SketchXAI: A First Look at Explainability for Human Sketches

Zhiyu Qu[1,3] Yulia Gryaditskaya[1] Ke Li[1,2] Kaiyue Pang[1] Tao Xiang[1,3] Yi-Zhe Song[1,3]
[1]SketchX, CVSSP, University of Surrey [2]Beijing University of Posts and Telecommunications
[3]iFlyTek-Surrey Joint Research Centre on Artificial Intelligence
{z.qu, y.gryaditskaya, kaiyue.pang, t.xiang, y.song}@surrey.ac.uk
{like1990}@bupt.edu.cn

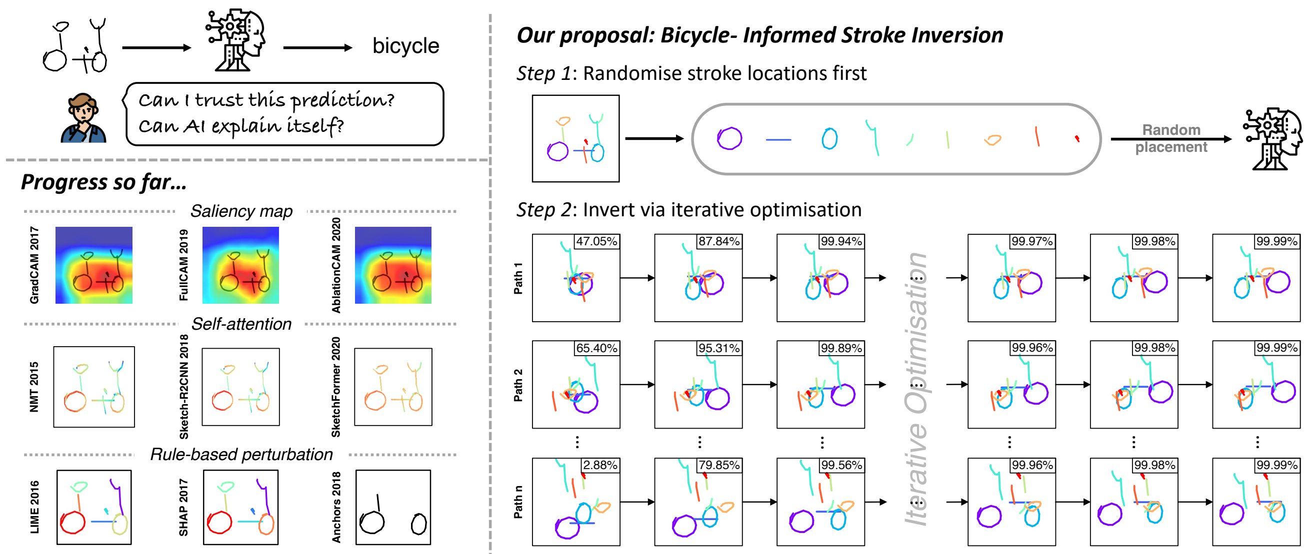

Figure 1. **Explainability, but for human sketches.** We demonstrate a new methodology for explaining AI decisions on human sketch data. Instead of one static explanation per instance as in existing works, our proposed method supports generating infinitely many explanation paths with each dynamically showcasing the inner working of an AI classifier. This enables infinite varieties of explanation paths and allows humans to enjoy a wider coverage on how AI functions, and therefore better scrutinise AI.

Abstract

This paper, for the very first time, introduces human sketches to the landscape of XAI (Explainable Artificial Intelligence). We argue that sketch as a "human-centred" data form, represents a natural interface to study explainability. We focus on cultivating sketch-specific explainability designs. This starts by identifying strokes as a unique building block that offers a degree of flexibility in object construction and manipulation impossible in photos. Following this, we design a simple explainability-friendly sketch encoder that accommodates the intrinsic properties of strokes: shape, location, and order. We then move on to define the first ever XAI task for sketch, that of stroke location inversion (SLI). Just as we have heat maps for photos, and correlation matrices for text, SLI offers an explainability angle to sketch in terms of asking a network how well it can recover stroke locations of an unseen sketch. We offer qualitative results for readers to interpret as snapshots of the SLI process in the paper, and as GIFs on the project page. A minor but interesting note is that thanks to its sketch-specific design, our sketch encoder also yields the best sketch recognition accuracy to date while having the smallest number of parameters. The code is available at `https://sketchxai.github.io`.

1. Introduction

It is very encouraging to witness a recent shift in the vision and language communities towards Explainable AI (XAI) [5,6,40,59,73,75,89]. In a world where "bag of visual words" becomes "bag of tricks", it is critically important that we understand why and how AI is making the decisions, especially as they overtake humans on a series of tasks [21, 26, 52, 67].

XAI research to date has focused on two modalities: photo [15,38,49,88] and text [16,39,42,66,76]. Great strides have been made in the XAI for the photo domain, with the trend of going from heat/saliency maps [11, 64, 68, 70, 86] to the rules/semantics-oriented approaches [28,29,65]. The text side is captivating due to the flexibility of sentence construction. Early works in text models explainability also started with visualisations [1, 68, 86], moving onto linguistic phenomena [8,39,80], and most recently to attention [20,61,71].

In this paper, we make a first attempt at XAI for human freehand sketches. The "why" we hope is obvious – sketches are produced by *humans* in the first place(!), from thousands of years ago in caves, and nowadays on phones and tablets. They are uniquely expressive, not only depicting an object/scene but also conveying stories – see a "Hunter and Arrows" here for a story dating back 25,000 years in France[1]. They, therefore, form an ideal basis for explainability which is also *human-facing*.

The sketch domain is uniquely different from both of the well-studied photo and text domains. Sketch differs from photo in that it can be freely manipulated, while photos are rigid and hard to manipulate. This is largely thanks to the stroke-oriented nature of sketches – jittering strokes might give the "same" sketch back, jittering pixels gives you a "peculiar"-looking image. Sketches have the same level of flexibility in semantic construction as text: strokes are the building block for a sketch as words are for text. With these unique traits of sketch, the hope of this paper is to shed some light on what XAI might look for sketch data, and what it can offer as a result to the larger XAI community. This, however, is only the very first stab, the greater hope is to stir up the community and motivate follow-up works in this new direction of "human-centred" data for XAI.

