Proceedings of the ASME 2020
International Design Engineering Technical Conferences
and Computers and Information in Engineering Conference
IDETC/CIE2020
August 16-19, 2020, St. Louis, MO, USA

IDETC2020-22394

AN UNSUPERVISED DEEP LEARNING MODEL TO DISCOVER VISUAL SIMILARITY BETWEEN SKETCHES FOR VISUAL ANALOGY SUPPORT

Zijian Zhang
IMPACT Laboratory
Dept. of Aerospace & Mechanical Engineering
University of Southern California
Los Angeles, California 90089
zijianz@usc.edu

Yan Jin*
IMPACT Laboratory
Dept. of Aerospace & Mechanical Engineering
University of Southern California
Los Angeles, California 90089
yjin@usc.edu
(*corresponding author)

ABSTRACT

Visual analogy has been recognized as an important cognitive process in engineering design. Human free-hand sketches provide a useful data source for facilitating visual analogy. Although there has been research on the roles of sketching and the impact of visual analogy in design, little work has been done aiming to develop computational tools and methods to support visual analogy from sketches. In this paper, we propose a computational method to discover visual similarity between sketches, considering the following practical application: Given a sketch drawn by a designer that reflects the designer's rough idea in mind, our goal is to identify the shape similar sketches that can stimulate the designer to make more and better visual analogies. The first challenge in doing so is how to discover the similar shape features embedded in sketches from various categories. To address this challenge, we propose a deep clustering model to learn a latent space which can reveal underlying shape features for multiple categories of sketches and cluster sketches simultaneously. An extensive evaluation of the clustering performance of our proposed method has been carried out in different configurations. The results have shown that the proposed method can discover sketches that have similar appearance, provide useful explanations of the visual relationship between different sketch categories, and has the potential to generate visual stimuli to enhance designers' visual imageries.

Keywords: Visual similarity, unsupervised deep learning, design by analogy, sketching, fixation

1. INTRODUCTION

In engineering design, mental stimulation is useful to boost innovative solutions for ill-defined design problems. During conceptual design, designers, especially the novices, usually struggle in choosing from which sources to gain inspiration when attempting to generate creative concepts. In our previous work, it has been shown that the shapes and structures of a design artifact may be more stimulating than the functions because of its high level of ambiguity[1]. Researchers have observed that designers often search intensively for images from various websites for inspiration [2, 3]. Most existing design-dedicated image search tools and methods [4-6] require designers to initiate search by entering keywords and avoid fixation through semantic-based approaches. The keyword-based search assumes that a designer has a clear goal in mind and can convert and express it by keywords. However, because of the ill-defined nature of design problems at the early design stages, the designers may only have a vague idea of the entity that is being designed. Therefore, requiring designers to formulate keywords when initiating the search can be a major impediment of these computational tools. How can a computational tool support designer to by capturing the vague ideas in their mind and then use them to search and retrieve visual stimuli?

Sketch as a visual representation in engineering design can help brief and ambiguous ideas take shapes on paper[7]. The briefness accelerates the transformation of a flash thought into reality. Ambiguity gives rise to an open-ended representation that contributes to many more possible interpretations. Sketching activity in conceptual design is primarily to provide potentially meaningful clues to infer emerging design concepts[8, 9]. The inspiration of sketches mostly comes from the shapes and the relationships among them. Designers can manipulate given shapes in imagery and combine them into meaningful and even new concepts in a short time. Sketching can reflect premature design ideas in designers' mind, and it is also an ideal stimulant to facilitate creative idea generation. Therefore, it is important to develop a computational tool which takes designers' rough sketches as queries and help them generate more creative ideas by stimulating their visual thinking process.

Research has been done to investigate visual analogy in the field of design. Goldschmidt and colleagues demonstrated visual analogy is considered as an effective cognitive strategy to stimulate designers to create innovative concepts for solving ill-structured design problems[10-12]. For novel idea generation, the use of picture stimuli outperforms words[13, 14]. In design, shapes may represent semantic concepts as well the objects to reflect designers' understanding of the visual world. From a cognitive point of view, when making visual analogy, designer can map shapes from low(geometric) level to high (symbolic, conceptual) level[15, 16]. At the high level, they are capable of interpreting and detecting the similarities between shapes in the same category or not. It means designers can abstract perceptual information to some visual prototypes which represent underlying shape paradigm in a cognitive space. In that space, they can manipulate and transform shapes by exploiting domain knowledge. From an engineering design point of view, high-dimensional design features (geometric space) actually lie on a lower-dimensional design manifold(semantic space)[17, 18]. In the manifold, high-dimensional shape features can be reduced to the minimal dimensionality which can reserve underlying pattern, constraints or configurations. It is more efficient to explore and exploit the low dimensional design space to discover novel designs. In this paper, we call this "high level" and "low dimensional" space as a *latent space*. Therefore, the research problem in this paper is how a computation tool can learn a latent space which can capture the "*visual prototype*" of sketches from different categories.

