Supplementary Material
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

Differentiable Rasterisation for P-SLA We differentiably render vector sketches $\mathrm{V} \in \mathbb{R}^{T\times 5}$ (differentiably convert vector $\rightarrow$ raster) by (i) calculating the minimum distance of each pixel in a blank canvas ($\mathrm{X} \in \mathbb{R}^{H\times W\times 3}$) from any stroke in V, and (ii) colouring all pixels by their distance, controlled with threshold hyperparameters. Essentially, these hyperparameters control how thick the rendered strokes are (Fig. S2), by regulating pixels' colour based on how far they sit from the stroke.

We simplify the first problem by calculating the distance of each pixel (p_x, p_y) from *linear segments* of vector strokes $(\overline{v_{t-1}, v_t})$ *i.e.*, consecutive points in V (Fig. S1). Next, we find the minimum of these distances from all such line segments to get the minimum distance from all strokes in V. Now, the distance of pixel coordinate $\mathbf{p}_{xy} = (p_x, p_y)$ from line segment $(\overline{v_{t-1}, v_t})$ is:

$$\texttt{dist}(\mathbf{p}_{xy}, v_{t-1}, v_t) = \begin{cases} |(\overline{\mathbf{p}_{xy}, v_t})|, & \text{if } \angle \mathbf{p}_{xy}\ v_t\ v_{t-1} > \frac{\pi}{2} \\ |(\overline{\mathbf{p}_{xy}, v_{t-1}})|, & \text{if } \angle \mathbf{p}_{xy}\ v_{t-1}\ v_t > \frac{\pi}{2} \\ |(\overline{\mathbf{p}_{xy}, v_t}) \times (\overline{v_t, v_{t-1}})| \div |(\overline{v_t, v_{t-1}})|, & \text{otherwise} \end{cases} \tag{1}$$

where *(i)* $|(\overline{A, B})|$ denotes the euclidean distance between coordinate points A and B, *(ii)* $(\overline{A, B}) \times (\overline{C, D})$ denotes cross product between vectors $(\overline{A, B})$ and $(\overline{C, D})$, and *(iii)* $\angle ABC$ denotes the angle formed by coordinates A, B, and C with B as the vertex.

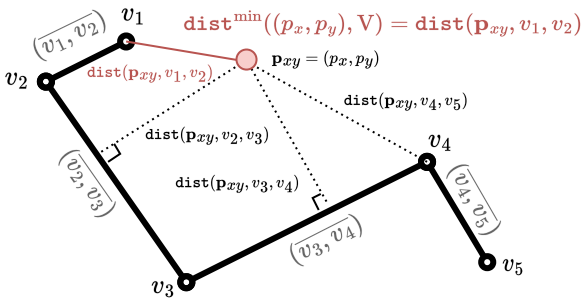


Figure S1. Calculation of minimum distance of pixel-coordinate $\mathbf{p}_{xy} = (p_x, p_y)$ from vector sketch $\mathrm{V} \in \mathbb{R}^{T\times 5}$ where $T = 5$

This equation is implemented as the distance of pixel coordinate $\mathbf{p}_{xy} = (p_x, p_y)$ from line segment $(\overline{v_{t-1}, v_t})$ in Algorithm 1. For each coordinate $\mathbf{p}_{xy}$, we compute the minimum distance as $\texttt{dist}^{\min}(\mathbf{p}_{xy}, \mathrm{V}) = \min_{t=2,\ldots,T}(\texttt{dist}(\mathbf{p}_{xy}, v_{t-1}, v_t))$ over all vector points in V to check whether this distance is under a given threshold. However, not all consecutive points in V are connected. At the end of one stroke (say, v_{t-1}) and the beginning of the next (v_t), the pen is lifted and moved to the new coordinate without drawing on the canvas. This motion is indicated with a pen-up state ($q^1_{t-1} = 0$). To prevent joining these points in $(\overline{v_{t-1}, v_t})$ in the final raster sketch, we exclude them from the minimum distance calculation by offsetting the distance value $\texttt{dist}(\mathbf{p}_{xy}, v_{t-1}, v_t)$ with a large number (10^6). Then, we calculate the minimum distance as :

$$\texttt{dist}^{\min}(\mathbf{p}_{xy}, \mathrm{V}) = \min_{t=2,\ldots,T} \Big(\texttt{dist}(\mathbf{p}_{xy}, v_{t-1}, v_t) + (1 - q^1_{t-1}) 10^6 \Big) \tag{2}$$

The colour (1→black, 0→white) of a pixel $\mathrm{X}(p_x, p_y)$ is determined by the minimum distance $\texttt{dist}^{\min}((p_x, p_y), \mathrm{V})$ where $\mathbf{p}_{xy} = (p_x, p_y)$, as:

$$\mathrm{X}(p_x, p_y) = \sigma(2 - 5 \cdot \texttt{dist}^{\min}((p_x, p_y), \mathrm{V})) \tag{3}$$

where σ represents the sigmoid function and $\mathrm{X} \in \mathbb{R}^{H\times W\times 3}$ represents the raster sketch image. Here, σ acts like a *soft-threshold* to convert $\texttt{dist}^{\min}$ to either $\sim$1 (black colour pixel) or $\sim$0 (white colour pixel). We control stroke thickness in Fig. S2 with the threshold hyperparameters empirically set to 2 and 5.

