International Journal of Computer Vision (2022) 130:2006–2021
https://doi.org/10.1007/s11263-022-01623-7

Generative Sketch Healing

Yonggang Qi[1,2] · Guoyao Su[1] · Qiang Wang[1] · Jie Yang[1] · Kaiyue Pang[2] · Yi-Zhe Song[2]

Received: 20 September 2021 / Accepted: 27 April 2022 / Published online: 7 June 2022

Abstract
To perceive and create a whole from parts is a prime trait of the human visual system. In this paper, we teach machines to perform a similar task by recreating a vectorised human sketch from its incomplete parts, dubbed as *sketch healing*. This is fundamentally different to prior works on image completion since (i) sketches exhibit a severe lack of visual cues and are of a sequential nature, and more importantly (ii) we ask for an agent that does not just fill in a missing part, but to *recreate* a *novel* sketch that closely resembles the partial input *from scratch*. We identify two key facets of sketch healing that are fundamental for effective learning. The first is encoding the incomplete sketches in a graph model that leverages the sequential nature of sketches to associate key visual parts centred around stroke junctions. The intuition is then that message passing within the graph topology will naturally provide the healing power when it comes to missing parts (nodes and edges). Second we show healing is a trade-off process between global semantic preservation and local structure reconstruction, and that it can only be effectively solved when both are taken into account and optimised together. Both qualitative and quantitative results suggest that the proposed method significantly outperforms the state-of-the-art alternatives on sketch healing. Last but not least, we show that sketch healing can be re-purposed to support the interesting application of sketch-based creativity assistant, which aims at generating a novel sketch from two partial sketches even without specifically trained so.

Keywords Sketch healing · Graph convolutional networks · Perceptual distance · Sketch generative model · Sketch representation

1 Introduction

The human visual system is so remarkable in its ability to reasoning. One of its important tricks is to perceive a whole by reasoning on parts—the Kanizsa triangle (Tallon et al., 1995) shown in Fig. 1a being a famous example. This had partially motivated a recent line of research on image completion (Pathak et al., 2016; Yang et al., 2017; Song et al., 2018c; Xiong et al., 2019; Yu et al., 2018; Sagong et al., 2019; Zheng et al., 2019) which aims at hallucinating missing pixels given contextual regions. Great strides have been made to date, with algorithms able to produce highly plausible filler patches. Such successes are however largely down to the data-driven nature of these algorithms, without much insight given towards reasoning.

Others have fixated on human sketches as a medium to gain insight into the human visual system—to see is to sketch. That is, the sketching process to a large extent reflects the visual perception of an object. This has triggered a large body of research on human sketch understanding, some more application oriented (Cao et al., 2010; Eitz et al., 2012; Berger et al., 2013; Sangkloy et al., 2016; Yu et al., 2017; Li et al., 2018; Shen et al., 2018; Simo-Serra et al., 2018; Pang et al., 2019; Bhunia et al., 2020b; Wang et al., 2021), others starting to tackle insightful problems such as sketch synthesis (Song et al., 2018a; Ge et al., 2020), sketch abstraction (Riaz Muhammad et al., 2018; Pang et al., 2018; Bhunia et al., 2020a) and sketch completion (Liu et al., 2019). Sketch is commonly perceived to be a more challenging visual modality compared with photo, because (i) it lacks visual features, (ii) it is abstract and iconic, and (iii) it is sequential in nature.

In this paper, we are also interested in studying sketches. In particular, we would like to use sketches to understand the visual reasoning problem of *devising the whole from parts*

Communicated by William Smith.

✉ Yonggang Qi
qiyg@bupt.edu.cn

[1] School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China

[2] SketchX Lab, CVSSP, University of Surrey, Guildford, UK

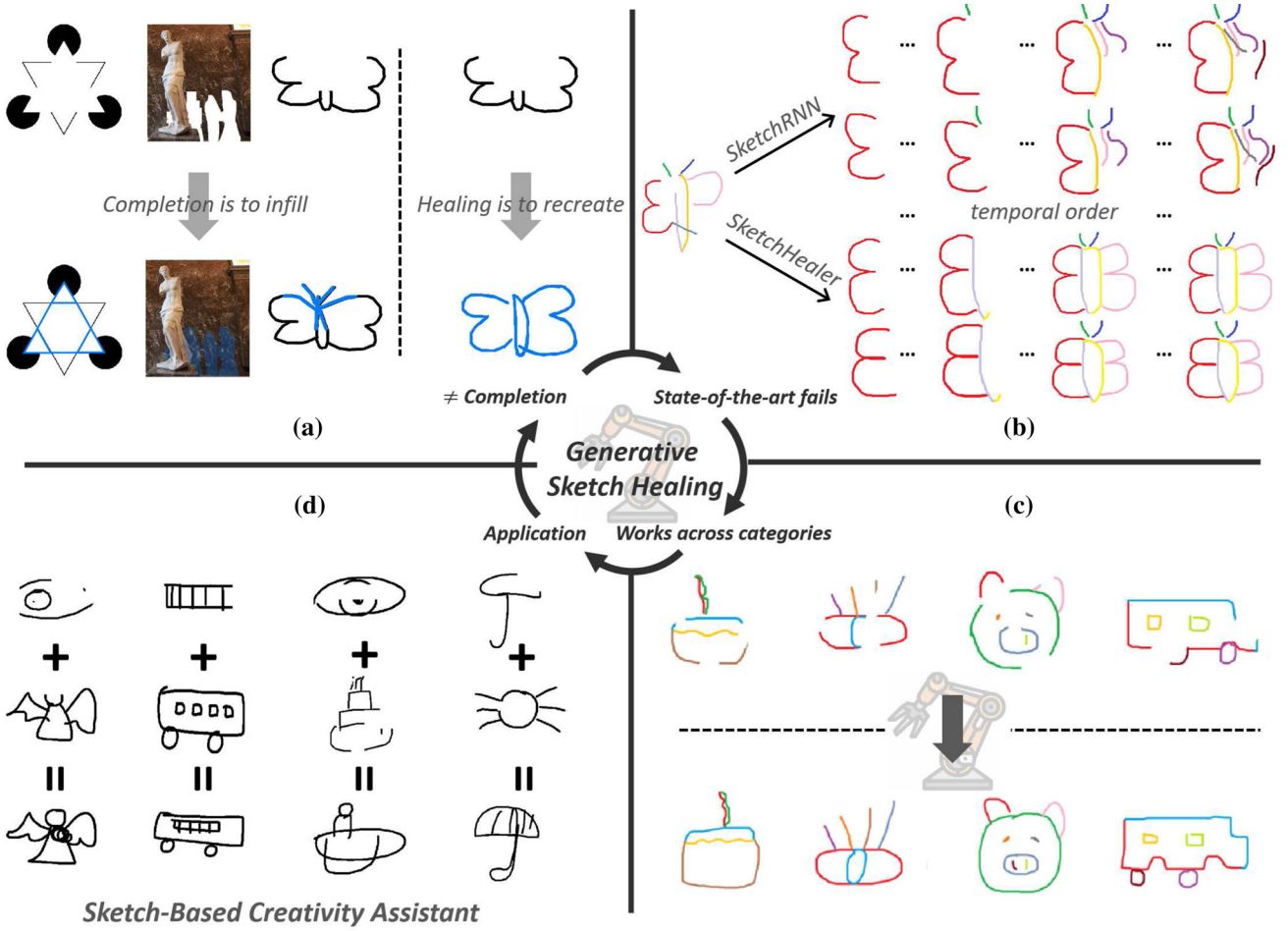

Fig. 1 **a** Visual completion is different to healing that performs rendering on missing parts only. **b** Sketch healing aims to recreate a novel visual imagery *from scratch* that closely resembles the partial sketch input, and proceeds in a *temporal* fashion. SketchRNN fails completely on sketch healing. **c** Exemplary result on how partial human sketches can be successfully recreated by our proposed SketchHealer across multiple categories. **d** One application of SketchHealer is a creativity assistant that generates a novel sketch with creative visual semantics from two partial sketch inputs

(albeit only to a superficial level). We differ significantly to the conventional problem of image completion. First, we do not treat sketches as pixelated photos, but as a sequence of strokes (represented in vector format) that reflects the actual drawing process. Furthermore, we require ourselves to generate a *novel* and *complete* sketch stroke-by-stroke that best resembles the partial input, other than just filling in the missing parts. Together, these constraints deviate us from image completion, and move towards a new problem which we call *sketch healing*.

On the outset, sketch healing is akin to the well studied problem of vector sketch synthesis. The pioneering work of SketchRNN (Ha & Eck, 2018), for example, already has the ability to generate realistic human-like sketch drawings at stroke-level, either from a random vector or conditioned on a partial sketch encoding. In hindsight though, we are completely different. We are not sketching towards a *recognisable* concept, but a complete sketch that closely *resembles* the partial input. For example, conditioned on the encoding of a partial butterfly sketch, SketchRNN is interested in sketching a *plausible* butterfly; we on the other hand are focusing on reproducing a complete version of the partial sketch, regardless of knowing whether it is a butterfly (Fig. 1b). We further insist on solving for random droppings of sketch parts as well, where more than one "hole" appear anywhere on a sketch. This setting is incompatible with the "completion mode" of SketchRNN, which dictates a strict sequential ordering, i.e., the synthesised sketch must be *on top of* of existing input strokes.

Solving the sketch healing task is non-trivial. It requires a sketch-specific representation that not only accommodates the unique traits of sketches (abstract and sequential), but also robust enough towards various levels of missing parts.

This is made even more challenging since we are after a more generic healer that works over *multiple categories*, other than training a single model per category.

