W \times 3}$, prior works [111] translate (non-differentiably) a vector sketch V to its equivalent raster sketch X using Algorithm 1. For this, the Bre-

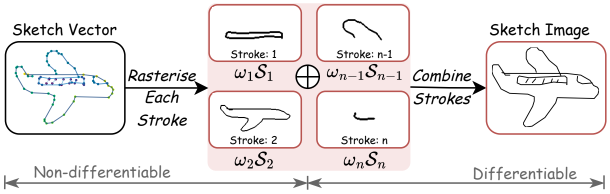

Figure 2. Coarse Stroke-level Attribution. Backpropagate gradients from raster sketch X to raster strokes $\mathcal{S}_i$, with weight ω_i.

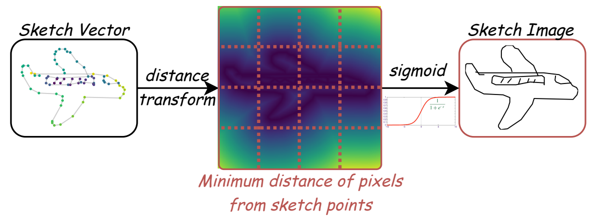

Figure 3. Partial Stroke-level Attribution. Backpropagate gradients from raster sketch X to vector sequence of coordinates V.

senham function $\mathbb{B}(\cdot)$ is used to connect two vector points $\{v_{t-1}, v_t\} \in \mathrm{V}$ in the pixel space $\{(p_x^1, p_y^1), \dots, (p_x^n, p_y^n)\} \in \mathbb{B}(v_{t-1}, v_t)$ via a continuous line. Next, we show how our sketch attributions overcome this non-differentiable rasterisation and backpropagate gradients (Eq. (1)) from pixels $\mathrm{X} \in \mathbb{R}^{H\times W\times 3}$ to strokes and points in $\mathrm{V} \in \mathbb{R}^{T\times 5}$.

4.2. Coarse Stroke-level Attribution (SLA)

In this section, we backpropagate gradients from pixel space in $\mathrm{X} \in \mathbb{R}^{H\times W\times 3}$ to strokes, defined as a continuous set of points $\{v_t, v_{t+1}, \dots v_{t+n}\}$ from the *first* pen-down $(1, 0, 0)$ till the pen-up $(0, 1, 0)$ state. Algorithm 1 converts the vector stroke points into a raster stroke $\mathcal{S}_i \in \mathbb{R}^{H\times W\times 3}$. The final raster sketch is then a differentiable composition[1] of m raster strokes $\mathrm{X} = \sum_{k=1}^{m} \mathcal{S}_k$. We compute SLA as,

$$\mathbb{A}_i^R = \frac{\partial F_\theta(\mathrm{X})}{\partial \mathcal{S}_i} = \frac{\partial F_\theta(\mathrm{X})}{\partial \mathrm{X}} \cdot \frac{\partial \sum_{k=1}^{m} \mathcal{S}_k}{\partial \mathcal{S}_i} \tag{2}$$

This, however, gives a degenerate solution where all strokes will have the same attribution $\partial \sum_{k=1}^{m} \mathcal{S}_k / \partial(\mathcal{S}_i) = 1$. To avoid this, we compute a weight factor $\omega_i \in \mathbb{R}^{H\times W\times 3}$ for each stroke $\mathcal{S}_i$ such that,

$$\omega_i(p_x, p_y) = \begin{cases} 1 & \text{if } (p_x, p_y) \in \mathbb{B}(v_{t-1}, v_t) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In other words, given consecutive vector sequence of points (v_{t-1}, v_t) in stroke $\mathcal{S}_i$, we find all points (p_x, p_y) using Bresenham function $\mathbb{B}(\cdot)$ that lie "on the stroke" and assign $\omega_i(p_x, p_y) = 1$. Hence, longer strokes will have more 1's compared to shorter strokes. Finally, Eq. (2) is adapted as

$$\mathbb{A}_i^R = \frac{\partial F_\theta(\mathrm{X})}{\partial \mathrm{X}} \cdot \frac{\partial \sum_{k=1}^{m} \omega_k \mathcal{S}_k}{\partial \mathcal{S}_i} \tag{4}$$

SLA makes the non-differentiable rasterisation (V$\rightarrow$X) *partially differentiable* (stokes $\mathcal{S}$ to sketch X). In other words, SLA answers *"Which strokes in a sketch are important"*. Next, we make the rasterisation fully differentiable and backpropagate gradients to vector points V to answer *"Which point in a sketch is important"*.

[1]For overlapping strokes in $\mathcal{S}_i$ and $\mathcal{S}_j$, we clamp the maximum pixel value using differentiable functions like `torch.clamp`

4.3. Partial Stroke-level Attribution (P-SLA)

Unlike SLA, which partially captures the sketch construction process (stroke-level), partial stroke-level attribution (P-SLA) can fully backpropagate gradients to the vector list of coordinates $\mathrm{V} \in \mathbb{R}^{T\times 5}$ traced on a drawing pad. Given a blank canvas $\mathrm{X} \in \mathbb{R}^{H\times W\times 3}$, we (i) calculate the minimum distance of every pixel (p_x, p_y) in X from a line segment (v_{t-1}, v_t) in V, and (ii) compute the pixel intensity of $\mathrm{X}(p_x, p_y)$ as function of the minimum distance as

$$\begin{aligned} \mathrm{X}(p_x, p_y) =& \sigma\big[2 - 5 \min_{t=2,\dots,T} \big(\\ & \mathtt{dist}((p_x, p_y), v_{t-1}, v_t) + (1 - q_{t-1}^1)10^6\big)\big] \end{aligned} \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function, and $\mathtt{dist}(\cdot)$ is a distance function (see Supp.) from a point (p_x, p_y) to a line segment (v_{t-1}, v_t). For pen-up states ($q_{t-1}^1 = 0$), we blow up ($\times 10^6$) the distance values that make the pixel intensities $\mathrm{X}(p_x, p_y) \rightarrow 0$, i.e., not render strokes for (v_{t-1}, v_t). Finally, we compute P-SLA

$$\mathbb{A}_t^V = \frac{\partial F_\theta(\mathrm{X})}{\partial v_t} = \frac{\partial F_\theta(\mathrm{X})}{\partial \mathrm{X}} \cdot \sum_{\forall p_x, p_y} \left\{ \frac{\partial \mathrm{X}(p_x, p_y)}{\partial v_t} \right\} \tag{6}$$

5. Applications of Stroke Attributions

Despite its simplicity, designing stroke attribution algorithms that capture the sketch construction process unlocks insights into numerous existing downstream tasks like classification [40], robust sketch-based image retrieval (category-level [33] and fine-grained [79]) and enables some novel sketch applications like assisted drawing [15], interactive sketch to photo generation [55], adversarial attacks on human-drawn sketches, and discovering the "arrow of time" [100] in raster sketches.