With that in mind, we focus our exploration on what makes sketches unique – yes, *strokes*. They allow for flexible object construction and make sketches free to manipulate. We then ask how strokes collectively form objects. For that, we identify three inherent properties associated with strokes: shape, location, and order. These three variables define a particular sketch: *shape* defines how each stroke looks like, *location* defines where they reside, and *order* encodes the temporal drawing sequence.

Our first contribution is a sketch encoder, that factors in all the mentioned essential properties of strokes. We hope that this encoder will build into its DNA how strokes (and in turn sketches) are represented, and therefore be more accommodating when it comes to different explainability tasks (now and in the future) – and for this, we name it SketchXAINet ("X" for E*X*plainability). We are acute to the fact that explainability takes simple forms [48], so we refrained from designing a complicated network. In fact, we did not go any further than introducing a branch to encode each of the three stroke properties (shape, location, and order), and simply feed these into a standard transformer architecture with a cross-entropy loss. Interestingly, however, just with this simple architecture, we already observe state-of-the-art sketch recognition performance improving on all prior arts.

With an explainability-compatible sketch encoder in place, we now want to examine if we can actually make anything explainable. First and foremost, of course, sketch explainability can be performed in the form of a heat map [11, 64, 70] – just treat sketches as a raster image and we are done. This, however, would be entirely against our very hope of spelling out sketch-specific explainability – the "explainability" one can obtain there is *at best* at the level of photo heatmaps (see Fig. 1).

Instead, we utilise our sketch encoder and put forward the first XAI task for sketch – that of stroke location inversion (SLI) (see Figs. 1 and 3). We study two types of tasks: recovery and transfer. Intuitively, during the recovery, we ask our optimisation procedure to jitter the stroke locations to *recover* sketch so that it belongs to the same class as the original sketch. During the transfer task, we ask our optimisation procedure to jitter the stroke locations to obtain a sketch that belongs to a new class that we pass as input to the optimiser. The idea is then that how well the network has learned is positively correlated with how well it does at this inversion task, and that explainability lies in visualising this process. So, in addition to heat maps for photos, and correlation matrices for text, for sketch, we now have visualisations, that theoretically be manifested of infinite variety, and in the form of a video/GIF to capture the SLI process. We finish by playing with variants of the proposed SLI: (i) sketch recovery, to offer insights on category-level understanding of a learned encoder, *i.e.*, reconstructing a sketch to the same category, and (ii) sketch transfer, to shed light on cross-category understanding, *i.e.*, using strokes of one category to reconstruct another.

Our contributions are as follows: (i) we argue for sketches to be introduced to the field of XAI, (ii) we identify strokes as the basic building block and build a sketch encoder, named as SketchXAINet, that encapsulates all unique sketch properties, (iii) we introduce stroke location inversion (SLI) as a first XAI task for sketch, (iv) we offer qualitative results of the inversion process and deliver best sketch recognition performance as a by-product.

2. Related work

Raster and vector sketch encoders. Sketch contains high-level human understanding and abstraction of visual signals and is a distinctive modality to photos. Many of the previous works [31, 36, 41, 53, 54, 57, 62, 63, 79], how-

[1] https://www.worldhistory.org/Lascaux_Cave/

ever, treat sketches with no difference to photos – they operate on raster format and feed them into contemporary CNNs for visual learning. Facilitated by the availability of sketch datasets with stroke-level information [17, 19], there is an ongoing trend of works that turn to model sketch as a temporal sequence of vector coordinates, hoping to open up new research insights for downstream sketch tasks [12, 33, 34, 37, 45, 51, 69, 83, 84]. Along with this representation change on sketch data is also the backbone upgrade, from CNN to Transformer [37, 58], the choice of which we also embrace in constructing our proposed sketch encoder. Scarcely few existing works have anchored their focus on the explainability of sketch models, with [51] [3] being moderately relevant to our best knowledge. At a high level, both works, just like ours, explore the impact of strokes on forming a sketch object. But instead of studying sketch abstraction, *i.e*., how strokes can be deleted or simplified without altering the holistic semantic meaning, we leverage the free-hand stroke itself as a building block to understand sketch model explainability.

Ante-hoc and post-hoc explainability methods. Several recent surveys and books discuss explainability methods in detail [4, 5, 24, 49]. Explainability methods are often split into two groups: *ante-hoc* [10, 32] and *post-hoc* [25, 47, 59, 60, 86, 89] methods. Ante-hoc methods are inherently and intrinsically interpretable, while post-hoc methods require designing separate techniques to provide probes into model explainability. The former, also known as the white/glass box approach, is preferable under the context of explainability, but limited by a few specific choices of instantiations, *e.g*., decision trees [78], generalised additive models [2]. The latter being less transparent has no restrictions on model learning and therefore often achieves better test-time task performance. Achieving the optimal trade-off of such is then the core to both schools of explainable AI [5, 24]. Our proposed sketch explainability method SLI is post-hoc, but facilitated by a tailor-designed, less black-box (ante-hoc alike) sketch encoder (that allows reasoning over a stroke-based decision into shape, location, and order). Notably, our final sketch model achieves state-of-the-art recognition performance.

Counterfactual explanation and adversarial attack. Our post-hoc explainability strategy SLI of "reshuffling first, recovery later" is also reminiscent of a specific AI explainability genre – counterfactual generation (CE) [27, 44, 77]. CE aims to provide explanations of a model by identifying the minimal changes required to revert the original prediction. If these compact but essential components do correspond to the most important visual semantics discriminating and defining an object, a model prediction is believed to have passed a confidence test. In this sense, SLI identifies the strokes that actually matter (*e.g*., the tires and the front handle for a bicycle in Fig. 1) through multiple randomly initialised counterfactual inversion tasks (because important strokes gets highlighted across trials). Closely related to counterfactual inversion is another field known of adversarial attack [7, 18, 50, 72], which aims at the generation of adversarial examples (AE) having imperceptible differences to human vision but results in completely different AI predictions. Conceptual connections between CE and AE have been extensively discussed in the literature [9, 56, 77], where [56] suggests that AE is part of a broader class of examples represented by CE. Our SLI built upon spatial reconfiguration of strokes differentiates from AE by definition – the movement of strokes is less likely to be imperceptible changes compared with those by local pixel jittering.