We make two key modifications to existing sketch generative models (Ha & Eck, 2018; Chen et al., 2017; Song et al., 2018a; Cao et al., 2019; Bhunia et al., 2020a) for the effective learning of this novel task: (i) As opposed to the common dichotomy of formulating sketch data into raster pixels or sequential points, we combine the best of two worlds by proposing to represent sketches as a graph structure that encodes both visual and temporal sketch traits. Specifically, we introduce a sketch graph construction module that organises key sketch parts in accordance with the order of drawing. We select representative stroke points as graph nodes to capture the most visual information and form the edge links via an adjacency matrix based on temporal proximity. Such seemingly simple change to sketch data organisation, however, proves to be critical: the learned graph offers part-oriented and structure-aware sketch representation that is robust to node removal, and at testing, the inherent node message passing mechanism inside graph model works naturally to fill in the missing gaps. (ii) Complementary to the traditional supervision of *local* per-point reconstruction, we introduce perceptual loss (Zhang et al., 2018), an easy-to-compute, label-free metric aiming at characterising visual appearance similarity from a *global semantic* perspective. This is based on a closer inspection of the problem, which resulted in the insight that generative healing follows a conditional multi-modal modelling process (i.e., one-to-many), and can not be sufficiently solved by faithful restoration to one particular ground-truth target (i.e., one-to-one). Perceptual metric then comes to the rescue by endorsing the validity of multiple reasonable targets and consequently loosening the reconstruction constraint and reducing overfitting on training data. We combine two distance metrics as two separate loss functions optimised in an end-to-end manner and show that the best performance is obtained by balancing the weights between them dynamically according to the various corruption levels of input sketches.

Overall, our framework, termed as SketchHealer,[1] is a graph-to-sequence network with a graph convolutional encoder that embeds a partial sketch graph into a latent vector, appended by a LSTM decoder to output a vectorised healed sketch stroke-by-stroke. Experiments show that SketchHealer performs reasonably well on the sketch healing task, in a stark contrast to the complete failure mode observed in the current representative sketch generative model, SketchRNN (Fig. 1c).
Our contributions are summarised as follows:

- We propose the problem of sketch healing, as an interesting yet changeling alternative to conventional sketch synthesis.
- We propose SketchHealer, a novel graph-to-sequence network that identifies and encapsulates two key aspects of healing-specific design.
- We evaluate SketchHealer on 17 categories selected from QuickDraw (Ha & Eck, 2018) dataset and validate its superiority over three contemporary vector sketch synthesis baselines both qualitatively and quantitatively.
- We showcase one practical application of SketchHealer as a potential creativity assistant for free-hand drawing (Fig. 1d).

2 Related Work

Vector Sketch Generation Much progress (Chen & Koltun, 2017; Johnson et al., 2016; Zhang et al., 2017a; Karras et al., 2018, 2019; Miyato et al., 2018; Johnson et al., 2018; Pumarola et al., 2018; Chan et al., 2019; Zhou et al., 2019) has been made on image generation tasks both in the supervised (Isola et al., 2017) and unsupervised settings (Zhu et al., 2017a; Kim et al., 2017; Yi et al., 2017). Given a cat image, we can now translate to other category of animals (Zhu et al., 2017b), render it in Monet style (Li et al., 2017) or even make it 3D animated (Shih et al., 2020). This is in stark contrast with the very few existing works that focus on vector image generation, where its temporal and spatial nature bring more challenges. The seminal work of Graves (2013) proposed a sequence-to-sequence model and for the first time achieved realistic vector handwritten digits generation in a wide variety of styles. With the availability of large-scale crowdsourced sketch datasets, this model was then adapted in Ha and Eck (2018), Chen et al. (2017) and Das et al. (2020) and achieved both unconditional and conditional vector sketch-to-sketch synthesis. Vector sketch generation was also extended beyond a single domain. Song et al. (2018a) proposed the first deep stroke-level photo-to-sketch synthesis model. To cope with the intrinsic noisy supervision of photo-sketch pairs, they addressed the limitations of cross-domain image translation models based on multi-task supervised and unsupervised hybrid learning. In this paper, we study a different problem—sketch healing, that takes a partial sketch as input and output *novel* sketch that closely resembles the input, while others focus on sketch synthesis (i.e., to sketch a recognisable rendition) (Ha & Eck, 2018; Chen et al., 2017;

[1] An earlier and preliminary version of this work was published in (Su et al., 2020). Compared with (Su et al., 2020) apart from more extensive experiments and analysis, this work differs in proposing a new healing-specific distance metric from a global perceptual view, which brings significant improvements in model performance. For ease of reading, we denote our framework as SketchHealer throughout, and only distinguish between the two works with a specific version number (1.0 vs. 2.0) when comparison is under way (mostly in the experiment section).

Das et al., 2020), and photo-sketch synthesis (Song et al., 2018a).

Graphical Sketch Representation Graph convolutional network (GCN) (Gori et al., 2005; Kipf & Welling, 2017) was proposed to extend deep neural networks to data with graph structures. By applying GCN-based models, state-of-the-art performance has been achieved over a range of vision tasks, such as image classification (Chen et al., 2019), image captioning (Yao et al., 2018), scene understanding (Yang et al., 2018) and 3D mesh deformation (Ranjan et al., 2018; Wang et al., 2018a). The sequential and sparse nature of sketch makes it an ideal data domain for graphical representation. But only until very recently, GCN-based sketch visual learning were attempted in Xu et al. (2019) and Yang et al. (2021b) for the problem of sketch recognition and segmentation respectively. They both constructed their graph nodes based on the absolute coordinates of the sampled sketch points and transformed them via multi-layer perceptrons and appropriate pooling methods. In contrast, our proposed SketchHealer uniquely embeds the temporal drawing order to build adjacency matrices.

Image Completion Image completion or also known as inpainting is to synthesise visual contents conforming to a plausible hypothesis in a missing or damaged region. There are two broad lines of work aiming to tackle this task: exemplary-based methods (Efros & Leung, 1999; Barnes et al., 2009; Wilczkowiak et al., 2005) searched and pasted visual patches from other known regions in the gallery. These algorithms worked well for stationary images (e.g., textures) but could lead to complete failure on non-stationary natural images. Deep generative CNN-based methods (Pathak et al., 2016; Song et al., 2018c; Xiong et al., 2019; Yu et al., 2018; Sagong et al., 2019; Lahiri et al., 2020) directly generated pixels inside the missing patch based on the semantics learned from large-scale dataset in an end-to-end fashion. This usually involved an encoder-decoder paradigm with various key modifications devised—including partial (Liu et al., 2018) and gated (Yu et al., 2019) convolutions, use of contextual attention modules (Song et al., 2018b) and adversarial discriminators (Iizuka et al., 2017). User guidance was also explored to improve inpainting results including edge lines (Sangkloy et al., 2017), semantic label maps (Wang et al., 2018b) and colour palettes (Zhang et al., 2017b). To our best knowledge, the only inpainting work on sketches was (Liu et al., 2019), which devised a cascade network to refine the completions in an iterative manner. SketchHealer is fundamentally different in that we not only tackle vector sketches, but also in the very nature of the problem itself—rather than just filling the holes, it heals a corrupted incomplete sketch by generating a novel full counterpart.

3 Methodology

Our goal is to recreate a vector sketch S from its partial version $\hat{S}$. A sketch S is a set of points denoted as $(s_1, s_2, \ldots, s_n)$, where each segment s_i is constructed between two consecutive points as a 5 dimensional vector $(\Delta x, \Delta y, ps_1, ps_2, ps_3)$. $(\Delta x, \Delta y)$ is the offset distance in the x and y directions of the pen from the previous point. (ps_1, ps_2, ps_3) is a one-hot vector describing current pen states, where $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$ denote touching, lifting and the end of sketch drawing respectively. $\hat{S}$ is then obtained from S by randomly removing a proportion of n points. Under these notations, our proposed framework, SketchHealer, is formulated as a graph-to-sequence network with a GCN-based encoder mapping $\hat{S}$ of its graphical form $G = (V, E)$ to a latent vector $z \in \mathcal{R}^{d_{model}}$. z is then leveraged as input to sequentially sample an output sketch S^* aiming to best resemble the original full sketch S in a LSTM decoder with output space parameterised as a Gaussian mixture model (GMM). A schematic illustration is shown in Fig. 2.

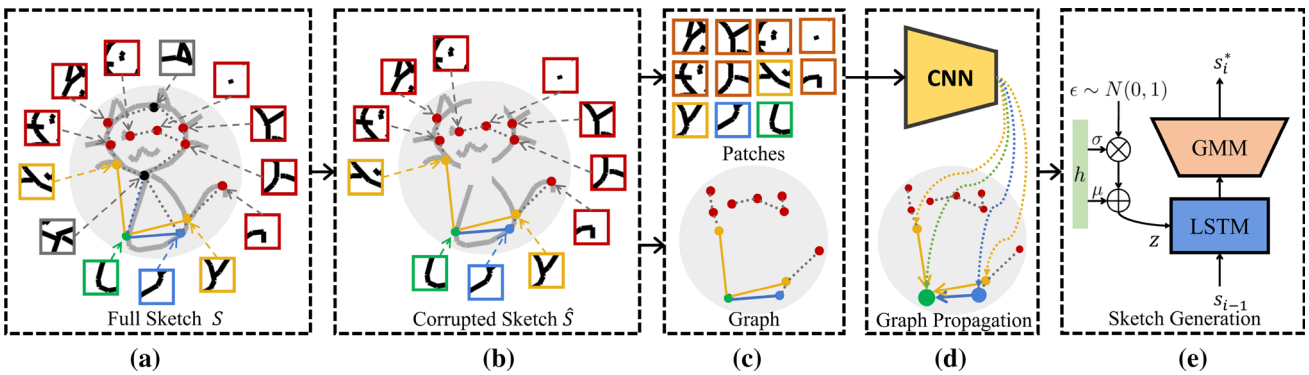

Fig. 2 A schematic illustration of SketchHealer. **a** A full sketch S in its graph form. For each graph node v_i, a visual patch is cropped out as p_i. **b** A corrupted partial sketch $\hat{S}$ in its graph form by masking out a fractional of nodes from S. The associated edge links are removed as well. **c** Model input: graph G and the visual patches P for $\hat{S}$. **d** GCN-based encoder. A graph node (green) will attend to its nearest neighbourhood (blue) and the second nearest (yellow) through graph propagation. **e** LSTM decoder. More details in text (Color figure online)

3.1 Sketch as Graph Representation

Graph Nodes V We consider two types of points as representative graph nodes: (i) the starting point of each stroke, which determines the main structural layout; (ii) internal points sparsely sampled within each single stroke in order to capture a rough path trace. We sample one graph node in every four points in a stroke throughout this paper. Consequently, a set of graph nodes $V = (v_1, v_2, \ldots, v_m)$ is a subset of S. While being more compact, V still preserve the key geometric landmarks of an input sketch.