5.1. Robust Sketch Based Image Retrieval

Given a query sketch $\mathrm{X} \in \mathbb{R}^{H\times W\times 3}$, category-level sketch-based image retrieval (SBIR) aims to fetch category-specific photos from a gallery of multi-category photos (*e.g.*, given sketch of a 'shoe' retrieve *any* photo 'shoe' from a gallery of 'shoes+hats+cows'). Conversely, fine-grained SBIR aims to retrieve *one* instance from a gallery of the *same* category photos (e.g., given the sketch of a 'shoe' retrieve *one* photo shoe from a gallery of *all* shoes). Deep learning frameworks

Table 1. Stroke attribution (SLA, P-SLA) make SBIR systems reliable. Sketches with a high correlation ($Corr$) of stroke saliency (predicted by SLA or P-SLA) with human-drawn temporal stroke order tend to have higher retrieval accuracy.

	Metrics	Full Dataset	$\text{Corr} \geq 0.5$ SLA	$\text{Corr} \geq 0.5$ P-SLA	$\text{Corr} \leq 0.1$ SLA	$\text{Corr} \leq 0.1$ P-SLA
Category Level	mAP	53.1	55.3	57.6	51.7	50.1
	P@200	65.9	66.7	68.5	64.6	61.5
Fine Grained	Acc.@1	15.3	16.4	17.6	13.8	12.7
	Acc.@5	34.2	36.9	39.4	31.1	28.3

learn a joint sketch-photo manifold (for category and fine-grained) via a feature extractor [26, 29, 103] trained using triplet loss [112]. Recent adoption of foundation models for SBIR [77] shifts focus to robust deployment using the open-set generalisation of CLIP [71].

Towards this goal of robust deployment, our sketch attribution algorithms ($\mathbb{A}_i^R$ and $\mathbb{A}_t^V$) can predict which strokes the network focuses on when retrieving a photo (Fig. 4). Apart from interpreting SBIR models, sketch attribution can also help detect potential failures at runtime (inference). First, we use the attribution scores $\mathbb{A}_i^R$ or $\mathbb{A}_t^V$ to rearrange the strokes from highest to lowest. The attribution scores indicate the most salient to the least salient strokes that affect model prediction. Second, we calculate a correlation ($Corr$) of our predictor stroke order with the ground-truth temporal stroke order drawn by a user (humans draw the most salient regions first and least salient areas last [33, 79]). A high correlation indicates that humans and our model prioritise strokes similarly, whereas a low $Corr$ denotes that the model and the user prioritise different strokes. We evaluate SBIR on a pre-trained SOTA model [77] using CLIP with prompt learning as a sketch and photo encoder.

Datasets: We use TU-Berlin [33] (for category-level SBIR) and Sketchy [79] (for fine-grained SBIR). TU-Berlin contains 250 categories, with 80 free-hand sketches in each, and $204,489$ images [116]. Sketchy [79] has $75,471$ sketches over 125 categories having 100 images in each [108].

Evaluation Metrics: Following [77], we use mean average precision (mAP) and precision for top 200 retrieved samples (P@200) for category-level SBIR. For fine-grained SBIR [23], we measure Acc.@q, i.e., the percentage of sketches whose true matched photo is in the top-q list.

Results: We divide the evaluation set into two sets: (i) those that have a high correlation $Corr \geq 0.5$, and (ii) those with a low correlation $Corr \leq 0.1$ of ground-truth and predicted stroke order. Tab. 1 shows sketches with a high $Corr \geq 0.5$ are $1.7/3.9$ more accurate in Acc.@1/Acc.@5 than those with $Corr \leq 0.1$ for fine-grained SBIR, and $3.3/1.7$ better in mAP/P@200 for category-level SBIR. **Full Dataset** indicates the performance of the pre-trained model on the entire evaluation set.

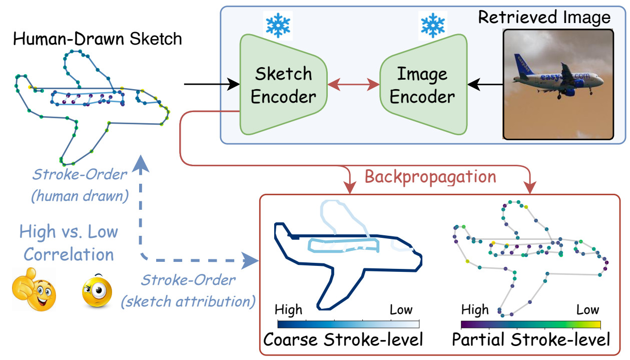

Figure 4. Sketch attributions from stroke-level and point-level for image retrieval. High correlation of human-drawn stroke order with that from sketch-attributions (high→low) indicate our sketch encoder gives more importance to salient strokes drawn *early on*.

5.2. Assisted Drawing via Noisy Stroke Removal

Although sketching has enabled many exciting applications [25, 55, 63, 118], the fear to sketch (i.e., *"I can't sketch"*) has proven fatal for its widespread adoption. To solve this, prior works [15] used complex (and hard to train [12]) reinforcement learning [80] to predict the importance of each stroke in a sketch. Next, a stroke subset selector removes noisy (less important) strokes, leaving only those positively contributing to the downstream tasks.