In this paper, we utilize the unsupervised deep learning to build a model, called *deep clustering sketch-pix2seq* or *dc-sketch-pix2seq* for short, to learn a low dimension and high-level latent space, in which "visual prototypes" can be found to distill underlying shape features of sketches in different categories; after that, a clustering space is learned in which visual similarity of underlying shape features of sketches can be visualized.

2. RELATED WORK

2.1 Computational tools for design by analogy

Design-by-analogy consists of two main steps: retrieving potentially-inspirational information in the source domains and mapping the inspirational information from source domains to the target domain[19]. Designers face difficulties when retrieving an appropriately inspirational source. Therefore, using effective searching and retrieving tools have the potential to enhance design-by-analogy. The large amount of inspirational resources available in various databases can benefit designers who have limited domain knowledge. Many computational tools and methods have been developed to support and enhance searching and retrieval in design-by-analogy. The goals are to strength experts' abilities and reduce the influence of experience gaps. Currently, biological systems and patents are the two major inspiration sources for design-by-analogy.

Biological systems provide a fruitful nature source of inspiration for engineering design. Vincent and Mann proposed Bio-TRIZ, which extends biological information and principals in the TRIZ database[20]. Chakrabarti et al. have created an automated analogical tool called IDEA-INSPIRE that searches relevant ideas from a biological database to solve a given design problem[21, 22]. Shu et al. used natural language analysis to correlate functional basis terms with useful biological keywords[23, 24]. DANE (Design by Analogy to Nature Engine) was proposed by Goel et al. to search and retrieve the functioning of biological systems in the Structure-Behavior-Function (SBF) library [25, 26]. Nagel et al. put forward a computational method to generate biologically inspired concepts based on function-based design tools[27]. AskNature is web-based tool to interactively classify biological information in the Biomimicry Taxonomy[28].

Patent databases can provide enormous cross-domain technology knowledge to inspire designers. Various computational tools and methods have been proposed to retrieve and analyze patents to support design-by-analogy. Murphy proposed a search methodology to identify inspiring patents which have functional semantic similarity with design problems[29]. Fu et al. created a computation method to cluster patents based on their functional and surface similarity, then designers can automatically retrieve analogical stimuli from these patents[30]. As many patent retrieval computational tools focus on mining patents generally, Song and Luo proposed a data-driven method to retrieve patents precisely related to a specific product[31]. Fu et al. proposed a technological distance to measure the "near" and "far" analogical stimuli based on the relative similarity of clusters of patents[32].

While the research into search and retrieve analogies from biological system and patents is prolific, the foundation of the most research is in linguistics and semantic transfer for analogical reasoning. There are few computational tools and methods that support and guide visual analogy.

2.2 Visual analogy in engineering design

CAD, sketches, photographs, line-drawings are major visual sources to promote analogical thinking[2, 33]. In engineering design, many researchers used the large assortment of visual displays to stimulate designers to output creative design concepts. Jin and Benami indicated that meaningfulness and relevance are the two overwhelmingly important creative properties of visual stimuli that influence design stimulation [1]. Yang et al. indicated that the quality and realism of the design can be improved when sketching during concept generation[8, 34]. Goldschmidt et al. demonstrated that visual stimuli are useful for both expert and novice designers to improve the quality of design and more effective for novice designers[11, 12]. Linsey et al. illustrated that designers frequently prefer visual representations to textual descriptions for idea generation and photographs are growing in popularity due to easy retrieval from the Internet[35, 36]. McKoy et al. showed that novice designers can generate higher quality and more novel design concepts when being presented with sketches rather than text-based examples [37].

However, displays of visual representations are less effective to produce creative design than reasoning by visual analogy. Casakin et al. found that if no instructions or directions are provided to guide visual analogy, the quality of the design solutions mostly diminished[38, 39]. It is often said that designers think more visually in their working environment. It means that the designers are more likely to take advantage of shapes and forms of displays as stimuli to tackle given design problems[10]. Shape emergence means unexpected or implicit shape features and relations appear only after the manipulation and transformation of explicit shapes. Visual imagery may provide a theoretical foundation for explicating shape emergence in design by linking shape perception and the high-level cognitive processes of visual reasoning. Therefore, designers take advantage of visual imagery to reinterpret and reformate underlying shapes from the visual stimuli for the idea generation. The precondition for shape emergence is shape ambiguity, which refers to the existence of numerous interpretations of visual representation[40].

Designers are prone to use sketches to represent rough ideas and obtain hints from the shapes of sketches[7]. Sketches as informal visual representations has the property of ambiguity, which makes it possible for engineers to perceive two different shapes from a single sketch. The power of visual analogy is that the designers making the analogy can see the similarities of the underlying shapes, despite differences in superficial shapes. Therefore, sketch is an ideal source to serve as a visual stimulus. How to effectively discover visual similarity from sketches is a major question in our work.