Graph Edges E Temporal-based nearest neighbour strategy is used to construct the edge links between nodes. That is, for each node v_i, we will connect it with the graph nodes nearby in accordance to their drawing orders in the original sequence of stroke points. We link v_i with its four nearest graph nodes, two prior to its rendering as parent nodes, and two after its presence as child nodes. An adjacency matrix $\mathbf{A} \in \mathcal{R}^{m \times m}$ can then be formed, where each entry a_{ij} represents the link strength between nodes v_i and v_j. We empirically found $a_{ij} = 0.3$ for edge link between two nearest nodes and $a_{ij} = 0.2$ for its linkage to a second nearest node to work well. a_{ij} is zero-valued to indicate no inter-node connections and for self-connection a_{ii}, we simply take its value to 0.5 for regularisation purpose.

Visual Patch P To assign each graph node v_i in $V = (v_1, v_2, \ldots, v_m)$ with its associated visual cue, a local image patch centred around each node is acquired. Specifically, we first render out a raster sketch image of size 640×640 from its vector format and crop a square visual patch p_i of size 128×128 based on the normalised coordinate of v_i. The relatively large patch size is to make sure enough informative visual cues are still captured given the sparse nature of human sketches. A set of node-driven patches $P = (p_1, p_2, \ldots, p_m)$ is thus obtained.

From Full S to Partial $\hat{S}$ To form a graph for partial sketch as final input, we randomly remove a fraction of graph nodes by a probability of p_{mask} and cut the connections in the resulting edge links. The corresponding image patch p_i in P will also become void.

3.2 SketchHealer: A Graph-to-Sequence Network

GCN-based Encoder Our proposed SketchHealer encoder consists of six convolutional layers with kernel size 2×2 followed by max pooling and batch normalisation. By feeding each image patch p_i into the encoder, we produce a visual feature vector $f_{v_i} \in \mathcal{R}^d$ for each node v_i. Then feature propagation is executed to form an updated node feature $u_{v_i} \in \mathcal{R}^d$, where a node v_i attends to all its linked neighbours defined in the adjacency matrix A. Such a spatial-dependent approach is natural to provide a healing effect for the absence of certain parts and enables more robust representation. Formally, we formulate this as follows:

$$u_{v_i} = \sum_{j=1}^{m} a_{ij} f_{v_j} \tag{1}$$

We then integrate all node features into a single vector $h \in \mathcal{R}^{d_{model}}$ for representing the partial sketch $\hat{S}$:

$$\begin{aligned} h &= W \odot G \\ W &= (w_1, w_2, \ldots, w_m) \\ G &= [g(u_{v_1}), g(u_{v_2}), \ldots, g(u_{v_m})] \end{aligned} \tag{2}$$

where $\odot$ denotes dot product, m is set as the maximum number of nodes among all training sketches ($m = 25$ in our case), $g : \mathcal{R}^d \rightarrow \mathcal{R}^{d_{model}}$ is a multilayer perceptron (MLP) unit, W is a learnable weight vector to linearly combine the MLP-produced node vectors G. To introduce generative components, h is further projected into two vectors, $\mu \in \mathcal{R}^{d_{model}}$ and $\sigma \in \mathcal{R}^{d_{model}}$, along with a vector of IID Gaussian variables $\mathcal{N}(0, 1)$ of size d_{model}, to construct the final latent vector z:

$$\begin{aligned} z &= \mu + \sigma \odot \mathcal{N}(0, 1) \\ \mu &= W_\mu h, \ \sigma = exp\left(\frac{W_\sigma h}{2}\right) \end{aligned} \tag{3}$$

LSTM Decoder Taking latent vector z as condition, a LSTM decoder is used to sequentially sample the point offset between the current and the last output sketch strokes. Concretely, the previous stroke ending point s_{i-1} together with the latent vector z are formed as input at each time step, i.e., $x_i = [s_{i-1}; z]$, with the next hidden state given by:

$$[h_i; c_i] = LSTM_{forward}(x_i, [h_{i-1}; c_{i-1}]) \tag{4}$$

We then define the per-step output as $y_i = w_y h_i + b_y \in \mathbb{R}^{6M+3}$, which can be unpacked into a set of parameters:

$$\begin{aligned} y_i = &[(\Pi, \mu_x, \mu_y, \delta_x, \delta_y, \rho_{xy})_1, \ldots, \\ &(\Pi, \mu_x, \mu_y, \delta_x, \delta_y, \rho_{xy})_M, (q_1, q_2, q_3)] \end{aligned} \tag{5}$$

The first M sets of parameters are used to form a Gaussian mixture model (GMM) with M Gaussian components for planar coordinate modelling. $\mu_x, \mu_y, \delta_x, \delta_y, \rho_{xy}$ are the respective means, deviations and covariance coefficients that uniquely determines a bivariate normal distribution. We can now finally represent the per-step point offset $(\Delta x, \Delta y)$ as:

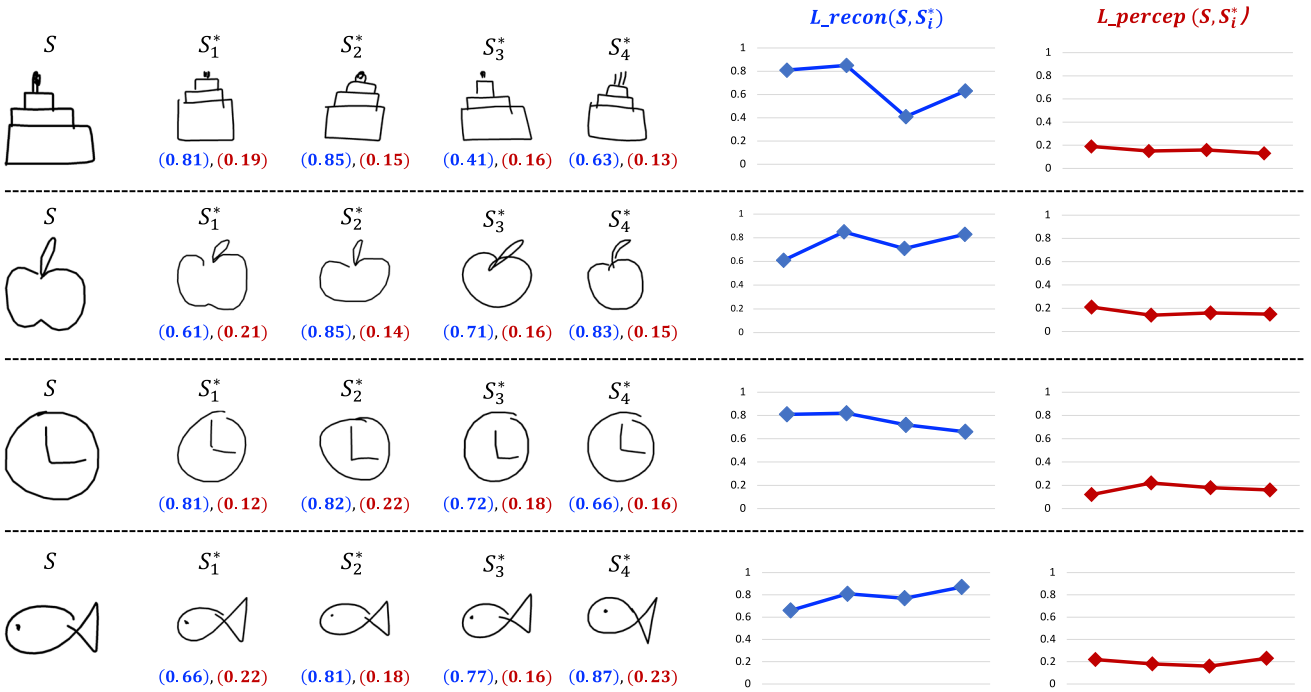

Fig. 3 Comparison of two distance metrics L_{recon} and L_{percep}. Given two similar sketches S and S_i^*, L_{recon} fails to reflect the visual semantic similarity that deems sensible in the human eyes. Contrarily, L_{percep} originated from a global perspective has robustly and successfully detected such visual signal, as manifested in the consistent measurements of close distance among multiple perceptually-aligned sketch pairs (S, S_i^*). We argue and verify that L_{percep} is critical for sketch healing of a multi-modal nature

$$
\begin{aligned}
p(\Delta x, \Delta y) &= \sum_{j=1}^{M} \Pi_j \mathcal{N}(\Delta x, \Delta y | \Phi_j) \\
\Phi_j &= \left(\mu_{x,j}, \mu_{y,j}, \delta_{x,j}, \delta_{y,j}, \rho_{xy,j}\right), \\
\sum_{j=1}^{M} \Pi_j &= 1
\end{aligned} \tag{6}
$$

The last three parameters (q_1, q_2, q_3) in Eq. (5) follow a categorical distribution, which is used to estimate the ternary pen state (ps_1, ps_2, ps_3) defined earlier.

$$
ps_k = \frac{\exp(q_k)}{\sum_{j=1}^{3} \exp(q_j)}, \quad k = 1, 2, 3 \tag{7}
$$

Please refer to Ha and Eck (2018) for more details.