In this section, we focus on assisted drawing – given a photo, we use attribution scores ($\mathbb{A}_i^R$ or $\mathbb{A}_t^V$) to help humans draw a faithful and clean sketch. Our method is significantly simpler than reinforcement learning alternatives [15].

For SLA, an input sketch $\mathrm{X} = \sum_{k=1}^{m} \omega_k \mathcal{S}_k$ is composed of m strokes. We calculate the cosine similarity ($\mathtt{sim}$) of input sketch X with its target photo $\mathrm{P} \in \mathbb{R}^{H \times W \times 3}$ to measure *"how faithfully* X *describes* P*"*. Next, we backpropagate gradients from the cosine similarity to strokes $\mathcal{S}_i$ and calculate stroke-level attribution score $\mathbb{A}_i^R$ as

$$\mathbb{A}_i^R = \frac{\partial \mathtt{sim}(F_\theta(\mathrm{X}), F_\theta(\mathrm{P}))}{\partial \mathrm{X}} \cdot \frac{\partial \sum_{k=1}^{m} \omega_k \mathcal{S}_k}{\partial \mathcal{S}_i} \tag{7}$$

The pre-trained sketch and photo encoder[2] $F_\theta(\cdot)$ must be highly accurate to judge sketch–photo correspondence. Hence, we use pre-trained CLIP+prompts encoder from [77]. To remove noisy strokes, we only update the weights $\omega_i \in \mathbb{R}^{H \times W \times 3}$ using a normalised attribution score $\mathbb{A}_i^R$ as

$$\omega_i^* = \omega_i \cdot \mathtt{Gumbel\_Softmax}\Big(\frac{\mathbb{A}_i^R}{\sum_{\forall i} \mathbb{A}_i^R} + \Delta\Big) \tag{8}$$

$\mathtt{Gumbel\_Softmax}(\cdot)$ makes the output one-hot (discrete value) in the forward pass but differentiable with a probability distribution that sum to 1 in the backward pass [47]. The

[2] The sketch and photo encoder $F_\theta(\cdot)$ could be a siamese-style shared network or two independent models with different network weights [77].

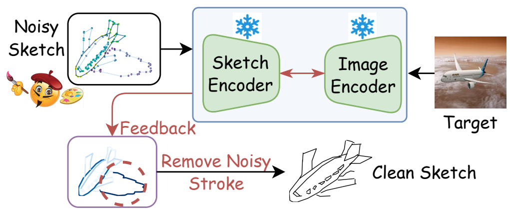

Figure 5. Assisted drawing via sketch healing (or filtering noisy strokes) using stroke attributions from SLA and P-SLA. This helps users having fear-to-sketch (*"I can't sketch"*).

modified sketch is constructed as $\mathrm{X} = \omega_1^* \mathcal{S}_1 + \cdots + \omega_m^* \mathcal{S}_m$. Intuitively, we keep strokes that contribute ($\mathbb{A}_i^R$) to a high cosine similarity matching human sketch X and target photo P and remove $\mathcal{S}_i$ with normalised $\mathbb{A}_i^R$ lower than $(0.5 - \Delta)$.

Similar to SLA, we can also use P-SLA to compute the attribution $\mathbb{A}_t^V$ for each point v_t in the vector sketch $\mathrm{V} \in \mathbb{R}^{H \times W \times 3}$, by measuring the cosine similarity between input X and target photo P as

$$\mathbb{A}_t^V = \frac{\partial \mathtt{sim}(F_\theta(\mathrm{X}), F_\theta(\mathrm{I}))}{\partial \mathrm{X}} \cdot \sum_{\forall p_x, p_y} \left\{ \frac{\partial \mathrm{X}(p_x, p_y)}{\partial v_t} \right\} \quad (9)$$

We remove noisy points v_t by updating the pen-states in Eq. (5) from pen-down $(1, 0, 0)$ to pen-up $(0, 1, 0)$ depending on its attribution $\mathbb{A}_t^V$ for point $v_t \in \mathrm{V}$.

$$q_{t-1}^{1*} = q_{t-1}^1 \cdot \mathtt{Gumbel\_Softmax}\left(\frac{\mathbb{A}_{t-1}^V}{\sum_{\forall t} \mathbb{A}_t^V} \right) \quad (10)$$

Using updated values for q_{t-1}^{1*}, we update $q_{t-1}^{2*} = 1 - q_{t-1}^{1*}$ and recalculate pixel intensities for raster sketch $\mathrm{X}(p_x, p_y)$. The value of hyper-parameter Δ significantly affects the stroke removal process. We found the optimal Δ for SLA and P-SLA is 0.3 and 0.1, respectively. A higher Δ for P-SLA gives broken lines with a drop in the visual quality of an input sketch. Next, we evaluate stroke filtering using SLA and P-SLA on popular human-drawn sketch datasets.

Dataset: For a fair comparison with prior works [15], we evaluate on fine-grained SBIR datasets QMUL-Shoe-V2 and QMUL-Chair-V2 [12, 67, 112]. It consists of $6,730/1,800$ sketches and $2,000/400$ photos from Shoe-V2/Chair-V2. We evaluate on the standard test-split of $679/525$ sketches and $200/100$ photos.

Evaluation Metric: We measure the retrieval accuracy of the clean sketch with the target photo by computing Acc.@1 and Acc.@5. A high accuracy need not correspond to high visual quality to the human eye [84]. Hence, we conducted a small human study with 5 participants and reported the mean opinion score (MOS) [46]; each was asked to compare two sets of 50 sketch pairs (GT sketch vs SLA filtered) and (GT sketch vs P-SLA filtered).

Results: Removing noisy strokes from human-drawn sketches is still a new topic; hence, to the best of our knowledge, there is only one work by Bhunia *et al.* [15]. From

Table 2. Noisy stroke removal using SLA and P-SLA attribution.