3. Methodology

In this section, we first introduce our classification model which is designed around strokes as sketch building blocks. We then introduce our method for model explainability.

As a pre-processing step, we simplify all sketches by the RDP algorithm [14]. For each stroke s_i consisting of k points, $\{s_{i,1}, s_{i,2}, ..., s_{i,k-1}, s_{i,k}\}$, we identify three inherent properties and learn respective descriptor for each: location l_i, shape sh_i and stroke order o_i. We use the starting point of s_i in absolute coordinate to encode l_i, *i.e*., $s_{i,1}$. In case of notation confusion, we leverage (x_i, y_i) as an alternative to $s_{i,1}$. As for sh_i, in order to be location-agnostic, we've done two things: use relative coordinates and require the same fixed starting point for all strokes, the canvas origin middle point in our case. As per convention, each sh_i point also contains a two-dimensional binary pen state [19] – (1, 0): stroke is being drawn, (0, 1): the end of the stroke, (0, 0): padding points to account for the fact that all strokes have a different number of points.

Sketch-specific encoder. Our proposed sketch encoder f_w, which we name SketchXAINet ("X" for E*X*plainability), separately reasons over l_i, sh_i and o_i before combining force for final decision. This tailored model design is then ready to undertake the novel explainability task defined later. A full high-level schematic is shown in Fig. 2. We use a bidirectional LSTM [23] to extract shape information of each stroke sh_i, and one linear layer for location l_i embedding learning. We pre-define the maximum number of strokes allowed and assign a learnable embedding for each order (time) embedding o_i. Finally, we sum them all and add one extra `[CLS]` token before feeding into a transformer encoder [13]. We adopt `[CLS]` for classification task, optimised under the conventional multi-class cross-entropy loss.

Sketch explainability - SLI. We introduce a new task for explaining sketch model, that of *Stroke Location Inversion*, SLI. Initiating from replacing each sketch stroke at a random location, SLI explains a sketch classifier through answering the question: can the classifier invert this random sketch

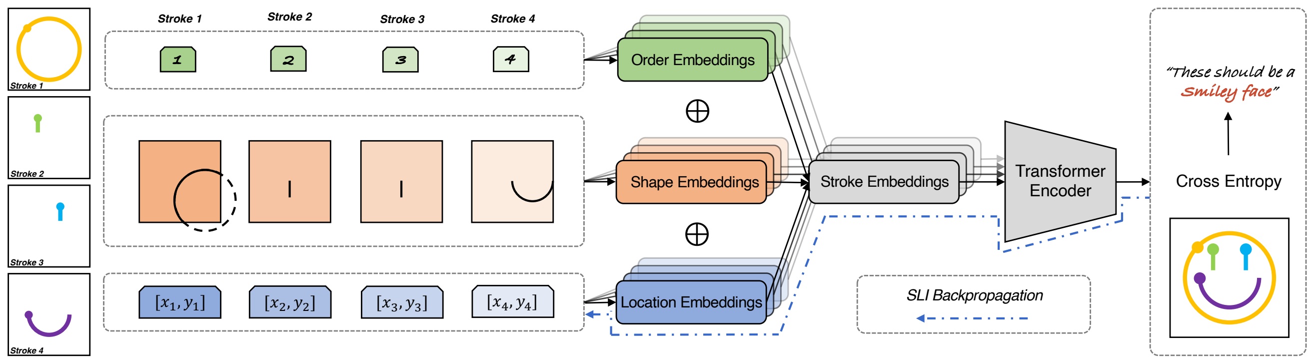

Figure 2. **SketchXAINet architecture.** We build a sketch classifier upon stroke vectors rather than raster pixels. All strokes are decomposed into three parts – order, shape and location. We use a bidirectional LSTM, a linear model and a learnable time embedding matrix to encode such decomposed stroke representation respectively. The dashed line refers to the gradient flow of the location parameters when we generate explanations by SLI with a trained classifier.

back to the visual semantics it should possess, and by doing so one is able to probe into the internal state of a once black-box classifier and therefore achieve explanation. SLI corresponds to an iterative optimisation problem dedicated to reconfigure strokes locations for increasing recognition confidence and a dynamic visualisation path for humans to scrutinise. Denoting a sketch composing of N strokes with class label y in bold $\mathbf{s}$, this process is formulated as:

$$\arg\min_{l_1,\cdots,l_N} \mathcal{L}\left(f_w\left(\text{Replacement}(\mathbf{s})\right), y\right), \tag{1}$$

Note that only because our proposed f_w disentangles l_i learning from everything else that enables such inversion.

In connection to counterfactual & latent optimisation. At first glimpse, SLI draws considerable similarity to counterfactual explanation – finding input variations that lead to complete change of prediction outcomes. We adapt this definition under our context with a slight modification to its original formulation [77]:

$$\arg\min_{l_1,\cdots,l_N} \mathcal{L}\left(f_w\left(\mathbf{s}'\right), y'\right) + d\left(\mathbf{s}, \mathbf{s}'\right), \tag{2}$$

where y' denotes another label different from y, $d(\cdot)$ is some distance measure and can be a simple sum of location difference here. The advantage of SLI becomes evident under such comparison that unlike the counterfactual approach restricted by the fixed optimisation starting point $\mathbf{s}$ and a local input search space, SLI enjoys a much bigger flexibility with each time explaining a different facet of fact (through random replacement of $\mathbf{s}$). Optimising towards rather than against correct labels also makes explanation less susceptible to adversarial examples. SLI is also connected to latent optimisation, a technique extensively explored in GAN literature [81]. If we dissect f_w into $f_l \circ f_{w\setminus l}$ and draw an analogy to the latent vector z and generator $G(\cdot)$ in GAN language respectively, this becomes a standard GAN inversion problem. The difference is instead of traversing along the non-interpretable z space, f_w is interpretable in nature with each update dictating the direction and pace of the next stroke movement.