3.3 Learning Objective: A Local and Global Tradeoff

Healing is Multi-Modal It is intuitive that given $\hat{S}$ and the generative output S^* produced by the SketchHealer in Sect. 3.2, the goal of optimisation is to ensure S^* as close as possible to its original uncorrupted sketch S. This corresponds to the per-point reconstruction loss as enjoyed by most existing sketch generative works[2] (Ha & Eck, 2018; Chen et al., 2017; Song et al., 2018a; Cao et al., 2019; Bhunia et al., 2020a):

$$
L_{recon} = -E_{q_\phi(z|\hat{S})}[\log p_\theta(S|z)] \tag{8}
$$

where $q_\phi(z|\hat{S})$ is the posterior probability of the generated data points, and $p_\theta(S|z)$ the target data distribution. Such practice of regression to one specific target on every local front, however, is intrinsically flawed for sufficient modelling of sketch healing in hindsight: healing is multi-modal in nature. A partial sketch input can correspond to many possible synthetic results that all have been reasonably healed—complete, smooth visual imagery with easily recognisable links to the partial input. Putting formally, this suggests a comparative metric $\mathcal{M}$ that supports multiple S^*s to be simultaneously close to S, i.e., $\mathcal{M}(S_1^*) \approx \mathcal{M}(S), \mathcal{M}(S_2^*) \approx \mathcal{M}(S), \mathcal{M}(S_3^*) \approx \mathcal{M}(S)$, ..., other than *only one* S^* that exactly reconstructs S as prescribed in Eq. 8. The effect of an expected $\mathcal{M}$ (also the one we adopted throughout the paper) is exemplified in Fig. 3. We can see that

[2] We follow Chen et al. (2017) to remove the KL-divergence term between $q_\phi(z|\hat{S})$ and $p(z)$ commonly observed in the formulation of a VAE-like model (Kingma & Welling, 2013), which is shown to benefit multi-class generations.

compared with the metric of self-reconstruction, $\mathcal{M}$ does not struggle with local line deformations and offsets (if not too much) and is acting globally with emphasis on perceptual visual similarity. And it's exactly this globally perceptual view that makes possible of relieving the once one-to-one generation constraint in Eq. 8 to a one-to-many counterpart that comes in great fit for the healing task. We detail our choice of $\mathcal{M}$ as follows.

Perceptual metric $\mathcal{M}$ The ability to compare visual items is perhaps the most fundamental problem in computer vision. Recent literature (Zhang et al., 2018; Blau & Michaeli, 2019; Tariq et al., 2020; Amir & Weiss, 2021) has consistently corroborated the finding of the unreasonable effectiveness of deep features as a good perceptual metric on visual similarity. We follow these advances to define a perceptual loss that measures the visual similarity between sketch pairs. As illustrated in Fig. 4, we first pre-train a SqueezeNet (Iandola et al., 2017) on Berkeley-Adobe Perceptual Patch Similarity (BAPPS) dataset (Zhang et al., 2018) as our fixed deep perceptual feature extractor $pf(\cdot)$. Since SqueezeNet only admits raster pixel input, we introduce a rasterisation module that renders S^*, S into their corresponding binary raster images S_I^*, S_I with nearest neighbour spatial interpolation. By feeding both S_I^* and S_I into $pf(\cdot)$, we then extract their deep representations from L different layers and denote them as $\{pf^l(S_I^*) :\Leftrightarrow pf^{l*}, pf^l(S_I) :\Leftrightarrow pf^l\} \in \mathbb{R}^{H_l, W_l, C_l}$. The perceptual loss is finally computed as the mean element-wise l_2 distance between the feature maps of the healed and ground-truth sketches:

$$L_{percep} = \sum_{l=1}^{L} \frac{1}{H_l W_l} \sum_{h,w} ||w_l \odot (pf_{hw}^{l*} - pf_{hw}^{l})||_2^2 \tag{9}$$

where $w_l \in \mathbb{R}^{C_l}$ is adopted to scale the feature activations channel-wise (Zhang et al., 2018).

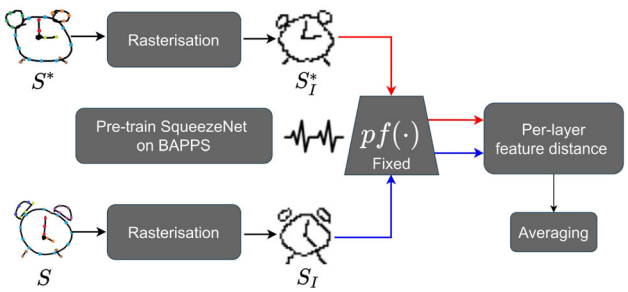

Fig. 4 Schematic illustration of the proposed perceptual loss between the generated sketch S^* and its reference sketch S. Given their original vector format, S and S^* are first rasterised as S_I and S_I^*, respectively. Next, a pre-trained CNN $pf(\cdot)$ optimised to perform image perceptual similarity is applied to obtain their feature maps, which are used to calculate the perceptual similarity. More details in text

Formulation Our final formulation is an adaptively weighted combination of two losses, L_{recon} and L_{percep}. The idea is that when the input sketch corruption level is low, the self-reconstruction loss plays a bigger role for more accurate local renderings. On the contrary, the perceptual loss should provide a stronger supervision signal of global control to better bridge the healing gap under insufficient visual cues. Denoting the input corruption level as p_{mask}, we define our optimisation objective as:

$$L_{total} = (1 - p_{mask})L_{recon} + p_{mask}L_{percep} \tag{10}$$

3.4 Model Deployment

Once trained, it is straightforward to apply the SketchHealer. Given a latent vector z encoded from a corrupted sketch input, we feed it together with a manually-defined starting point ($\Delta x = 0, \Delta y = 0, ps_1 = 1, ps_2 = 0, ps_3 = 0$) into the LSTM decoder. The generated data point will be fed again with z to produce the next data point in a recurrent manner, until the stop signal is reached, i.e., ($ps_1 = 0, ps_2 = 0, ps_3 = 1$).

4 Experiment

4.1 Experimental Setting

Dataset We evaluate our proposed model on *QuickDraw* (Ha & Eck, 2018), which is the largest human sketch drawing dataset to date. It provides over 50 million vector sketches across 345 object categories, where we select a subset for our experiments. In particular, the 17 categories[3] we choose generally respect the following rules: (i) both complex and simple drawings are included, e.g., `angel` and `belt`; (ii) instances inside categories exhibit similar global appearances but only differ in very local subtle details, e.g., `cat` and `pig`; (iii) common life object category contains diverse sub-category variations, e.g., `bus`, `umbrella` and `clock`. For each class, 70,000 sketches are used for training and 2,500 are randomly selected for testing.

Competitors To date, there are six conditional vector sketch generation frameworks publicly available, including SketchRNN (Ha & Eck, 2018), SketchPix2seq (Chen et al., 2017), MGT (Xu et al., 2019), SketchLattice (Qi et al., 2021), SketchAA (Yang et al., 2021a) and RPCL-pix2seq (Zang et al., 2021). We evaluate them all for baseline comparisons.

[3] `airplane, alarm_clock, angel, apple, belt, bus, butterfly, cake, cat, clock, eye, fish, pig, sheep, spider, umbrella, The Great Wall.`

- SketchRNN (Ha & Eck, 2018) is a sequence-to-sequence model that takes in the offset distance between consecutive points as temporal input. For fair comparison, we retrain the model without KL-divergence term, which is shown to be beneficial for multi-class scenario.
- SketchPix2seq (Chen et al., 2017) differs itself from SketchRNN in the proposed convolutional encoder, which scraps the vector representation of sketches and accepts raster sketch image input instead, with the hope of better visual learning via CNNs.
- Built upon Transformer (Vaswani et al., 2017), MGT (Xu et al., 2019) additionally injects graph learning into the framework to explicitly capture the stroke structural geometry.
- Similarly to SketchPix2eq, SketchLattice (Qi et al., 2021) takes raster sketch image as input for vectorised sketch generation. It, however, enables the preservation of structural cues unique in vector sketch representation by sampling a set of points from the pixelative format of the sketch using a lattice graph.
- SketchAA (Yang et al., 2021a) advocates granularity controllable sketch representation by organising visual learning in a coarse-to-fine hierarchy. Originally designed for sketch classification, the network can be further adapted with a LSTM decoder for sketch healing task.
- RPCL-pix2seq (Zang et al., 2021) is a GMM-based sketch generative model, which leverages Rival Penalised Competitive Learning (RPCL) to discover an optimal number of instance-specific Gaussian components, thus achieving higher generation quality.

We note that these competitors were not specifically designed for sketch healing, yet they still represent the closest alternatives and are procedural-wise compatible once re-purposed.[4] To comply with the different sketch data format required by different competitors, we also process corrupted sketches into both point sequence and raster image forms. Specifically, given a list of strokes to be masked, we simply discard them from the point sequence and do not proceed with extra padding operation. We then render an image version from the resulting sequence of the corrupted sketches for methods working with raster pixels. We also compare with SketchHealer-1.0, which is the earlier version (Su et al., 2020) of this work, which removes L_{percep} with only L_{recons} included in optimisation.

Evaluation protocol Apart from qualitative comparisons, we design two metrics to allow quantitative evaluations. Our measures specifically answer two questions: (i) How recognisable are the vector sketches generated by different competitors? (ii) What are the human preferences among the healed sketches generated by different competitors? A good score for the former requires the sketches healed from their partial parts to be realistic and diverse, while the latter directly involves human as judge and also as a way to confirm any conclusions obtained from the former—generative models are notoriously hard to be fairly evaluated by heuristically-designed discriminative approaches (Theis et al., 2016). More specifically, we formulate two metrics: (i) *sketch recognition accuracy*: we train a multi-category classifier by AlexNet using the all the training split of 345 categories in *QuickDraw* dataset. The classifier is then used to assign a class label to measure how recognisable a generated sketch is. 2500 testing sketches from each of 17 categories are used for this purpose. (ii) *human preference percentage*: We recruit a total of 10 human participants, each of whom is asked to complete 50 independent trials. In each trial, a partial sketch and different generated healed versions in randomised orders are then presented. The participant is expected to make a *single* choice of selecting the best healed sketch based on two criteria: input resemblance and overall visual aesthetics.

Implementation details Our model is implemented on PyTorch (Paszke et al., 2017) with a single Nvidia Tesla T4 GPU. The Adam optimiser is used with the parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The learning rate is set to 10^{-3} with a decay rate of 0.999 every iteration. The proportion of stroke points to mask out during model training is set to be random value in range of $p_{mask} \in [10\%, 30\%]$. We test for different levels of sketch corruption, i.e., p_{mask} can be 50%.

4.2 Comparison with Baselines

Qualitative results We illustrate some examples produced by our method (SketchHealer-2.0) under different values of p_{mask} in Fig. 5. Follow observations can be made: (i) SketchHealer-2.0 is not only able to render a novel sketch just like humans do, but can also faithfully recreate the essential subtle visual elements even when the majority part of specific visual cues are missing in the partial input. For example candle over the cake keeps presented up to $p_{mask} = 70\%$, despite the input sometimes only shows very weak evidence of candle visual signals. (ii) Given one human sketch and different random removals of visual elements on different levels, SketchHealer-2.0 delivers consistent generation results - albeit subtle details are uniquely rendered, global appearances and structures are unanimously kept. (iii) The sensitiveness of our proposed model to different corruption levels of inputs vary across object categories. But overall, the model performs reasonably well when $p_{mask} \leq 50\%$. We further qualitatively compare with generative healing results between six competitors in Fig. 6. Even under a cor-

[4] Suggested optimal parameter settings in the original papers are adopted for all competitors.