	Metrics	GT Sketch	SLA filtered	P-SLA filtered	Bhunia *et al.* [15]
Shoe-V2	Acc.@1	33.4	36.1	36.5	43.7
	Acc.@5	67.8	68.7	69.3	74.9
	MOS	28.6	85.7	57.1	–
Chair-V2	Acc.@1	53.3	54.9	56.5	64.8
	Acc.@5	74.3	76.6	77.1	79.1
	MOS	35.8	71.4	57.1	–

Tab. 2, both Bhunia *et al.* [15], and ours improve fine-grained SBIR performance by 10.3% and 3.1%, respectively. However, [15] outperforms our SLA and P-SLA filtered methods by 7.6% and 7.2%, respectively. This performance gap is likely because [15] trains the baseline model [12] using actor-critic version of PPO [80], whereas our SLA and P-SLA work post-hoc [98] *without training* the baseline model [12]. Additionally, Bhunia *et al.* [15] aims to design a robust SBIR pipeline, whereas our SLA/P-SLA filtering aims to assist humans in drawing sketches. For human study (MOS): (i) users prefer SLA-filtered 78.5% vs. 21.5% for GT sketch, (ii) however, P-SLA-filtered are preferred only 57.1% vs. 42.9% for GT sketch. This is verified by Fig. 5 where P-SLA-filtered sketches have broken strokes that degrade their visual quality.

5.3. Interactive Sketch To Photo Generation

The upsurge of large-scale image generation models (e.g., Stable-Diffusion [74], GigaGAN [50]) helped develop sketch-conditional image generation [63, 118]. However, a key limitation of conditional image generation is that these models do not always faithfully follow the input condition. This was resolved in two stages for text-to-image generation: (i) find word tokens with low influence on generated image, and (ii) iteratively update its activation until it reaches a minimum required value. In this section, we design a pipeline for faithful sketch-to-image generation.

Using sketch attributions from SLA and P-SLA, we design a post-hoc [98] method that gives *feedback to the user* – which strokes the model *focuses* on and which strokes are being *ignored*. Given this feedback, a user can interact with the system to ensure the model attends all salient regions.

Our interactive pipeline is built on top of a pre-trained sketch-to-photo generation model [55], comprising a modified ResNet-50 [41] as sketch encoder and StyleGAN [52] as image decoder. Given a raster sketch X, the modified sketch encoder computes a latent vector $z_{s2p}^+ = F_\theta(\mathrm{X})$, where $z_{s2p}^+ \in \mathbb{R}^{14 \times 512}$. Next, the StyleGAN [52] decoder generates the underlying image from z_{s2p}^+. For a faithful sketch-to-image generation, we measure the "influence" of each stroke/vector point on the latent code z_{s2p}^+. Particularly, prior works [106] suggest that z_{s2p}^+ is disentangled into 14-level semantic feature hierarchy, where $z_{s2p}^{+1} \in \mathbb{R}^{512}$ has coarse-level features controlling major semantic struc-

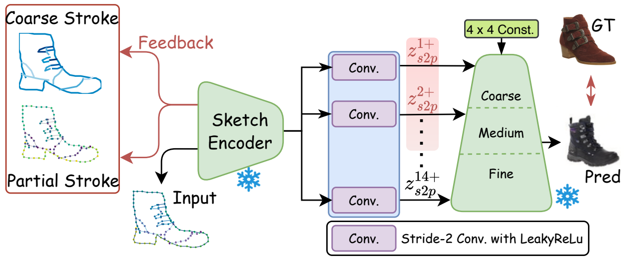

Figure 6. Interactive Sketch to Photo Generation: Our stroke attribution algorithms make existing sketch-to-photo generation pipelines [55] more *faithful*. We achieve this by computing the stroke-level $\mathbb{A}_i^S$ or coordinate-level $\mathbb{A}_t^V$ attribution that has a *maximal* influence on the latent code z_{s2p}^{+} used by the image decoder.

tures and $z_{s2p}^{+14} \in \mathbb{R}^{512}$ has fine-level features controlling colour schemes, etc. Since sketches primarily convey semantic structure, we use the sum of the first 7-layers to compute stroke attribution $\mathbb{A}_i^R$ or $\mathbb{A}_t^V$ as

$$\begin{aligned} \mathbb{A}_i^R &= \frac{\partial \sum_{k=1}^{7} z_{s2p}^{+k}}{\partial \mathrm{X}} \cdot \frac{\partial \sum_{k=1}^{m} \omega_k \mathcal{S}_k}{\partial \mathcal{S}_i} \\ \mathbb{A}_t^V &= \frac{\partial \sum_{k=1}^{7} z_{s2p}^{+k}}{\partial \mathrm{X}} \cdot \sum_{\forall p_x, p_y} \left\{ \frac{\partial \mathrm{X}(p_x, p_y)}{\partial v_t} \right\} \end{aligned} \tag{11}$$

Next, we qualitatively evaluate our iterative sketch to photo generation pipeline, as shown in Fig. 6.

5.4. Adversarial Attacks on Human Sketches

Szegedy *et al*. [94] discovered that predictions by deep networks can be manipulated with extremely low-magnitude input perturbations. For images, these can be restricted to be imperceptible to human vision, but their effect can completely change the output prediction by a deep network. Such adversarial attacks are possible in image classification [94], semantic segmentation [7, 42], object detection [97, 115], object tracking [22, 48], etc. Studying these quirks is crucial as it can pose a real threat to deep learning as a pragmatic technology [3]. While major work has been dedicated to attacks on images, the recent surge of deployable sketch applications [77, 118] motivates us to *present the first study* on adversarial attacks for human-sketches.