Formal Definition. We now define two types of SLI tasks, where stroke relocation is leveraged as a gateway to explaining a sketch classifier. *Recovery:* During the recovery task, we randomise the locations of all strokes and only keep their shapes. We specify the target label y as the original sketch label and use Eq. (1) to optimise $(l_1, \cdots, l_N)$. We study the entire optimisation process to understand the inner workings of the classifier. *Transfer:* For the transfer task, we keep stroke shapes and locations intact, while specifying the target label y as a different category to that of the input sketch. We use this setup to build cross-category understandings.

4. Experiments

4.1. Experimental Settings

We adopt the QuickDraw dataset [19] to train f_w, which contains 345 object categories with 75K sketches each. Following convention the 75K sketches are divided into training, validation and testing sets with size of 70K, 2.5K and 2.5K, respectively. For the analysis of generated explanations by SLI, we randomly select 30 categories. We compare our model with a variety of sketch recognition models: CNN-based [22,85], hybrid-based [35,82,83] and Transformer variants [13, 43, 58]. We use the same learning rate of 0.00001, `Adam` optimiser [30], and 20 epochs for all methods. All experiments of this stage are run on 5 NVIDIA 3090 GPUs with a batch size of 100 per GPU. For better SLI training stability, we use `gradient clip` [55], `CosineAnnealingLR` scheduler [46] and `SGD` optimiser without momentum to limit the distance a stroke can move.

Methods	Acc. (%)	Params
ResNet-50 [22]	78.76	24.2
Sketch-a-Net [85]	68.71	<u>8.5</u>
SketchMate [82]	80.51	64.7
ViT-Base [13]	77.90	86.6
Swin-Base [43]	78.71	87.8
SketchFormer [58]	78.34	13.1
SketchAA [83]	81.51	26.7
Sketch-R2CNN [35] (ResNet-50)	84.81	32.7
Sketch-R2CNN [35] (ResNet-101)	85.30	51.7
SketchXAINet-Tiny (No Shape)	31.04	-
SketchXAINet-Tiny (No Location)	81.41	-
SketchXAINet-Tiny (No Order)	83.66	-
SketchXAINet-Tiny	<u>85.93</u>	**6.1**
SketchXAINet-Base	**87.18**	91.7

Table 1. Recognition accuracy (%) and parameters (million) of different methods on 345 categories of QuickDraw [19] dataset. Sketch-R2CNN is the previous SoTA. **Bold** and <u>underline</u> denote the best and the second best method. -Base / -Tiny follow the architecture setting in the original ViT work.

4.2. Main Results

SLI achieves SoTA sketch recognition. We use top-1 classification accuracy to assess the sketch recognition task. Tab. 1 shows performance comparison between all selected models and ours. We include all five major sketch recognition works in contemporary time, Sketch-a-Net [85], SketchMate [82], SketchAA [83], SketchFormer [58] and Sketch-R2CNN [35] and find out Sketch-R2NN has significant edges over others. We also experiment with not sketch-specific but more mainstream vision representation learning architecture, Vision Transformer (ViT) [13] and its more advanced variant Swin Transformer [43]. Both however is only on par to SketchFormer, a Transformer-based framework on point, other than patch pixel embedding. SketchXAINet demonstrates that Transformer *can* outperform CNN (Sketch-R2CNN with ResNet-101) on sketch recognition tasks. We achieve a new state-of-the-art sketch recognition performance, improving on all prior arts. We also conduct controlled study to verify the relative importance of ech component in our decomposed stroke representation. Without surprise, the shape feature plays a major role while the order information is the least important.

SLI provides probe for understanding deep classifier. Fig. 3 shows the generated visual explanations with SLI taking effect in both recovery and transfer tasks. We first analyse the recovery results with the following observations: i) despite the recovered sketches are often visually different from the original inputs, they reveal the essential category-specific semantics for viewers to interpret, and in turn, build their *own explainability* on how trustworthy the current classifier prediction is. For instance, in the `[sun]` case, the classifier learns the concept of light by trying to relocate random clustered strokes back to the surroundings of the circle. It is also a bit surprising to see in the `[tree]` case that the classifier has fostered fine-grained understanding by even mainly relocating one single stroke, that from the flower stem to the tree trunk. ii) The iteration steps to which optimisation converges vary across samples and randomised starting points, with 100 iterations being a generous enough budget for all scenarios and only taking a few seconds on a modern GPU. Iterative optimisation also allows viewers to selectively look into the explanation path and identify far more diverse evidence for AI attribution than that of final static output. In the `[cell_phone]` example, the classifier seems to have not learned a solid correct spatial composition whereas in `[tree]` the classifier while being acute to conceptual difference is also not bullet-proof for human scrutiny – after the 1^{st} iteration, the recognition confidence reaches 95.45% compared to 32.81% but without convincing visual effect change. iii) Randomisation provides a contrastive way to explain different functioning facets of a classifier and thus leaves viewers a better place to decide whether and to what extent to build a AI trust case. Through comparison, we can, for example, establish trust by setting up a minimum recognition confidence baseline for each category, that is we can't trust a prediction unless it is confident up to a level. This conclusion stems from our *dynamic* visualisation that different random starting points dictate a different exposure on classifier and in some cases even with more than 95% recognition confidence can it be less reassuring, *e.g.*, `[sun]`. Randomisation here, therefore, serves as a generative explanation role so that viewers have enough examples and interpret a classifier statistically. Back to the transfer task, we can see that the generated explanation path becomes less effective but still partially understandable. Even the stroke shapes making up different categories have significant visual independence, SLI is able to deliver a sensible message by putting strokes at the right place representing the *just right abstraction* of visual semantics. The seat stroke of a `[chair]` turns into the head of a `[broom]` and the `[bicycle]` is totally anatomised to resemble the looking of a `[camera]`. Downsides of a classifier are also implied where inverting a `[sun]` into `[apple]` reveals the vulnerability of the apple classifier under a pineapple attack. In summary, SLI provides an interpretable tool to visually probe into the functioning of a sketch classifier and enable various AI explainability projections.