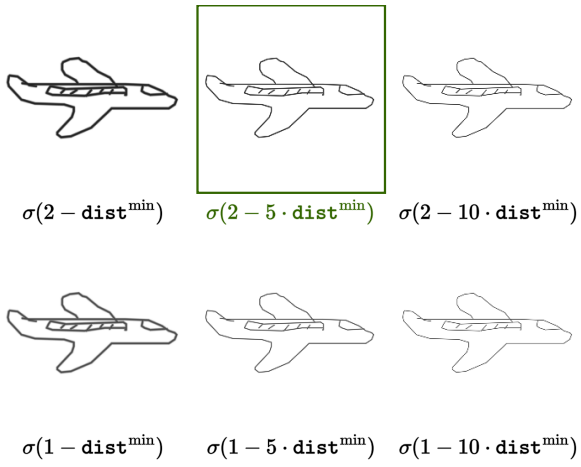
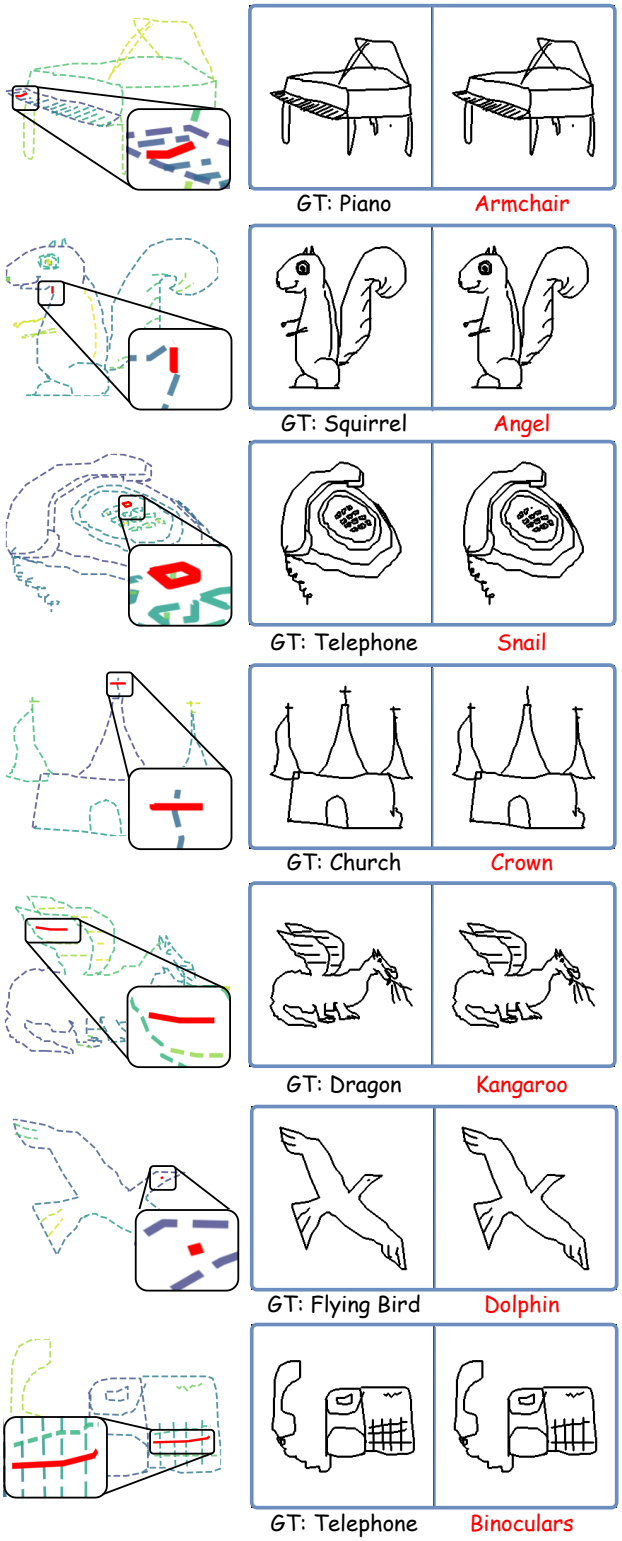
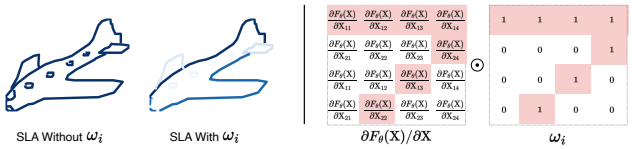


Figure S2. Modulating stroke thickness by a soft-threshold on the minimum distance $\texttt{dist}^{\min}$ of each pixel from any stroke.