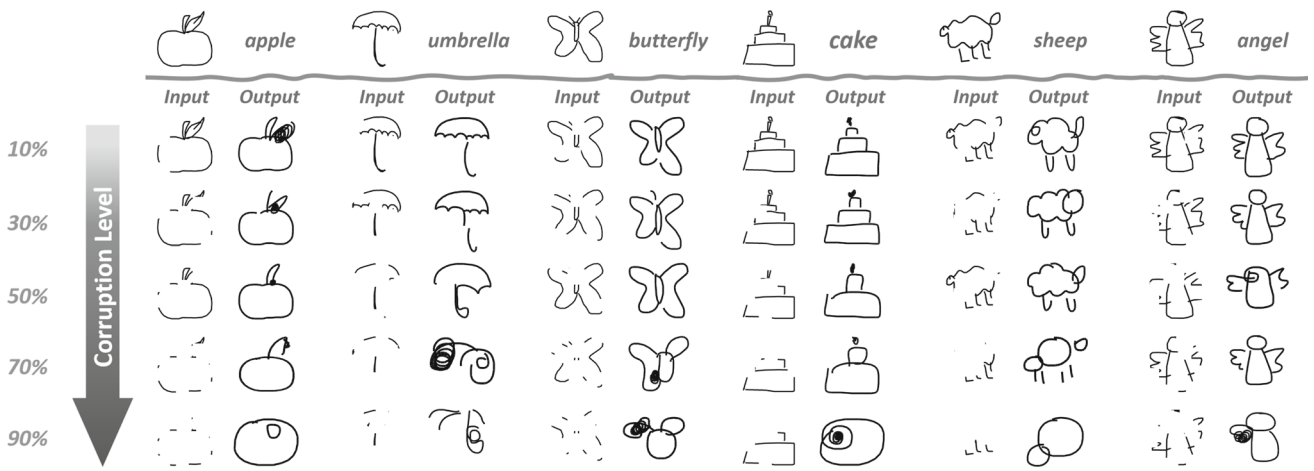

Fig. 5 Exemplary results of our approach (SketchHealer-2.0) under different corruption values of p_{mask}. We intentionally select categories that encapsulates diverse visual semantics ranging from simple to complex

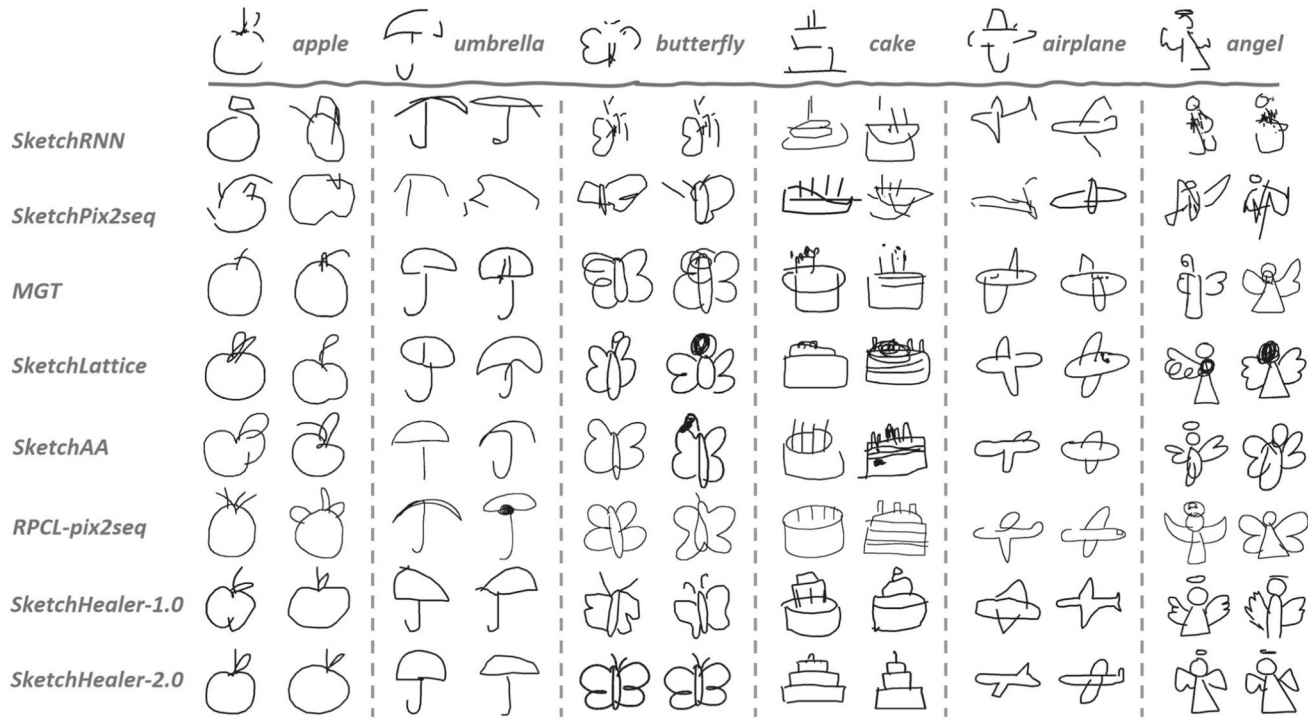

Fig. 6 Qualitative comparisons between the proposed SketchHealer-2.0 and three other contemporary competitors. For all sketch partial inputs, corruption level of $p_{mask} = 10\%$ is applied throughout

ruption level of 10%, SketchRNN and SketchPix2seq fails to recreate a desired vector sketch in most cases. The healed sketches obtained by MGT, SketchLattice, SketchAA and RPCL-pix2seq are clearly more reasonable but somehow struggle to produce clean structures, i.e., many noisy strokes can be observed and these methods are generally inferior in preserving local appearances of the input partial sketches—see how the wings and body of healed angels are mismatched with the corrupted input. And while the earlier version of this work, SketchHealer-1.0, performs considerably better, its gap with SketchHealer-2.0 is significant: See how the butterfly wings and umbrella panel get consistently healed under multiple generative renderings and more closely resemble the partial input.

Quantitative Results We compare the performance of all different competitors under the two metrics (see Sect. 4.1) in Table 1: (i) Under the recognition metric, SketchHealer-2.0 beats all the competitors over all corruption levels. Interestingly, when the uncorrupted full sketches ($p_{mask} = -$) are fed as input, SketchRNN can achieve better results except

Table 1 Quantitative comparisons between competitors on sketch healing

p_{mask}	Metric		SketchRNN	SketchPix2seq	MGT	SketchLattice	SketchAA	RPCL-pix2seq	SH-1.0	SH-2.0
–	Recognition	Top 1	57.98%	22.14%	23.53%	32.91%	21.41%	37.29%	50.64%	**67.98%**
		Top 5	**81.92%**	34.36%	38.18%	46.04%	33.41%	59.35%	70.61%	79.09%
		Top 10	**88.02%**	40.56%	44.35%	51.68%	38.11%	67.82%	76.71%	81.87%
	Human	N/A	5.24%	12.24%	6.00%	12.94%	19.94%	5.12%	8.47%	**36.18%**
10%	Recognition	Top 1	24.41%	21.88%	19.47%	30.83%	23.18%	32.53%	49.77%	**71.32%**
		Top 5	46.23%	31.92%	30.01%	43.59%	32.76%	53.53%	69.92%	**80.01%**
		Top 10	56.28%	36.91%	35.47%	49.53%	37.76%	60.41%	80.01%	**81.63%**
	Human	N/A	7.71%	9.06%	7.23%	10.76%	13.23%	5.25%	6.29%	**40.35%**
30%	Recognition	Top 1	3.14%	9.51%	11.17%	29.34%	22.18%	18.65%	41.59%	**65.27%**
		Top 5	10.25%	16.06%	18.41%	41.36%	33.00%	34.18%	62.76%	**77.38%**
		Top 10	15.91%	20.26%	23.29%	47.42%	38.53%	40.41%	68.12%	**80.27%**
	Human	N/A	12.59%	6.29%	5.82%	7.47%	12.17%	4.00%	5.53%	**46.11%**
50%	Recognition	Top 1	–	–	6.17%	26.42%	20.35%	10.00%	26.32%	**40.62%**
		Top 5	–	–	11.88%	38.01%	29.06%	18.88%	48.94%	**61.98%**
		Top 10	–	–	15.18%	43.83%	34.41%	24.29%	57.89%	**63.92%**
	Human	N/A	–	–	–	–	–	–	–	–

Best performance are highlighted in bold

Recognition results are obtained by classifying generated healed sketches with a pre-trained multi-category sketch classifier. Human represents human's preference of choice among the synthetic outputs by different competitors. We omitted the numbers for SketchRNN and SketchPix2seq at $p_{mask} = 50\%$ as they are around the accuracy of random chances ($\approx 0.3\%$). Human preference between SketchHealer-1.0 (SH-1.0) and SketchHealer-2.0 (SH-2.0) at $p_{mask} = 50\%$ can be found in Table 3

the Top 1 recognition accuracy. It, however, collapses dramatically even when only 10% of stroke points are masked out and a complete failure when the proportion rises to 30%. (ii) It draws attention that our model achieves slightly better Top 1 recognition accuracy conditioned on mild corrupted inputs (71.32% when $p_{mask} = 10\%$) compared with that in full uncorrected inputs (67.98%). This is expected, since SketchHealer-2.0 is trained using corrupted sketch inputs only (p_{mask} ranges from 10% to 30%), it naturally generalises better to corrupted sketch input (in-distribution) than that from uncorrupted source (out-of-distribution). We argue that the slightly worse healing performance on unseen uncorrupted sketch data, on the contrary, confirms the versatility of SketchHealer-2.0. (iii) Under the human metric, our model still outperforms all competitors as indicated by the percentage of human preference of choices. While human subjectivity on interpreting healing quality may vary, there is strong consensus reached over all corruption levels that the competitors are less qualified for sketch healing ($< 14\%$ out of total trials are deemed as better than ours on average). (iv) The clear improvement over SketchHealer-1.0 under both metrics confirms the role of the newly introduced perceptual metric in SketchHealer-2.0. The correlation between the efficacy of perceptual metric and corruption level is also observed, where the former tends to play a greater role as the latter becomes more severe.