4.3. On Stroke Shape Embedding

To analyse our learned shape embedding[2], we conduct t-SNE [74] across the strokes of the selected samples from all sketch categories and run K-means on their reduced di-

[2]Analysis on order embedding can be found in the supplementary.

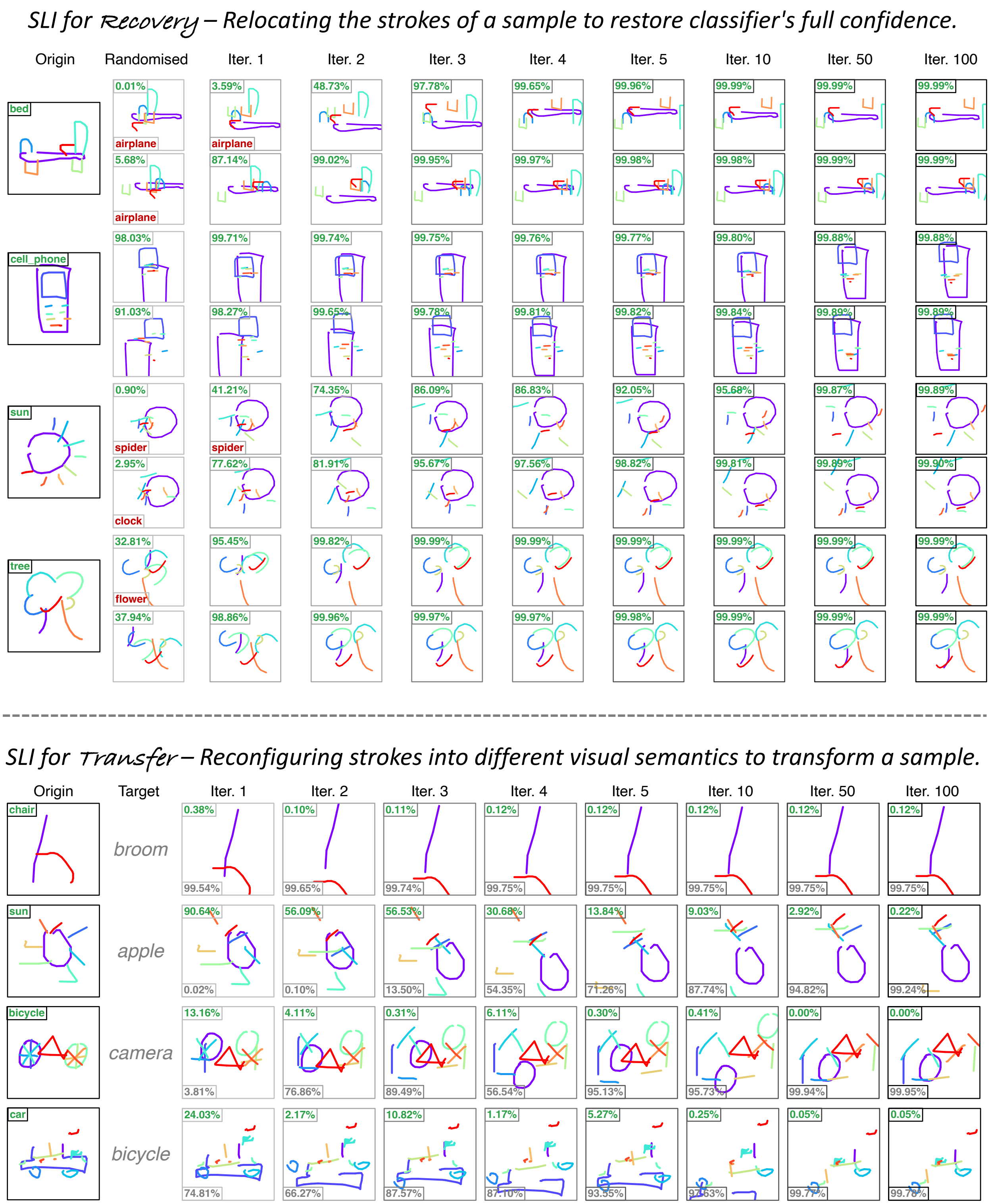

Figure 3. **SLI explains SketchXAINet in Recovery and Transfer tasks.** Here we show the visualisations of the 100 optimisation steps of SLI (Eq. 1). Origin refers to a free-hand sketch sampled from the Quickdraw dataset, where in recovery we randomise its constituent strokes to form different explainable inputs, and in transfer, we keep it intact but leverage it to explain a classifier of the different target category. The number in the top-left corner (the bottom-left corner when present) indicates model confidence in the current sketch to belong to the original label (to the new counterfactual label). We use bounding boxes with gradient colours (from light grey to black) to highlight the progressive nature of SLI.