Additional Results on Adversarial attacks

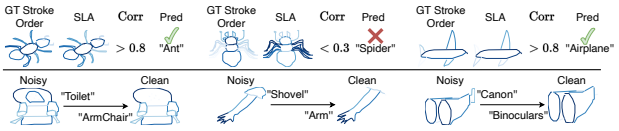


SLA: Avoiding degenerate solutions We use $\omega_i \in \mathbb{R}^{H\times W}$ as a binary mask (1 for stroke pixels, 0 for background). In SLA, we accumulate raster strokes $\mathcal{S}_i \in \mathbb{R}^{3\times H\times W}$ to form the sketch $X = \sum_i \omega_i \mathcal{S}_i$. Here ω_i helps us select only stroke pixels (non-zero pixels which contain the stroke) for the summation. While background pixels (0 by default) do not change the final sketch in the summation, they lead to $\mathrm{X} = \sum_i \mathcal{S}_i \implies \frac{\partial \mathrm{X}}{\partial \mathcal{S}_i} = 1$ for all i. Thus, to obtain gradient of the output $F_\theta(\mathrm{X})$ with respect to any stroke $\mathcal{S}_i$, we compute $\frac{\partial F_\theta(\mathrm{X})}{\partial \mathcal{S}_i} = \frac{\partial F_\theta(\mathrm{X})}{\partial \mathrm{X}} \cdot \frac{\partial \mathrm{X}}{\partial \mathcal{S}_i} = \frac{\partial F_\theta(\mathrm{X})}{\partial \mathrm{X}}$. This term is independent of i, implying all strokes have same attribution (degenerate solution). We infer it arises as 0 pixels are summed up same as non-zero pixels, and have the same gradients as them. Weight ω_i helps us prevent this degenerate solution by only including necessary pixels (i.e. stroke pixels) in the summation.



SLA *vs.* P-SLA A good attribution algorithm should be faithful and human interpretable [73]. While SLA captures stroke-level attributions, P-SLA captures attributions for *co-ordinates* of a stroke. Being *more fine-grained*, P-SLA outperforms SLA on robust SBIR (Tab. 1), noisy stroke removal (Tab. 2), and adversarial attack (Tab. 3). Despite being faithful, P-SLA is less human-interpretable (Fig. 4). Consequently, humans prefer SLA (better MOS scores) over more accurate P-SLA for noisy stroke removal.

Qualitative Results We include some qualitative results for Robust SBIR (Table 1) and Noisy Stroke Removal (Table 2)



Future Directions We introduce fine-grained attributions in sketch-based networks with a plug-and-play explainability toolbox. While we use vanilla gradient-based attributions as a design choice for simplicity, strokes can be attributed with more intricate attribution algorithms [1–4] in future works. Specifically, SLA and P-SLA can be paired with any pixel-attribution algorithm (like Guided Integrated Gradients [2]), where, from attributions of all pixels only those for pixels containing the stroke $(\mathrm{X}(p_x, p_y))$ can be selected. We emphasise that selecting an optimal attribution algorithm is out-of-scope of this paper, as here we primarily demonstrate the applicability of fine-grained attributions in downstream sketch tasks.

Algorithm to compute `dist(·)`

Algorithm 1: Compute `dist`$(\cdot)$

Function `dist` (p_x, p_y, v_{t-1}, v_t)**:**

$(x_{t-1}, y_{t-1}, q^1_{t-1}, q^2_{t-1}, q^3_{t-1}) \leftarrow v_{t-1}$;
$(x_t, y_t, q^1_t, q^2_t, q^3_t) \leftarrow v_t$;
$\delta x \leftarrow x_t - x_{t-1}$;
$\delta y \leftarrow y_t - y_{t-1}$;
norm $\leftarrow px \cdot px + py \cdot py$;
$u' \leftarrow ((p_x - x_{t-1}) \cdot \delta x + (p_y - y_{t-1}) \cdot \delta y$;
$u \leftarrow u'/$norm ;
if $u > 1$ **then**
  $u \leftarrow 1$;
end
else if $u < 0$ **then**
  $u \leftarrow 0$
end
$x \leftarrow x_{t-1} + u \cdot \delta x$;
$y \leftarrow y_{t-1} + u \cdot \delta y$;
$\Delta x \leftarrow x - p_x$;
$\Delta y \leftarrow y - p_y$;
return $(\Delta x \cdot \Delta x + \Delta y \cdot \Delta y)^{1/2}$

References

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

[2] Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. Guided integrated gradients: An adaptive path method for removing noise. In *CVPR*, 2021. 2

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

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