4.3 Ablation Study

Impact of graph representation To verify the importance of encoding sketch as a graph in sketch healing, we strip off the generative part to maximally disentangle its impact and ask the question: how recognisable are the latent vectors of corrupted partial sketches (z) encoded by different type of encoders? A better representation is expected to be more category discriminative and robust to corruption level changes. We realise such goal with a quantitative metric from the sketch-to-sketch retrieval task—by examining the performance of retrieving sketches of the same label given a partial sketch query. We form our query set with 500 testing sketches from each of the selected 17 categories under our experimental setting and the rest as gallery.

The results are shown in Table 2. We can find that compared with point (SketchRNN) and pixel (SketchPix2seq, RPCL-pix2seq) based sketch representation, graph-based representation (MGT, SketchLattice and Ours) performs significantly better. Our model achieves the best among all graph-based alternatives and exhibits surprisingly stable behaviours when the corrupted level of sketch input increases.

We also visualise some sketch-to-sketch retrieval results in Fig. 7. Even under mild condition of $p_{mask} = 10\%$, all the compared methods have clearly many more false

Table 2 Sketch-to-sketch retrieval result (%) obtained on encoded z to verify the efficacy of different types of sketch representation

Competitor	p_{mask} (%)	Retrieval (%)				p_{mask} (%)	Retrieval (%)			
		Top 1	Top 3	Top 5	Top 10		Top 1	Top 3	Top 5	Top 10
SketchRNN	10	50.65	69.76	77.60	86.21	30	43.48	63.67	72.03	82.39
SketchPix2seq	10	45.20	68.34	77.54	87.09	30	27.66	51.82	63.52	78.99
MGT	10	62.35	79.71	83.24	86.76	30	50.00	72.06	77.35	83.82
SketchAA	10	42.89	68.02	76.77	86.98	30	37.29	62.06	73.51	85.00
RPCL-pix2seq	10	53.24	71.76	76.76	83.23	30	40.29	66.47	75.29	82.35
SketchLattice	10	63.03	79.15	84.83	90.60	30	58.73	77.56	83.97	90.34
SketchHealer-2.0	10	**85.23**	**93.23**	**95.27**	**97.53**	30	**83.48**	**91.47**	**93.29**	**95.97**

Best performance are highlighted in bold

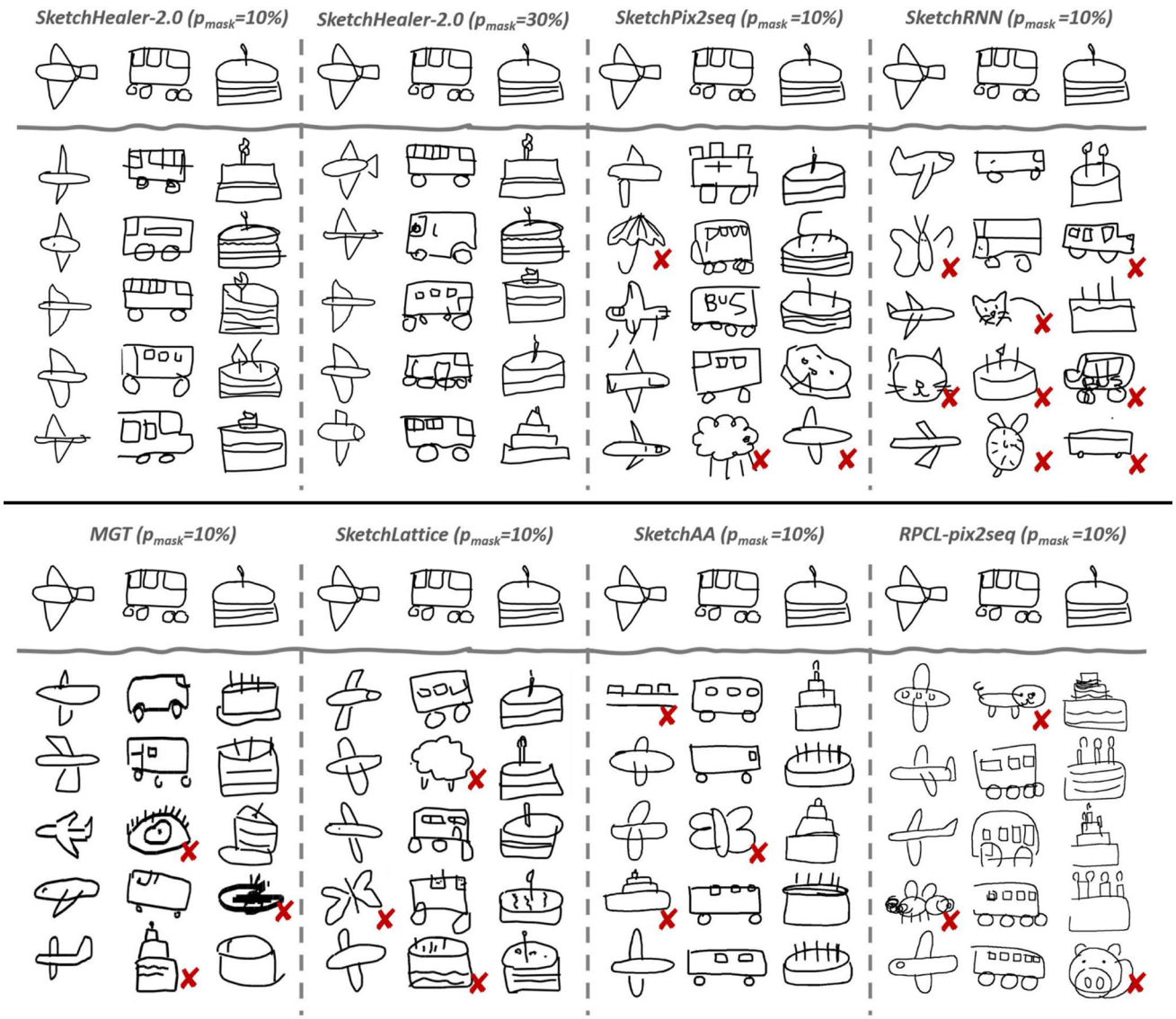

Fig. 7 Qualitative comparisons for sketch-to-sketch retrieval result. Top 5 is returned. Instances surrounded by red cross indicate false positive from wrong category. Our model can still achieve promising results under more challenging scenario ($p_{mask} = 30\%$), while other competitors have many false positives even under mild corruption (Color figure online)

Table 3 Quantitative comparisons between different ablated versions of SketchHealer on sketch healing

p_{mask}	Metric		SketchHealer-1.0 (%)	SketchHealer-Percep (%)	SketchHealer-Static (%)	SketchHealer-2.0 (%)
10%	Recognition	Top 1	49.77	58.62	54.64	**71.32**
		Top 5	69.92	73.88	75.60	**80.01**
		Top 10	71.32	76.29	80.01	**81.63**
	Human	N/A	8.88	27.81	**31.99**	31.32
30%	Recognition	Top 1	41.59	49.08	51.23	**65.27**
		Top 5	62.76	72.09	72.92	**77.38**
		Top 10	68.12	78.32	78.92	**80.27**
	Human	N/A	9.92	9.99	26.70	**53.39**
50%	Recognition	Top 1	26.32	39.02	37.12	**41.13**
		Top 5	48.94	61.23	61.32	**62.12**
		Top 10	57.89	**64.82**	64.21	64.00
	Human	N/A	14.14	18.76	22.02	**45.08**

Best performance are highlighted in bold
Without duplication, definition of Recognition and Human results (%) can be found in Table 1 caption

positives. In contrast, our graph-based representation is not only category-discriminative in the more challenging setting ($p_{mask} = 30\%$), but also learns to respect finer-grained details (e.g., the dense side-by-side windows of the bus).

Impact of perceptual metric The comparison between SketchHealer-1.0 and SketchHealer-2.0 in Table 1 has confirmed the significance of the perceptual metric. In this section, we provide further evidence to unveil its inner workings with more ablated analysis. Specifically, we introduce two more competitors, SketchHealer-Percep and SketchHealer-Static, which differ from SketchHealer-1.0 (L_{recons}) and SketchHealer-2.0 (L_{total}) in training the graph-to-sequence network with loss function L_{percep} and $L_{recons} + L_{percep}$ respectively. Results in Table 3 suggest that among all corruption levels:

(i) Barely depending on per-point reconstruction loss offers the worst performance for sketch healing. While this may not be surprising to many due to the arguments we have elaborated in Sect. 3.3 on the natural mismatch of L_{recon} for this task, it does draw attention that perceptual loss alone (SketchHealer-Percep) is bringing superior performance, sometimes even better than SketchHealer-Static, a naive equal combination of both type of losses.

(ii) Dynamics between L_{percep} and L_{recons} matters. Our full model, SketchHealer-2.0, that integrates two losses in an adaptive fashion, advances SketchHealer-Static with noticeable margins. With Top 1 recognition accuracy improving from 54.64% to 71.32% at $p_{mask} = 10\%$, 51.23% to 65.27% at $p_{mask} = 30\%$ and 37.12% to 41.13% at $p_{mask} = 50\%$.