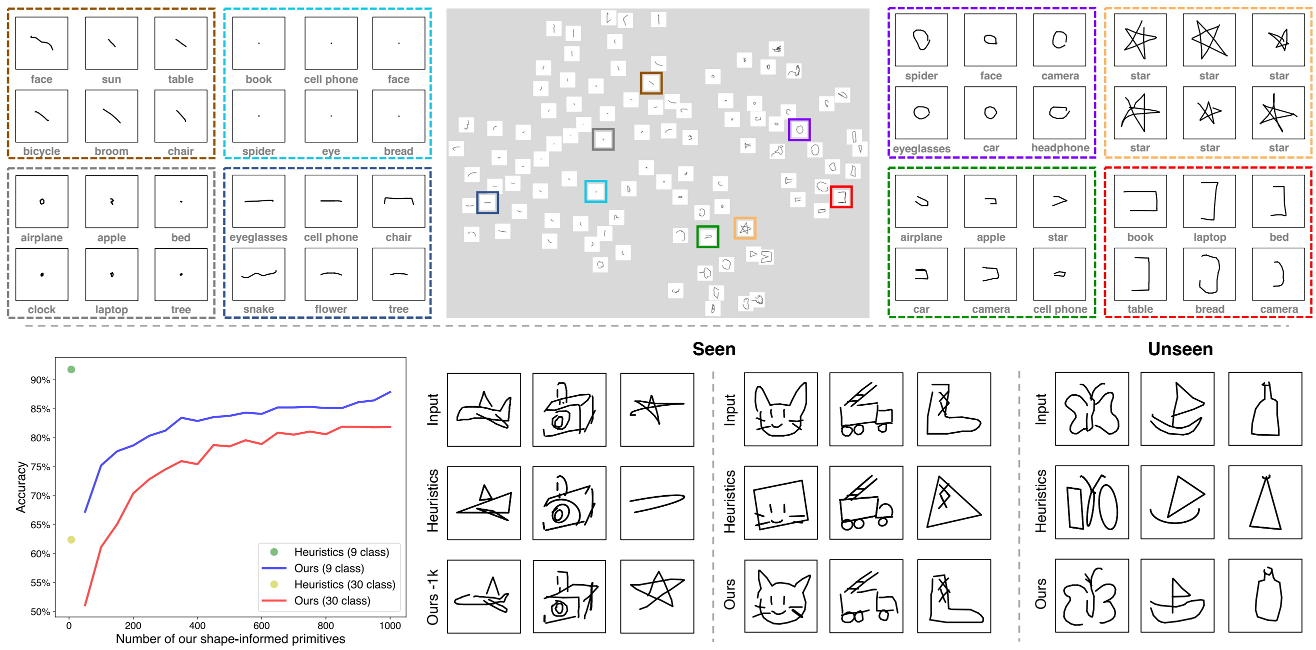

Figure 4. **Analysis on shape embedding.** Top: t-SNE visualisation on 100 stroke primitives across 30 sketch categories. Strokes with similar semantics are grouped together regardless of the original categories sourced from. Bottom: we compare our learned stroke primitives with [3], where 7 stroke primitives are heuristically pre-defined and their efficacy to reconstruct a sketch (*i.e.*, replace any stroke with a primitive) is evaluated on a carefully curated 9-class setting. The table shows the method largely fails when extending the evaluation to a more open-world setting of 30 classes. Ours can not only deal with less regularised sketches from seen classes (*e.g.*, star), but also generalises well to unseen cases.

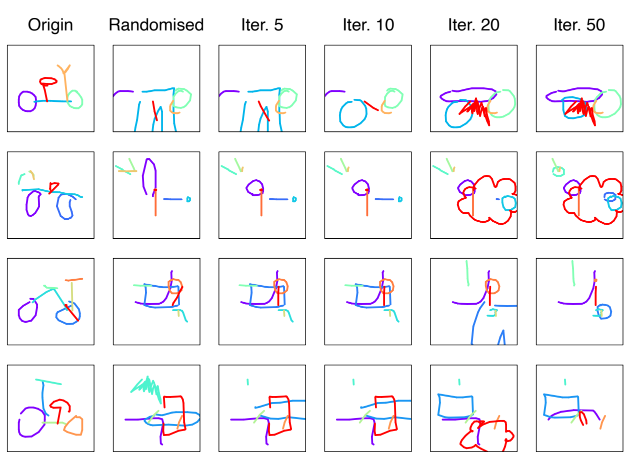

Figure 5. **Shape, not location, Inversion.** With automatically generated stroke primitives, we can now proceed inversion tasks on stroke shapes, just like how we do for locations – updates on high-dimensional shape embedding can be now visualised to changes of shape primitives if that update becomes significant enough. We however fail to identify explainable factors in such inversion.

mensions. We simply define each cluster centroid as the stroke sample (during training) closest to and see that as the *representative stroke primitive* of all stroke samples belonging to the same centroid. A natural outcome is that the larger centroid numbers we set in K-means, the finer primitives incorporating more diverse drawing styles are expected. The first row of Fig. 4 shows the t-SNE clustering results with 100 centroids on 30 sketch categories and confirms the shape embedding has formed semantics understanding to group visually similar strokes together regardless of the original category they come from – see how dots with different hollow types are well recognised by the embedding. For more quantitative evaluation, we replace all strokes of a sketch sample with their primitives and feed them into SketchXAINet for classification. Comparing with the results reported in the past work [3] which manually define a fixed set of heuristics-based shape primitives (line, arc, square, circle, triangle, U-shape, L-shape), our learning-based method is flexible in how a stroke is to be abstracted and how to trade-off recognition at the whole sketch level therein. We demonstrate the comparison in the bottom row of Fig. 4. Apart from the 9-class setting from [3] that specifically choose certain classes with visual semantics biased to their analysis (*e.g.*, round-shaped silhouette), [3] mostly fails under more open setting, with recognition accuracy plummeting from 91.8% to 62.4% in 30-class setting and complete reconstruction failure for less regularised sketch samples (*e.g.*, shoe, star). Finally, with learned stroke primitives, we can now try to conduct shape, rather than stroke inversion explainability task by modifying Eq. 1 to optimise $sh_1, sh_2, ..., sh_n$ in-

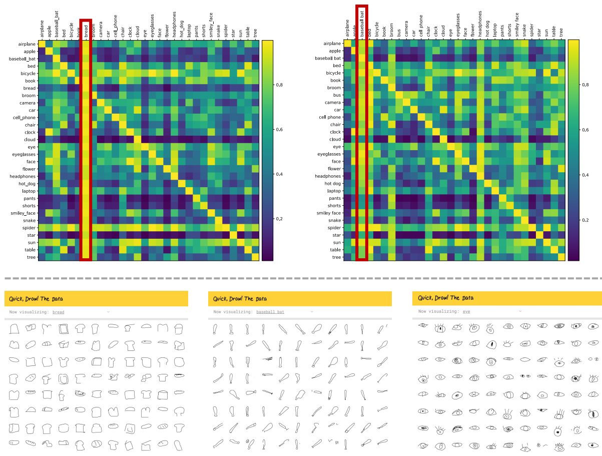

Figure 6. **SLI exposes dataset bias**. Top: we apply SLI on transfer tasks between every two categories out of a total of 30 and observe all sketch samples regardless of the origin can be transferred to `[bread]` (left). To confirm, we exclude `[bread]` and replace it with a new category `[bus]` and this time all sketches transfer to `[baseball_bat]`. Bottom: we showcase the screenshots (best view in zoom) of three QuickDraw categories, `[bread]`, `[baseball_bat]`, `[eye]`, which yields an explanation to the said phenomenon. More details in text.