We show how our stroke attribution algorithms (SLA and P-SLA) provide the necessary information for adversarial attacks. For brevity, we focus on adversarial attacks on sketch classification [40, 111]. Intuitively, we use sketch attribution to remove the smallest stroke (in SLA Attack) and minimum number of points (in P-SLA Attack), yet have the maximum impact on changing prediction of a pre-trained classifier $F_\theta(\mathrm{X}) = y_{\text{cls}}$. Ours is *(i)* a *white-box* [69, 95] setting – we have access to network weights and gradients of our ResNet-18 [41] classifier, pre-trained on QuickDraw [40] or TU-Berlin [33]; *(ii) untargeted* attack – while targeted attacks [60, 121] misclassify $F_\theta(\cdot)$ from $y_{\text{cls}}^{\text{GT}}$ to *a specific target class* y_{cls}^{*}, untargeted attacks [93] aim to misclassify to *any arbitrary class* $y_{\text{cls}}^{\text{GT}} \neq y_{\text{cls}}$. For a neater description of SLA and P-SLA attacks, we define the rasterisation process using $\mathcal{R}(\cdot)$ as $\mathrm{X} = \omega_1\mathcal{S}_1 + \cdots + \omega_m\mathcal{S}_m = \mathcal{R}(\mathcal{S})$ for SLA and following Eq. (5) for P-SLA we define $\mathrm{X} = \mathcal{R}(\{v_1, \dots v_T\}) = \mathcal{R}(\mathrm{V})$. Next, we find a stroke $\mathcal{S}_{adv}$ with stroke length $|\mathcal{S}_i|$ less than some threshold ϵ as

$$\mathcal{S}_{adv} = \arg \max_{|\mathcal{S}_j| \leq \epsilon} \mathcal{L}_{\text{cls}}\big(F_\theta(\mathcal{R}(\mathcal{S} - \{\mathcal{S}_j\})),\ y_{\text{cls}}^{\text{GT}}\big) \tag{12}$$

Table 3. Sketch Adversarial Attacks: Using stroke attributions, we remove a small stroke ($|\mathcal{S}_i| \leq \epsilon$) that misclassifies an input sketch.

	No Attack	SLA Attack $\epsilon=5$	SLA Attack $\epsilon=15$	P-SLA Attack $\epsilon=5$	P-SLA Attack $\epsilon=15$
QuickDraw	67.2	65.7	64.5	65.1	63.7
TU-Berlin	74.9	71.5	68.5	70.2	68.1

Unlike typical adversarial attacks on images that *add* a small noise $(\mathrm{X} + \Delta x)$ with $||\Delta x||_\infty \leq \epsilon$, for sketch adversarial attacks, we *remove* a small stroke $(\mathrm{X} - \mathcal{S}_{adv})$ such that the stroke length is less than ϵ as $|\mathcal{S}_{adv}| \leq \epsilon$. For P-SLA attack, we find a subset of ϵ vector points $\mathrm{V}_{adv} = \{v_1^{adv}, \dots v_\epsilon^{adv}\}$ from input sketch $\mathrm{V} \in \mathbb{R}^{T \times 5}$ which maximises the categorical cross-entropy loss $\mathcal{L}_{\text{cls}}$ as

$$\begin{aligned} v_t^{adv} &= \mathcal{L}_{\text{cls}}\big(F_\theta(\mathcal{R}(\mathrm{V} - \{v_t\})),\ y_{\text{cls}}^{\text{GT}}\big) \\ \mathrm{V}_{adv} &= \texttt{top@k}(\{v_1^{adv}, v_2^{adv}, \dots v_T^{adv}\}, \epsilon) \end{aligned} \tag{13}$$

where, $\texttt{top@k}(\cdot, \epsilon)$ picks the highest ϵ elements. Fig. 7 shows the adversarial strokes $\mathcal{S}_{adv}$ and points V_{adv} in red.

Dataset: We evaluate sketch adversarial attacks on QuickDraw [40] and TU-Berlin [33]. We use a subset [103] of $50M$ sketches in QuickDraw [40] as $3.8M$ samples across 345 categories split in $2.1M$ sketches for training, $0.3M$ for validation and $0.4M$ for evaluation. See Sec. 5.1 for details on TU-Berlin [33] dataset.

Evaluation: We measure the drop in classification accuracy when using SLA and P-SLA attacks in Tab. 3. Unlike image-based adversarial attacks where ϵ is a pixel intensity (non-integer, decimal value), our sketch attacks occur on stroke/point length, making ϵ an integer. A higher $\epsilon \geq 20$ removes "visible" strokes in SLA and broken lines in P-SLA (Fig. 7). Hence, we evaluate accuracy drop for $\epsilon = 5$ and $\epsilon = 15$ in Tab. 3. We observe for $\epsilon = 5$ P-SLA offers a better adversarial attack than SLA by a margin of 1.3%.

6. Human Study

Interpretability aims to help humans understand a model's reasoning process (*transparency*), verify that its predictions are based on the right constraints (*fairness*), and evaluate its confidence (*trustworthiness*) [73, 81]. In Sec. 5, we evaluated interpretability using several automatic metrics (e.g.,

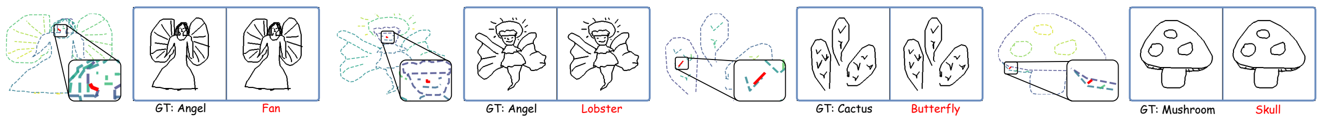

Figure 7. Adversarial attacks on human drawn sketches using SLA and P-SLA. The adversarial strokes are marked in RED.

classification or retrieval accuracy) on different evaluation datasets [33, 40, 79]. However, highlighting salient regions by backpropagating gradients for downstream applications does not capture how helpful end-users find these attributions [53, 73]. In this section, we take a *human-centred approach to interpretability* – how well our stroke attribution algorithms align with the reasoning process of humans and the trade-off, interpretability vs. accuracy.

Setup: We recruited 7 participants from different geographical regions, in the age group $20-30$ years. All participants had some background in AI research, but only 3 reported having prior experience in interpretability. Once recruited, each user is assigned a unique ID for anonymity. For SLA and P-SLA, we conduct 5 human studies, with each having 10 multiple choice questions (MCQs). Hence, each participant answers 50 MCQs for SLA and 50 for P-SLA.