In summary, a rich body of research on design by analogy has yet to be integrated into the extensive work on visual analogy. Our goal in this paper is fill the gap of the two research areas by developing a computational method which can support visual analogy.

3. METHODS

3.1 Deep learning models for sketch representation

Recent advances in deep neural network models drastically increased machines' ability to learn a common and general feature space for sketches and images [41-43]. Karimi et al. used a supervised learning method to learn the feature vectors of sketches given the category labels and then create clusters of visually similar sketches based on the learned feature vectors[44]. However, in this paper, we want to learn a latent space which can represent the underlying shape feature of objects only using lines and curves of shapes in the sketches rather than the labels of categories. Therefore, we need to turn to unsupervised learning. Sketch-rnn is an unsupervised learning model based on Variational AutoEncoder (VAE) framework for constructing stroke-based drawings of common objects; it can mimic how human sketch and draw similar but unique objects[45, 46]. Sketch-rnn uses a bi-directional recurrent neural network RNN[47] as an encoder to capture the features of training data in a latent space Z (e.g., the feature distribution of training data) and applies an autoregressive RNN[48] as a decoder to reconstruct data via a sampled vector z from Z. It means all training data can be mapped to a latent space Z which can capture abstract and underlying shape features. However, the performance of sketch-rnn to extract shape features of objects from multiple categories is not satisfactory. Therefore, a modified sketch-rnn should be put forward to robustly present underlying shape features of multi-category objects in a latent space, which can support the measurement of shape similarity.

During the training stage, the input of sketch-rnn is a set of n sketches $\boldsymbol{x} = \{x_i \epsilon X\}_{i=1}^{n}$. X is the data space. The VAE encoder compresses $\boldsymbol{x}$ into n latent vector $\boldsymbol{z} = \{z_i \epsilon Z\}_{i=1}^{n}$. Z is the latent space. The dimensionality of Z is typically much smaller than X in order to avoid the "curse of dimensionality"[49]. The VAE decoder samples n sketches $\boldsymbol{x}' = \{x_i' \epsilon X\}_{i=1}^{n}$ conditional on given latent vector $\boldsymbol{z}$. The entire training process optimizes the following loss function:

$$L(\theta, \phi; x) = E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}'|\boldsymbol{z})] - D_{KL}\big(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \| p_\theta(\boldsymbol{z})\big) \quad (1)$$

where $q(\cdot)$ denotes the encoder, and $p(\cdot)$ denotes the decoder. ϕ and θ are the parameters to be trained in the encoder and decoder, respectively. The parameters are typically the weights and biases of the neural networks. The first term, $E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}(\cdot)$, is the reconstruction loss that ensures the similarity between the generated strokes and the strokes within the sketches in the training set. The second term, $D_{KL}(\cdot)$ is the Kullback-Leibler (KL) loss that ensures the distribution of the generated strokes $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is similar to that of the training set $p_\theta(\boldsymbol{z})$.

Sketch-pix2seq is a modified version of sketch-rnn[50]. There are two modifications in sketch-pix2seq. First, it replaces the bidirectional RNN by a convolutional neural network (CNN) as the encoder. The reason is CNN has a good performance in capturing local structure of images. As the way of human to perceive a sketch is based on shape instead of remembering the sketching process, it means CNN seems to be better choice for encoder. Second, it removes $D_{KL}(\cdot)$ loss from $L(\theta, \phi; x)$. The assumption $p_\theta(\boldsymbol{z}) \sim N(0, I)$ in the sketch-rnn model might not be suitable, as the distribution of the input sketches which belong to different categories could be drawn from other distributions rather than Gaussian distribution. Because of this assumption, the term $D_{KL}\big(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \| p_\theta(\boldsymbol{z})\big)$ forces the encoder to learn the posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ which should be similar to Gaussian distribution. This contributes to the unsatisfied performance of sketch-rnn for learning latent features from sketches in multiple categories. Therefore, the loss function of the sketch-pix2seq model is:

$$L_r = E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}'|\boldsymbol{z})] \quad (2)$$

3.2 Deep clustering sketch-pix2seq

For sketch-pix2seq, input sketches are encoded in the latent space according to their shapes, since samples that look similar are close to each other[50]. Therefore, even sketches are from different categories, we can detect the similar underlying shapes within them in the latent space. Clustering is an unsupervised learning method which can cluster similar data points into in the same group. How can we cluster sketches based on shape