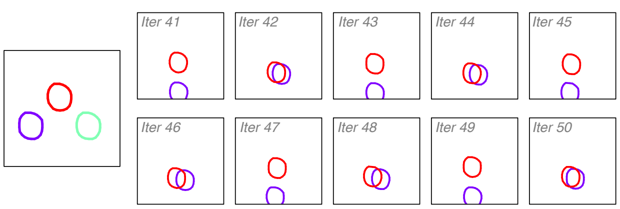

Figure 7. **Limitation.** SLI relies on gradient descent and thus inherits its weakness. Here we demonstrate with a simple `sun` transfer task how optimisation is trapped in local optima.

stead. After each gradient descent, we replace the updated shape embeddings with their closest primitives and use them as initialisation for the next step. Examples in Fig .5 show that shape inversion hardly delivers any explainable outcome and implicitly justifies our location inversion choice.

5. Discussion

Explaining dataset bias with SLI In our transfer explainability setting, we showed that by relocating the strokes and in some cases removing the strokes from the canvas (moving them out of the canvas bounding box) we can transfer a sketch from category A to category B. Here, we conduct an additional experiment. We sample 100 sketches for each of the 30 training categories and apply a transfer task for each pair of sketches. In the top part of Fig. 6, we visualise as a heat map the average recognition confidence values to belong to the target category of sketches transferred from one category to another. We find that for almost all sketch categories the average confidence is high for a transfer to a sketch of `[bread]`. Then, we naturally ask the question of how this behaviour can be explained. We start by looking at the example of the sketches from the `[bread]` category. In Fig. 6 bottom, we show sketch samples from the QuickDraw dataset for bread sketches[3], we can see that many look like something else, e.g. a `[shirt]`. Our SLI task allowed us to find a category for which sketches are ambiguous with respect to an assigned category. The next category with high average confidence of the transfer task, `[baseball_bat]`, also contains many ambiguous sketches, for example, resembling a `[knife]`. We also show the `[eye]` sketches, which we find to be the category hardest to transfer to. We can see that all sketches do look like eyes. Therefore, we can see how our SLI task can help to identify categories for which humans struggle to produce easily recognisable sketches. Such dataset bias needs to be taken into account when training deep models. To conclude, this pilot study provides further insights into how SLI contributes towards explainability.

Limitation. SLI is based on gradient descent and therefore inherits its limitations: SLI can be susceptible to local optima by oscillating around stroke location and not progressing further. We exemplify this in Fig. 7 where we use three circles to explain the sun concept. The expectation is then that two circles will be driven away off the canvas and one circle left. In practice, however, one circle is driven away and two circles are trapped in a tug-of-war. Solutions to alleviate this issue can be inspired by the optimisation literature, *e.g.*, look ahead optimiser [87] is designed to break the optimisation deadlock by maintaining two sets of fast and slow weights.

6. Conclusion

Sketches form a great data modality for explainability research because of their inherent "human-centred" nature. We started our journey by first identifying strokes as the basis for explanation. We then introduced SketchXAINet to encode the three innate properties of sketch strokes: shape, location, and order. Leveraging this encoder, we propose the first sketch-specific explainability task, that of stroke location inversion (SLI). Compared to your typical static explanations (*e.g.*, saliency map), SLI is a dynamic process that explains the credibility of a sketch model by examining its ability to relocate randomly reshuffled strokes to reconstruct a sketch given a category. We attest to the efficacy of SLI with extensive analysis and contribute a new SoTA sketch recognition model as a by-product. Last but not least, we repeat that this is only the very first stab, yet at what we believe to be a very important and interesting area for XAI.

[3] https://quickdraw.withgoogle.com/data/bread

References

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

[2] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *NeurIPS*, 2021. 3

[3] S. Alaniz, M. Mancini, A. Dutta, D. Marcos, and Z. Akata. Abstracting sketches through simple primitives. In *ECCV*, 2022. 3, 7

[4] Anna Markella Antoniadi, Yuhan Du, Yasmine Guendouz, Lan Wei, Claudia Mazo, Brett A Becker, and Catherine Mooney. Current challenges and future opportunities for XAI in machine learning-based clinical decision support systems: a systematic review. *Applied Sciences*, 2021. 3

[5] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 2020. 1, 3

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 1

[7] Shumeet Baluja and Ian Fischer. Adversarial transformation networks: Learning to generate adversarial examples. In *AAAI*, 2017. 3

[8] Terra Blevins, Omer Levy, and Luke Zettlemoyer. Deep rnns encode soft hierarchical syntax. In *ACL*, 2018. 2

[9] Kieran Browne and Ben Swift. Semantics and explanation: why counterfactual explanations produce adversarial examples in deep neural networks. *arXiv:2012.10076*, 2020. 3

[10] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *SIGKDD*, 2015. 3

[11] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*, 2018. 2

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

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3, 4, 5

[14] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the International Journal for Geographic Information and Geovisualization*, 1973. 3

[15] Hugo Jair Escalante, Sergio Escalera, Isabelle Guyon, Xavier Baró, Yağmur Güçlütürk, Umut Güçlü, Marcel van Gerven, and Rob van Lier. *Explainable and interpretable models in computer vision and machine learning*. Springer, 2018. 2

[16] Andrea Galassi, Marco Lippi, and Paolo Torroni. Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2020. 2

[17] Songwei Ge, Vedanuj Goswami, C Lawrence Zitnick, and Devi Parikh. Creative sketch generation. In *ICLR*, 2020. 3

[18] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 3

[19] David Ha and Douglas Eck. A neural representation of sketch drawings. In *ICLR*, 2018. 3, 4, 5

[20] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Self-attention attribution: Interpreting information interactions inside transformer. In *AAAI*, 2021. 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 1

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 5

[23] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997. 3

[24] Andreas Holzinger, Georg Langs, Helmut Denk, Kurt Zatloukal, and Heimo Muller. Causability and explainability of artificial intelligence in medicine. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2019. 3