Evaluating Transparency: For an attribution to be useful, humans must *understand* a model's behaviour for correct and incorrect predictions. In this section, we evaluate if our SLA and P-SLA can make existing sketch classifiers (i.e., Sketch-A-Net [111]), pre-trained on TU-Berlin [33] dataset *transparent* to humans. Accordingly, we choose a random category and select 4 sketch instances – (i) 3 misclassified and 1 correctly classified, and (ii) 1 misclassified and 3 correctly classified. Next, as shown in Fig. 8, we compute stroke attribution (SLA and P-SLA) for the selected sketches and ask users: "Only 1 of these 4 sketches are correctly (or incorrectly) recognised by our model. Please select that correct (or incorrect) sketch.". We find users can identify correct/incorrect model predictions 75.9%/63.4% of times for SLA and 76.3%/65.2% for P-SLA.

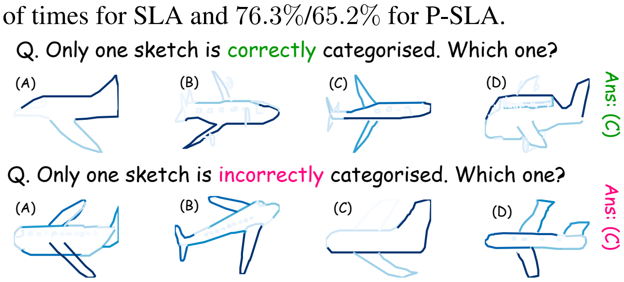

Figure 8. Evaluating transparency: Can a human understand the behaviour of an existing (pre-trained) classifier with SLA/P-SLA.

Evaluating Fairness: End users are much better positioned to make a decision with help from a model if intelligible explanations are provided. In this section, we evaluate if SLA and P-SLA can help end users understand *"What went wrong?"* (i) For a pre-trained sketch classifier [111], we show users a misclassified sketch instance and ask users to identify the (wrongly) predicted category. (ii) For fine-grained SBIR [13], we show a sketch (whose GT photo is not in top-10) and ask users to identify the top-1 (wrongly) retrieved photo in Fig. 9. Humans can identify the misclassified category 62.4%(66.3%) and the incorrectly retrieved photo 39.1%(37.2%) for SLA (P-SLA).

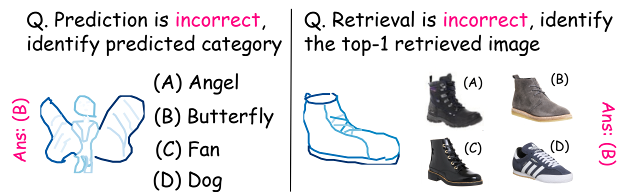

Figure 9. Evaluating Fairness: For an incorrect model prediction, we evaluate if humans can "identify what went wrong".

Evaluting Trustworthiness: Determining trust in individual predictions is important when used for decision-making (e.g., medical diagnosis [57]). We train two copies of the same sketch classifier [111], a strong classifier with 73.8% accuracy on TU-Berlin [33] and a weak one reaching 57.1%. We present the stroke attribution (SLA/P-SLA) for both models and ask users to identify the strong/weak classifier, as shown in Fig. 10. Humans identify the stronger classifier 71.4%(68.7%) of the time for SLA (P-SLA).

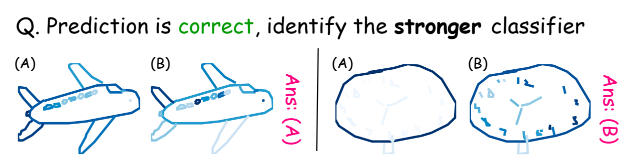

Figure 10. Evaluating trustworthiness: We present the stroke attributions (SLA/P-SLA) from two sketch classifiers (strong and weak), and ask users to identify the strong/weak model.

7. Conclusion

This work emphasises the pivotal role of strokes in human-drawn sketches, offering unique insights compared to pixel-based images. Our lightweight explainability solution seamlessly integrates with pre-trained models, addressing rasterisation challenges and contributing to diverse sketch-related tasks. Through applications in Retrieval, Generation, Assisted Drawing, and Sketch Adversarial Attack, our model showcases adaptability and significance. The proposed stroke-level attribution provides nuanced insights into model behaviour, underscoring the importance of explainability in bridging human expression with model predictions in the evolving field of sketch interpretation.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *NeurIPS*, 2018. 1, 2, 3

[2] Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. *arXiv preprint arXiv:2011.05429*, 2020. 2

[3] Naveed Akhtar, Ajmal Mian, Navid Kardan, and Mubarak Shah. Advances in adversarial attacks and defenses in computer vision: A survey. *arXiv preprint arXiv:2108.00401*, 2021. 7

[4] Shm Garanganao Almeda, J.D. Zamfirescu-Pereira, Kyu Won Kim, Pradeep Mani Rathnam, and Bjoern Hartmann. Prompting for discovery: Flexible sense-making for ai art-making with dreamsheets. *arXiv preprint arXiv:2310.09985*, 2023. 2

[5] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017. 2

[6] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. A unified view of gradient-based attribution methods for deep neural networks. In *NIPSW*, 2017. 2

[7] Anurag Arnab, Ondrej Miksik, and Philip H.S. Torr. On the robustness of semantic segmentation models to adversarial attacks. In *CVPR*, 2018. 7

[8] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 2015. 2

[9] Hmrishav Bandyopadhyay, Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Aneeshan Sain, Tao Xiang, Timothy Hospedales, and Yi-Zhe Song. Sketchinr: A first look into sketches as implicit neural representations. In *CVPR*, 2024. 2

[10] Hmrishav Bandyopadhyay, Subhadeep Koley, Ayan Das, Ayan Kumar Bhunia, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Doodle your 3d: From abstract freehand sketches to precise 3d shapes. In *CVPR*, 2024. 2

[11] 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. 2

[12] Ayan Kumar Bhunia, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Sketch less for more: On-the-fly fine-grained sketch based image retrieval. In *CVPR*, 2020. 1, 5, 6

[13] 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. 8

[14] 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

[15] 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. 1, 4, 5, 6

[16] Moritz B ohle, Mario Fritz, and Bernt Schiele. B-cos: Networks: Alignment is all we need for interpretability. In *CVPR*, 2022. 3