(iii) With the increasing of corruption levels, the reconstruction loss tends to bring fewer benefits as an optimisation

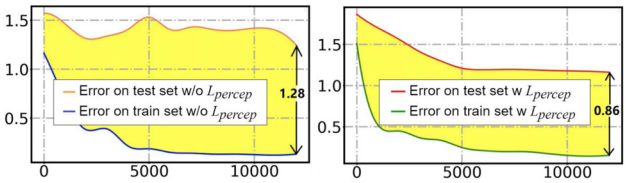

Fig. 8 Importance of the involvement of perceptual metric on model generalisability. Y-axis: loss produced by L_{total}. X-axis: training iterations

objective. At $p_{mask} = 50\%$, only marginal improvement of recognition accuracy under Top 1 (41.13% vs. 39.02%) is observed between SketchHealer-2.0 and SketchHealer-Percep, and even slightly worse under Top 10 (64.00% vs. 64.82%). This aligns with our intuition that given a highly corrupted sketch with visual cues largely missing, reconstructing to a single pre-specified target is too strong a constraint that leans model to severe overfitting. Perceptual loss greatly alleviates this issue by encouraging multi-modal generations and in turn more generalisable healing capabilities. To see it even clearer, we demonstrate the loss curves on both training and testing data during the learning process in Fig. 8—the optimisation landscape of SketchHealer-2.0 progressively fitting on the training data generalises to testing data as well, as opposed to the severe train-test mismatch phenomenon in the lack of perceptual metric. Visualising model's generalisability along their training iterations also allows us to peek into the dynamics between the global and local metric, L_{percep} and L_{recons}. We showcase two exemplary train-test loss curves for partial sketches at corruption level 10% and 50% in Fig. 9 and observe that the success

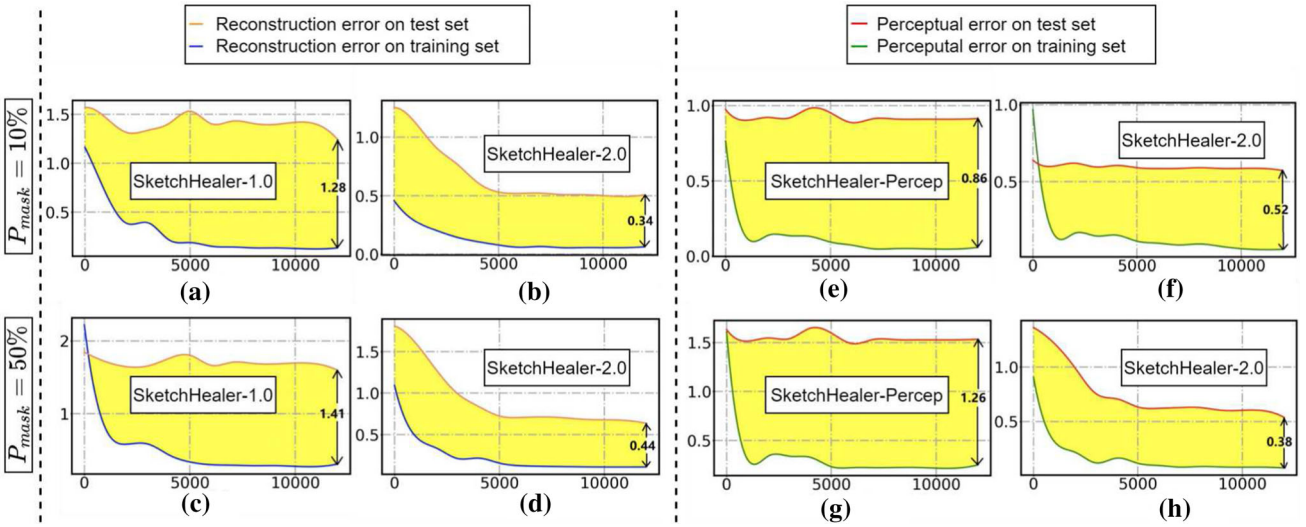

Fig. 9 SketchHealer-2.0 leads to better model generalisability. Compared with the models trained by reconstruction and perceptual metric separately (SketchHealer-1.0 (L_{recons}) and SketchHealer-Percep (L_{percep})), SketchHealer-2.0 can greatly reduce train-test discrepancy under both metrics. X-axis: training iterations

of SketchHealer-2.0 relies critically on (i) an integration of both local and global healing perspective that combines the best of two worlds: SketchHealer-2.0 is able to achieve much smaller generalisation error on both reconstruction and perceptual metric when compared with SketchHealer-1.0 and SketchHealer-percep, two separate models optimised by single metric of L_{recons} and L_{percep} respectively (Fig. 9b,d,f,h vs. a,c,e,g). (ii) the flexibility to adapt between local and global mode: when corruption level is small ($p_{mask} = 10\%$), SketchHealer-2.0 is mainly working with local reconstruction objective so to achieve the best possible healing results and leave perceptual metric nearly un-optimised on test set (Fig. 9b vs. f). In fact, we believe these two insights derived from SketchHealer-2.0 are not surprising. Both of them only further echo the conclusions we've been drawing throughout the paper.

4.4 Application: Sketch-Based Creativity Assistant

We have demonstrated the superior healing capability of SketchHealer-2.0 for partial incomplete single sketch input. This section explores the possibility of whether SketchHealer-2.0 can cope with two similar or distinct partial sketch inputs and what visual output it can render. A desirable result would be a realistic and creative visual primitive with complete and smooth structures and novel but meaningful semantics that combines the key traits from the two. Upon success, this supports a unique and useful application of a sketch-based creativity assistant. Feeding any two sketches (likely corrupted due to the "can't or lazy to sketch" reality) as input, a novel and even imaginative visual object can be automatically rendered and provided as a creative assistant. Specifically, we portray such creative visual manipulation as a latent vector arithmetic problem. Given z encoded from two (partial) sketches, we calculate their sum before forwarding to a generative decoder. Figures 1d and 10 illustrate some typical examples. We can find that SketchHealer-2.0 can reasonably cater to our goal of a creativity assistant by extracting key visual semantics from two partial sketches, and combining them to recreate a novel and interesting sketch primitive. SketchHealer-2.0 also works robustly regardless of the category and the corruption level conformity between the two inputs. See how a corrupted pig sketch plus a partial sheep, or a spider plus a bus transform into a bizarre and surreal rendering with visual traits from both, or the perceptually meaningful within-category fine-grained visual manipulation (e.g., the pig nose).

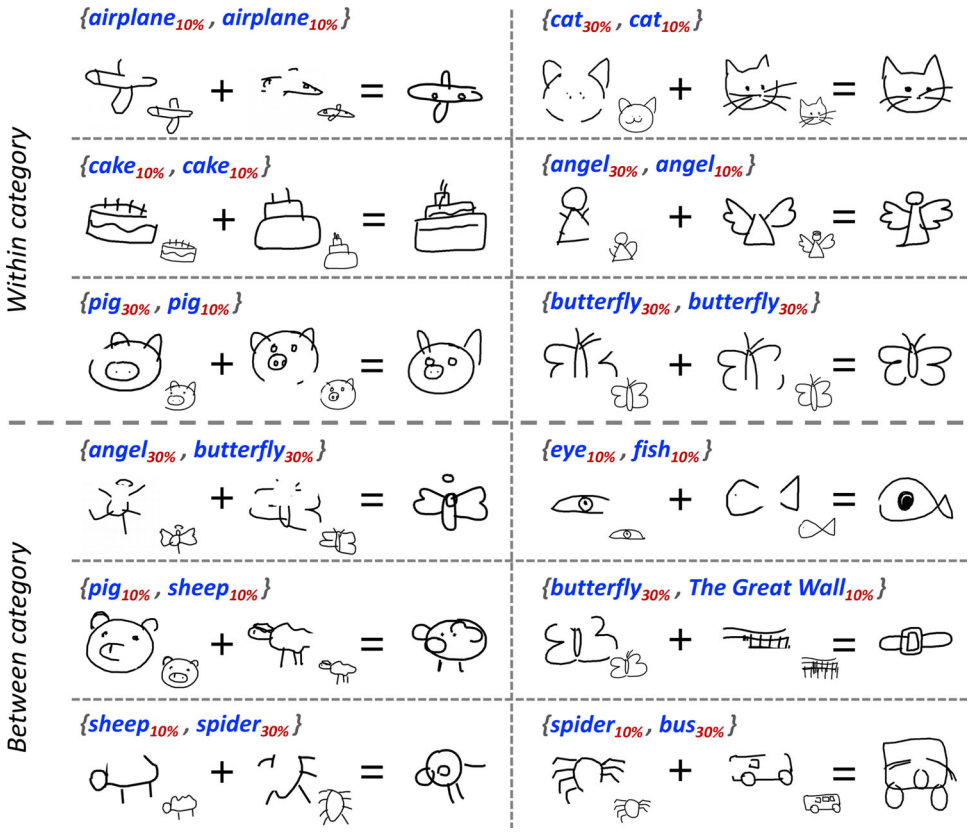

Fig. 10 SketchHealer-2.0 for creative visual manipulation. Given two partial sketches of either same or different corruption level (p_{mask} value) and category, SketchHealer-2.0 is able to recreate a novel sketch by grasping and combining key visual traits from both inputs. This makes SketchHealer-2.0 a potentially effective sketch-based creativity assistant. Subscript under each category name is the input corruption level (p_{mask} value)

5 Conclusion

We introduced the problem of sketch healing that asks a new question: given a partial sketch, can we synthesise a complete and novel sketch that best resembles the partial input. We achieved this by introducing two healing-specific designs that importantly gives us both feature robustness and flexibility in handling missing information. On sketch representation, we uniquely encapsulated two unique traits of sketches (temporal and abstract) into a graph-to-sequence model. On learning objective, we went beyond the traditional single per-point local reconstruction loss and complemented it with a global perceptual loss critical for promising performance. By experiments, we showed that our approach was able to produce visually complete sketches that closely resemble the partial input, whereas alternatives re-purposed for the problem worked less well. We also presented the possibility of our framework as a key enabler for creative visual application.

Funding Funding was provided by the National Natural Science Foundation of China (Grant No. 61601042).

References

Amir, D., & Weiss, Y. (2021). Understanding and simplifying perceptual distances. In *CVPR*.

Barnes, C., Shechtman, E., Finkelstein, A., & Goldman, DB. (2009) . Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM transactions on graphics (Proc. SIGGRAPH)*.

Berger, I., Shamir, A., Mahler, M., Carter, E., & Hodgins, J. (2013). Style and abstraction in portrait sketching. *ACM Transactions on Graphics (TOG), 34*, 1–12.

Bhunia, A. K., Das, A., Muhammad, U. R., Yang, Y., Hospedales, T. M., Xiang, T., Gryaditskaya, Y., & Song, Y. Z. (2020a). Pixelor: A competitive sketching AI agent. so you think you can sketch? In *ACM transactions on graphics (Proc. SIGGRAPH Asia)*.