[25] Mark Ibrahim, Melissa Louie, Ceena Modarres, and John Paisley. Global explanations of neural networks: Mapping the landscape of predictions. In *AAAI*, 2019. 3

[26] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 2021. 1

[27] Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. In *NeurIPS Workshop*, 2020. 3

[28] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 2

[29] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *ICML*, 2018. 2

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4

[31] Brendan Klare, Zhifeng Li, and Anil K Jain. Matching forensic sketches to mug shot photos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010. 2

[32] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, and David Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 2015. 3

[33] Changjian Li, Hao Pan, Adrien Bousseau, and Niloy J Mitra. Free2cad: parsing freehand drawings into cad commands. *ACM Transactions on Graphics*, 2022. 3

[34] Ke Li, Kaiyue Pang, Jifei Song, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Honggang Zhang. Universal sketch perceptual grouping. In *ECCV*, 2018. 3

[35] Lei Li, Changqing Zou, Youyi Zheng, Qingkun Su, Hongbo Fu, and Chiew-Lan Tai. Sketch-r2cnn: An attentive network for vector sketch recognition. *IEEE Transactions on Visualization and Computer Graphics*, 2020. 4, 5

[36] Yi Li, Timothy M Hospedales, Yi-Zhe Song, and Shaogang Gong. Fine-grained sketch-based image retrieval by matching deformable part models. In *BMVC*, 2014. 2

[37] Hangyu Lin, Yanwei Fu, Xiangyang Xue, and Yu-Gang Jiang. Sketch-bert: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. In *CVPR*, 2020. 3

[38] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable AI: A review of machine learning interpretability methods. *Entropy*, 2021. 2

[39] Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 2016. 2

[40] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 2018. 1

[41] Fang Liu, Changqing Zou, Xiaoming Deng, Ran Zuo, Yu-Kun Lai, Cuixia Ma, Yong-Jin Liu, and Hongan Wang. Scenesketcher: Fine-grained image retrieval with scene sketches. In *ECCV*, 2020. 2

[42] Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaicheng Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, Zi-Yi Dou, and Graham Neubig. Explainaboard: An explainable leaderboard for nlp. In *ACL*, 2021. 2

[43] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 4, 5

[44] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. In *ECML PKDD*. Springer, 2021. 3

[45] Raphael Gontijo Lopes, David Ha, Douglas Eck, and Jonathon Shlens. A learned representation for scalable vector graphics. In *CVPR*, 2019. 3

[46] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 4

[47] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NeurIPS*, 2017. 3

[48] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 2019. 2

[49] Christoph Molnar. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book, 2022. 2, 3

[50] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, 2017. 3

[51] Umar Riaz Muhammad, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning deep sketch abstraction. In *CVPR*, 2018. 3

[52] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022. 1

[53] 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. 2

[54] 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

[55] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML*, 2013. 4

[56] Martin Pawelczyk, Shalmali Joshi, Chirag Agarwal, Sohini Upadhyay, and Himabindu Lakkaraju. On the connections between counterfactual explanations and adversarial examples. In *AISTATS*, 2022. 3

[57] Yonggang Qi, Yi-Zhe Song, Tao Xiang, Honggang Zhang, Timothy Hospedales, Yi Li, and Jun Guo. Making better use of edges via perceptual grouping. In *CVPR*, 2015. 2

[58] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. Sketchformer: Transformer-based representation for sketched structure. In *CVPR*, 2020. 3, 4, 5

[59] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *SIGKDD*, 2016. 1, 3

[60] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI*, 2018. 3

[61] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 2020. 2

[62] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics*, 2016. 2

[63] Rosália G Schneider and Tinne Tuytelaars. Sketch classification and classification-driven analysis using fisher vectors. *ACM Transactions on Graphics*, 2014. 2

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

[65] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020. 2

[66] Xing Shi, Inkit Padhi, and Kevin Knight. Does string-based neural MT learn source syntax? In *EMNLP*, 2016. 2

[67] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 2016. 1

[68] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classi-

fication models and saliency maps. In *ICLR Workshop*, 2014. 2

[69] Jifei Song, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning to sketch with shortcut cycle consistency. In *CVPR*, 2018. 3

[70] Suraj Srinivas and Francois Fleuret. Full-gradient representation for neural network visualization. In *NeurIPS*, 2019. 2

[71] Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. Linguistically-informed self-attention for semantic role labeling. In *EMNLP*, 2018. 2

[72] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 3

[73] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (XAI): Toward medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 1

[74] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008. 5

[75] Michael Van Lent, William Fisher, and Michael Mancuso. An explainable artificial intelligence system for small-unit tactical behavior. In *NCAI*, 2004. 1

[76] Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. Context-aware neural machine translation learns anaphora resolution. In *ACL*, 2018. 2

[77] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 2017. 3, 4

[78] Alvin Wan, Lisa Dunlap, Daniel Ho, Jihan Yin, Scott Lee, Henry Jin, Suzanne Petryk, Sarah Adel Bargal, and Joseph E Gonzalez. Nbdt: neural-backed decision trees. In *ICLR*, 2021. 3

[79] Alexander Wang, Mengye Ren, and Richard Zemel. Sketchembednet: Learning novel concepts by imitating drawings. In *ICML*, 2021. 2

[80] Adina Williams, Samuel R Bowman, et al. Do latent tree learning models identify meaningful structure in sentences? *Transactions of the Association for Computational Linguistics*, 2018. 2

[81] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 4

[82] 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. 4, 5

[83] Lan Yang, Kaiyue Pang, Honggang Zhang, and Yi-Zhe Song. Sketchaa: Abstract representation for abstract sketches. In *ICCV*, 2021. 3, 4, 5

[84] Lan Yang, Kaiyue Pang, Honggang Zhang, and Yi-Zhe Song. Finding badly drawn bunnies. In *CVPR*, 2022. 3

[85] Qian Yu, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy Hospedales. Sketch-a-net that beats humans. In *BMVC*, 2015. 4, 5

[86] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 2, 3

[87] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. *NeurIPS*, 2019. 8

[88] Yongfeng Zhang, Xu Chen, et al. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 2020. 2

[89] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 1, 3