[17] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*, 2019. 2

[18] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *ICCV*, 2015. 2

[19] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. In *NeurIPS*, 2019. 2

[20] Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. Deepfacedrawing: Deep generation of face images from sketches. *ACM TOG*, 2020. 2

[21] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *ICCV*, 2018. 2

[22] Xuesong Chen, Xiyu Yan, Feng Zheng, Yong Jiang, Shu-Tao Xia, Yong Zhao, and Rongrong Ji. One-shot adversarial attacks on visual tracking with dual attention. In *CVPR*, 2020. 7

[23] 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. 2, 5

[24] Pinaki Nath Chowdhury, Aneeshan Sain, Yulia Gryaditskaya, Ayan Kumar Bhunia, Tao Xiang, and Yi-Zhe Song. Fs-coco: Towards understanding of freehand sketches of common objects in context. In *ECCV*, 2022. 2

[25] Pinaki Nath Chowdhury, Tuanfeng Wang, Duygu Ceylan, Yi-Zhe Song, and Yulia Gryaditskaya. Garment ideation: Iterative view-aware sketch-based garment modeling. In *3DV*, 2022. 5

[26] John Collomosse, Tu Bui, and Hailin Jin. Livesketch: Query perturbations for guided sketch-based visual search. In *CVPR*, 2019. 1, 2, 5

[27] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *NeurIPS*, 2017. 2

[28] Ayan Das, Yongxin Yang, Timothy Hospedales, Tao Xiang, and Yi-Zhe Song. Béziersketch: A generative model for scalable vector sketches. In *ECCV*, 2020. 3

[29] 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. 2, 5

[30] Kedar Dhamdhere, Mukund Sundararajan, and Qiqi Yan. How important is a neuron? *arXiv preprint arXiv:1805.12233*, 2018. 2

[31] Gabriele Dominici, Pietro Barbiero, Lucie Charlotte Magister, Liò Pietro, and Nikola Simidjievski. Sharcs: Shared concept space for explainable multimodal learning. *arXiv preprint arXiv:2307.00316*, 2023. 3

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

[33] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM TOG*, 2012. 4, 5, 7, 8

[34] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *Technical Report, Univeristé de Montréal*, 2009. 2

[35] Thomas Fel, Julien Colin, Rémi Cadène, and Thomas Serre. What i cannot predict, i do not understand: A human-centered evaluation framework for explainability methods. *arXiv preprint arXiv:2112.04417*, 2021. 2

[36] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *ICCV*, 2019. 2

[37] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*, 2017. 2

[38] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. Sketchycoco: Image generation from freehand scene sketches. In *CVPR*, 2020. 2

[39] Benoit Guillard, Edoardo Remelli, Pierre Yvernay, and Pascal Fua. Sketch2mesh: Reconstructing and editing 3d shapes from sketches. In *CVPR*, 2021. 2

[40] David Ha and Douglas Eck. A neural representation of sketch drawings. In *ICLR*, 2018. 3, 4, 7, 8

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 6, 7

[42] Yang He, Shadi Rahimian, Bernt Schiele, and Mario Fritz. Segmentations-leak: Membership inference attacks and defenses in semantic image segmentation. In *ECCV*, 2020. 7

[43] Aaron Hertzmann. Why do line drawings work? a realism hypothesis. *Perception*, 2020. 1, 2

[44] Josh Holinaty, Alec Jacobson, and Fanny Chevalier. Supportingreferenceimageryfordigitaldrawing. In *ICCVW*, 2021. 1

[45] Conghui Hu, Da Li, Yongxin Yang, Timothy M Hospedales, and Yi-Zhe Song. Sketch-a-segmenter: Sketch-based photo segmenter generation. *IEEE TIP*, 2020. 2

[46] Quan Huynh-Thu, Marie-Neige Garcia, Filippo Speranza, Philip Corriveau, and Alexander Raake. Study of ratling scales for subjective quality assessment of high definition video. *IEEE TBC*, 2010. 6

[47] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. 5

[48] Yunhan Jia, Yantao Lu, Junjie Shen, Qi Alfred Chen, Hao Chen, Zhenyu Zhong, and Tao Wei. Fooling detection alone is not enough: Adversarial attack against multiple object tracking. In *ICLR*, 2020. 7

[49] Zeyang Jia, Eli Ben-Michael, and Kosuke Imai. Bayesian safe policy learning with chance constrained optimization: Application to military security assessment during the vietnam war. *arXiv preprint arXiv:2307.08840*, 2023. 3

[50] 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. 6

[51] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Viégas, and Michael Terry. Xrai: Better attributions through regions. In *ICCV*, 2019. 2, 3

[52] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 6

[53] Sunnie S. Y. Kim, Nicole Meister, Vikram V. Ramaswamy, Ruth Fong, and Olga Russakovsky. Hive: Evaluating the human interpretability of visual explanations. In *ECCV*, 2022. 2, 3, 8

[54] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, 2019. 2

[55] 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, 4, 5, 6, 7

[56] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *IUI*, 2015. 3

[57] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *IUI*, 2015. 8

[58] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. An evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1902.00006*, 2019. 2

[59] Mengtian Li, Zhe Lin, Radomir Mech, Ersin Yumer, and Deva Ramanan. Photo-sketching: Inferring contour drawings from images. In *WACV*, 2019. 2

[60] Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. Towards transferable targeted attack. In *CVPR*, 2020. 7

[61] Fengyin Lin, Mingkang Li, Da Li, Timothy Hospedales, Yi-Zhe Song, and Yonggang Qi. Zero-shot everything sketch-based image retrieval, and in explainable style. In *CVPR*, 2023. 1, 2

[62] Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, Jing Liao, Bin Jiang, and Wei Liu. Deflocnet: Deep image editing via flexible low-level controls. In *CVPR*, 2021. 2

[63] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. 5, 6

[64] Meike Nauta, Ron Van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *CVPR*, 2021. 2

[65] Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *ICMLW*, 2016. 2

[66] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. 2

[67] 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. 6

[68] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018. 2

[69] Buu Phan, Fahim Mannan, and Felix Heide. Adversarial imaging pipelines. In *CVPR*, 2021. 7