Bhunia, A. K., Yang, Y., Hospedales, T. M., Xiang, T., & Song, Y. Z. (2020b). Sketch less for more: On-the-fly fine-grained sketch-based image retrieval. In *CVPR*.

Blau, Y., & Michaeli, T. (2019). Rethinking lossy compression: The rate-distortion-perception tradeoff. In *ICML*.

Cao, N., Yan, X., Shi, Y., & Chen, C. (2019). Ai-sketcher: A deep generative model for producing high-quality sketches. In *AAAI*.

Cao, Y., Wang, H., Wang, C., Li, Z., Zhang, L., & Zhang, L. (2010). Mindfinder: Interactive sketch-based image search on millions of images. In *ACM MM*.

Chan, C., Ginosar, S., Zhou, T., & Efros, A. A. (2019). Everybody dance now. In *ICCV*.
Chen, Q., & Koltun, V. (2017). Photographic image synthesis with cascaded refinement networks. In *ICCV*.
Chen, Y., Tu, S., Yi, Y., & Xu, L. (2017) . Sketch-pix2seq: A model to generate sketches of multiple categories. arXiv:1709.04121
Chen, Z. M., Wei, X. S., Wang, P., & Guo, Y. (2019). Multi-label image recognition with graph convolutional networks. In *CVPR*.
Das, A., Yang, Y., Hospedales, T., Xiang, T., & Song, Y. Z. (2020). Bézierskетch: A generative model for scalable vector sketches. In *ECCV*.
Efros, AA., & Leung, TK. (1999) . Texture synthesis by non-parametric sampling. In: ICCV
Eitz, M., Hays, J., & Alexa, M. (2012). How do humans sketch objects? In *ACM Transactions on graphics (Proc. SIGGRAPH)*.
Ge, S., Goswami, V., Zitnick, C. L., & Parikh, D. (2020). Creative sketch generation. In *ICLR*.
Gori, M., Monfardini, G., & Scarselli, F. (2005). A new model for learning in graph domains. In *IJCNN*.
Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv:1308.0850
Ha, D., & Eck, D. (2018). A neural representation of sketch drawings. In *ICLR*.
Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2017). Squeezenet: Alexnet-level accuracy with 50× fewer parameters and <0.5 mb model size. In *ICLR*.
Iizuka, S., Simo-Serra, E., & Ishikawa, H. (2017). Globally and locally consistent image completion. In *ACM transactions on graphics (Proc. SIGGRAPH)*.
Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *CVPR*.
Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *ECCV*.
Johnson, J., Gupta, A., & Fei-Fei, L. (2018). Image generation from scene graphs. In *CVPR*.
Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). Progressive growing of gans for improved quality, stability, and variation. In *ICLR*.
Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *CVPR*.
Kim, T., Cha, M., Kim, H., Lee, J. K., & Kim, J . (2017). Learning to discover cross-domain relations with generative adversarial networks. In *ICML*.
Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. arXiv:1312.6114
Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *ICLR*.
Lahiri, A., Jain, A. K., Agrawal, S., Mitra, P., & Biswas, P. K. (2020). Prior guided gan based semantic inpainting. In *CVPR*.
Li, K., Pang, K., Song, J., Song, Y. Z., Xiang, T., Hospedales, T. M., & Zhang, H. (2018). Universal sketch perceptual grouping. In *ECCV*.
Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., & Yang, M. H. (2017). Universal style transfer via feature transforms. In *NIPS*.
Liu, F., Deng, X., Lai, Y. K., Liu, Y. J., Ma, C., & Wang, H. (2019). Sketchgan: Joint sketch completion and recognition with generative adversarial network. In *CVPR*.
Liu, G., Reda, F. A., Shih, K. J., Wang, T. C., Tao, A., & Catanzaro, B. (2018). Image inpainting for irregular holes using partial convolutions. In *ECCV*.
Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. In *ICLR*.
Pang, K., Li, D., Song, J., Song, Y. Z., Xiang, T., & Hospedales, T. M. (2018). Deep factorised inverse-sketching. In *ECCV*.
Pang, K., Li, K., Yang, Y., Zhang, H., Hospedales, T. M., Xiang, T., & Song, Y. Z. (2019). Generalising fine-grained sketch-based image retrieval. In *CVPR*.
Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in pytorch.
Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *CVPR*.
Pumarola, A., Agudo, A., Martinez, A. M., Sanfeliu, A., & Moreno-Noguer, F. (2018). Ganimation: Anatomically-aware facial animation from a single image. In *ECCV*.
Qi, Y., Su, G., Chowdhury, P. N., Li, M., & Song, Y. Z. (2021). Sketchlattice: Latticed representation for sketch manipulation. In *ICCV*.
Ranjan, A., Bolkart, T., Sanyal, S., & Black, M. J. (2018). Generating 3d faces using convolutional mesh autoencoders. In *ECCV*.
Riaz Muhammad, U., Yang, Y., Song, Y. Z., Xiang, T., & Hospedales, T. M. (2018). Learning deep sketch abstraction. In *CVPR*.
Sagong, M., Shin, Y., Kim, S., Park, S., & Ko, S. (2019). Pepsi: Fast image inpainting with parallel decoding network. In *CVPR*.
Sangkloy, P., Burnell, N., Ham, C., & Hays, J. (2016). The sketchy database: learning to retrieve badly drawn bunnies. In *ACM transactions on graphics (Proc. SIGGRAPH)*.
Sangkloy, P., Lu, J., Fang, C., Yu, F., & Hays, J. (2017). Scribbler: Controlling deep image synthesis with sketch and color. In *CVPR*.
Shen, Y., Liu, L., Shen, F., & Shao, L. (2018). Zero-shot sketch-image hashing. In *CVPR*.
Shih, M. L., Su, S. Y., Kopf, J., & Huang, J. B. (2020). 3d photography using context-aware layered depth inpainting. In *CVPR*.
Simo-Serra, E., Iizuka, S., & Ishikawa, H. (2018). Mastering sketching: Adversarial augmentation for structured prediction. *ACM Transactions on Graphics (TOG), 37*, 1–3.
Song, J., Pang, K., Song, Y. Z., Xiang, T., & Hospedales, T. M. (2018a). Learning to sketch with shortcut cycle consistency. In *CVPR*.
Song, Y., Yang, C., Lin, Z., Liu, X., Huang, Q., Li, H., & Jay Kuo, C. C. (2018b). Contextual-based image inpainting: Infer, match, and translate. In *ECCV*.
Song, Y., Yang, C., Shen, Y., Wang, P., Huang, Q., & Kuo, C. C. J. (2018c). Spg-net: Segmentation prediction and guidance network for image inpainting. In *BMVC*.
Su, G., Qi, Y., Pang, K., Yang, J., Song, Y. Z., & SketchX, C. (2020). Sketchhealer: A graph-to-sequence network for recreating partial human sketches. In *BMVC*.
Tallon, C., Bertrand, O., Bouchet, P., & Pernier, J. (1995). Gamma-range activity evoked by coherent visual stimuli in humans. *European Journal of Neuroscience, 7*, 1285–1291.
Tariq, T., Tursun, O. T., Kim, M., & Didyk, P. (2020). Why are deep representations good perceptual quality features? In *ECCV*.
Theis, L., Oord, A., & Bethge, M. (2016). A note on the evaluation of generative models. In *ICLR*.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *NeurIPS*.
Wang, A., Ren, M., & Zemel, R. (2021). Sketchembednet: Learning novel concepts by imitating drawings. In *ICML*.
Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., & Jiang, Y. G. (2018a). Pixel2mesh: Generating 3d mesh models from single RGB images. In *ECCV*.
Wang, T. C., Liu, M. Y., Zhu, J. Y., Tao, A., Kautz, J., & Catanzaro, B. (2018b). High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*.
Wilczkowiak, M., Brostow, G. J., Tordoff, B., & Cipolla, R. (2005). Hole filling through photomontage. In *BMVC*.
Xiong, W., Yu, J., Lin, Z., Yang, J., Lu, X., Barnes, C., & Luo, J. (2019). Foreground-aware image inpainting. In *CVPR*.
Xu, P., Joshi, C. K., & Bresson, X.(2019). Multi-graph transformer for free-hand sketch recognition. *TNNLS*.

Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., & Li, H. (2017). High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*

Yang, J., Lu, J., Lee, S., Batra, D., & Parikh, D. (2018). Graph R-CNN for scene graph generation. In *ECCV*.

Yang, L., Pang, K., Zhang, H., & Song, Y. Z. (2021a). Sketchaa: Abstract representation for abstract sketches. In *ICCV*.

Yang, L., Zhuang, J., Fu, H., Wei, X., Zhou, K., & Zheng, Y. (2021). Sketchgnn: Semantic sketch segmentation with graph neural networks. *ACM Transactions on Graphics (TOG), 40*, 1–13.

Yao, T., Pan, Y., Li, Y., & Mei, T. (2018). Exploring visual relationship for image captioning. In *ECCV*.

Yi, Z., Zhang, H., Tan, P., & Gong, M. (2017). Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2018). Generative image inpainting with contextual attention. In *CVPR*.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2019). Free-form image inpainting with gated convolution. In *ICCV*.

Yu, Q., Yang, Y., Liu, F., Song, Y. Z., Xiang, T., & Hospedales, T. M. (2017). Sketch-a-net: A deep neural network that beats humans. *IJCV*.

Zang, S., Tu, S., & Xu, L. (2021). Controllable stroke-based sketch synthesis from a self-organized latent space. *Neural Networks, 137*, 138–150.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. N. (2017a). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*.

Zhang, R., Zhu, J. Y., Isola, P., Geng, X., Lin, A. S., Yu, T., & Efros, A. A. (2017b). Real-time user-guided image colorization with learned deep priors. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.

Zheng, C., Cham, T. J., & Cai, J. (2019). Pluralistic image completion. In *CVPR*.

Zhou, H., Liu, Y., Liu, Z., Luo, P., & Wang, X. (2019). Talking face generation by adversarially disentangled audio-visual representation. In *AAAI*.

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017a). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*.

Zhu, J. Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., & Shechtman, E. (2017b). Toward multimodal image-to-image translation. In *NIPS*.