[70] Zhiyu Qu, Yulia Gryaditskaya1, Ke Li, Kaiyue Pang, Tao Xiang, and Yi-Zhe Song. Sketchxai: A first look at explainability for human sketches. In *CVPR*, 2023. 1, 2

[71] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 5

[72] Sukrut Rao, Moritz Böhle, and Bernt Schiele. Towards better understanding attribution methods. In *CVPR*, 2022. 2

[73] Marco T. Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *KDD*, 2016. 3, 7, 8

[74] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bj orn Ommer. High-resolution image synthesis with latent diffusion. In *CVPR*, 2022. 6

[75] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 2019. 3

[76] 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

[77] 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, 3, 5, 7

[78] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE TNNLS*, 2016. 2

[79] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM TOG*, 2016. 4, 5, 8

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

[81] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localisation. In *ICCV*, 2017. 2, 3, 7

[82] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 2

[83] Harshay Shah, Prateek Jain, and Praneeth Netrapalli. Do input gradients highlight discriminative features? In *NeurIPS*, 2021. 2

[84] Hamid Rahim Sheikh and Alan C. Bovik. Image information and visual quality. *IEEE TIP*, 2006. 6

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

[86] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *ICML*, 2017. 2, 3

[87] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLRW*, 2014. 2, 3

[88] Leon Sixt, Martin Schuessler, Oana-Iuliana Popescu, Philipp Weiß, and Tim Landgraf. Do users benefit from interpretable vision? a user study, baseline, and dataset. *arXiv preprint arXiv:2204.11642*, 2022. 2

[89] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 2

[90] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 1

[91] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *ICLRW*, 2015. 2, 3

[92] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, 2017. 2, 3

[93] Christian Szegedy, Wojciech Zaremba, Ilyua Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 7

[94] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 7

[95] Yusuke Tashiro, Yang Song, and Stefano Ermon. Diversity can be transferred: Output diversification for white- and black-box attacks. In *NeurIPS*, 2020. 7

[96] Aditay Tripathi, Rajath R Dani, Anand Mishra, and Anirban Chakraborty. Sketch-guided object localization in natural images. In *ECCV*, 2020. 2

[97] James Tu, Mengye Ren, Siva Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for lidar object detection. In *CVPR*, 2020. 7

[98] Hugues Turbé, Mina Bjelogrlic, Christian Lovis, and Gianmarco Mengaldo. Evaluation of post-hoc interpretability methods in time-series classification. *Nature Machine Intelligence*, 2023. 1, 6
[99] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch your own gan. In *CVPR*, 2021. 2
[100] Donglai Wei, Joseph Lim, Andrew Zisserman, and William T. Freeman. Learning and using the arrow of time. In *CVPR*, 2018. 4
[101] Dennis Wei, Rahul Nair, Amit Dhurandhar, Kush R. Varshney, Elizabeth M. Daly, and Moninder Singh. On the safety of interpretable machine learning: A maximum deviation approach. In *NeurIPS*, 2022. 3
[102] Chufeng Xiao, Deng Yu, Xiaoguang Han, Youyi Zheng, and Hongbo Fu. Sketchhairsalon: deep sketch-based hair image synthesis. *ACM TOG*, 2021. 1
[103] Peng Xu, Yongye Huang, Tongtong Yuan, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, Zhanyu Ma, and Jun Guo. Sketchmate: Deep hashing for million-scale human sketch retrieval. In *CVPR*, 2018. 5, 7
[104] Rui Xu, Zongyan Han, Le Hui, Jianjun Qian, and Jin Xie. Domain disentangled generative adversarial network for zero-shot sketch-based 3d shape retrieval. *AAAI*, 2022. 2
[105] Guowei Yan, Zhili Chen, Jimei Yang, and Huamin Wang. Interactive liquid splash modeling by user sketches. *ACM TOG*, 2020. 2
[106] Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. *IJCV*, 2021. 6
[107] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. In *NeurIPS*, 2019. 2
[108] Sasi Kiran Yelamarthi, Shiva Krishna Reddy, Ashish Mishra, and Anurag Mittal. A zero-shot framework for sketch based image retrieval. In *ECCV*, 2018. 5
[109] Ran Yi, Zipeng Ye, Ruoyu Fan, Yezhi Shu, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin. Animating portrait line drawings from a single face photo and a speech signal. In *ACM SIGGRAPH*, 2022. 2
[110] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *CVPR*, 2019. 2
[111] Qian Yu, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy Hospedales. Sketch-a-net that beats humans. In *BMVC*, 2015. 3, 7, 8
[112] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In *CVPR*, 2016. 3, 5, 6
[113] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 2, 3
[114] Yu Zeng, Zhe Lin, and Vishal M Patel. Sketchedit: Mask-free local image manipulation with partial sketches. In *CVPR*, 2022. 2
[115] Haichao Zhang and Jianyu Wang. Towards adversarially robust object detection. In *ICCV*, 2019. 7
[116] Hua Zhang, Si Liu, Changqing Zhang, Wenqi Ren, Rui Wang, and Xiaochun Cao. Sketchnet: Sketch classification with web images. In *CVPR*, 2016. 5
[117] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *IJCV*, 2018. 2
[118] Lvmin Zhang, Anyi Rao, and Maneesh Agarwala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 3, 5, 6, 7
[119] Song-Hai Zhang, Yuan-Chen Guo, and Qing-Wen Gu. Sketch2model: View-aware 3d modeling from single free-hand sketches. In *CVPR*, 2021. 2
[120] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 2
[121] Hang Zhou, Dongdong Chen, Jing Liao, Weiming Zhang, Kejiang Chen, Xiaoyi Dong, Kunlin Liu, Gang Hua, and Nenghai Yu. Lg-gan: Label guided adversarial network for flexible targeted attack of point cloud-based deep networks. In *CVPR*, 2020. 7
[122] Changqing Zou, Qian Yu, Ruofei Du, Haoran Mo, Yi-Zhe Song, Tao Xiang, Chengying Gao, Baoquan Chen, and Hao Zhang. Sketchyscene: Richly-annotated scene sketches. In *ECCV*, 2018. 2