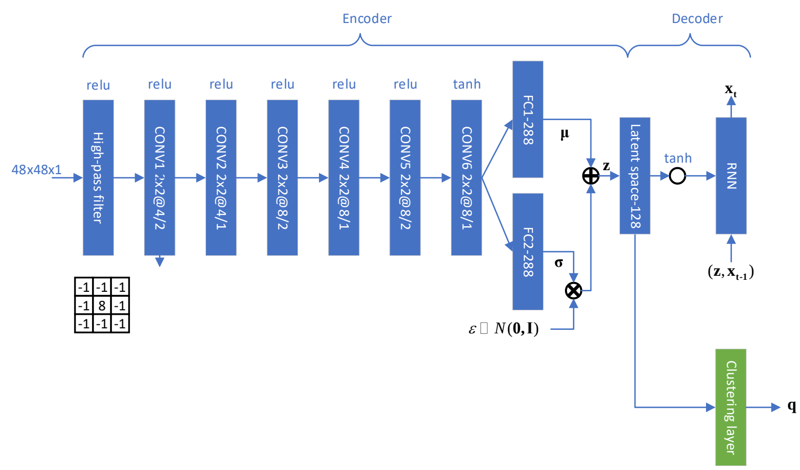


Figure 1: Structure of Deep Clustering Sketch-Pix2Seq

similarity of sketches in the latent space? Deep Embedded Clustering (DEC) provides a way to deal with this problem[51]. The biggest contribution of DEC is the clustering layer (or target distribution P, to be specific). It can output a soft label between the data points and the cluster centroids and make each cluster denser and away from other clusters.

The design of the deep clustering sketch-pix2seq model, called dc-sketch-pix2seq, extends sketch-pix2seq with a clustering layer, as shown in Figure 1. The clustering layer and loss are directly borrowed from DEC. The clustering layer clusters all latent vectors in the latent space Z by simultaneously learning a set of K cluster centers $\{\mu_j \epsilon Z\}_{j=1}^{K}$ and mapping each latent vector z_i into a soft label q_i by student's t-distribution[52]. $\boldsymbol{q_i} = [q_{i1}, \ldots, q_{ij}, \ldots q_{ik}]$ is a soft label which quantifies the similarity between z_i and cluster center μ_j .

$$q_{ij} = \frac{\left(1 + \|z_i - \mu_j\|^2\right)^{-1}}{\sum_j \left(1 + \|z_i - \mu_j\|^2\right)^{-1}} \tag{3}$$

where q_{ij} is the jth entry of $\boldsymbol{q_i}$, representing the probability of z_i belonging to cluster j.

The clustering loss L_c is defined as a KL divergence between the distribution of soft labels Q measured by student's t distribution and the predefined target distribution P derived from Q. The clustering loss is defined as

$$L_c = D_{KL}(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{4}$$

where the target distribution P is defined as

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j (q_{ij}^2 / \sum_i q_{ij})} \tag{5}$$

Raising q_{ij} to the second power and then dividing by the frequency per cluster allows the target distribution P to improve cluster purity and stress on confident labels, while normalizing the contribution of each centroid on the clustering loss to prevent large clusters from distorting the latent space. Therefore, data points with high confidence are used as supervision and points in each cluster distribute more densely. where D_{KL} measures the non-symmetric difference between two probability distributions, P and Q are defined by (5) and (3), and matches Q to P.

In dc-sketch-pix2seq, two components are essential: the sketch-pix2seq and clustering loss. The sketch-pix2seq is used to learn representations in an unsupervised manner and the learned latent space can preserve underlying shape features in sketches. The clustering loss is responsible for manipulating the latent space in order to cluster sketches based on shape similarity.

Therefore, the objective of dc-sketch-pix2seq is

$$L_{rc} = L_r + \tau L_c \tag{6}$$

where L_r is a reconstruction loss from **Error! Reference source not found.** and L_c is a clustering loss from (4). The coefficient τ is better to be less than 1 and more than 0. When $\tau = 0$, the above function reduces to the objective of sketch-pix2seq. When $\tau = 1$ and $L_r = 0$, the above function reduces to the objective of DEC.

We first pretrain the parameters of the dc-sketch-pix2seq by setting $\tau = 0$ to get a latent space. After pretraining, the cluster centers are initialized by performing k-means on latent features of all sketches to get initial cluster centers $\{\mu_j \epsilon Z\}_{j=1}^{K}$. Based on (3) and (5), we can get the initial distribution of soft labels Q and initial target distribution P. Then update the deep clustering weights, cluster centers and target distribution P as follows.

1) Update weights and cluster centers. The gradients of L_c with respect to each latent vector z_i and each cluster center u_j can be computed as:

$$\frac{\partial L_c}{\partial z_i} = 2\sum_{j=1}^{k}\left(1+\left\|z_i-\mu_j\right\|^2\right)^{-1}(p_{ij}-q_{ij})(z_i-\mu_j) \quad (7)$$

$$\frac{\partial L_c}{\partial u_j} = 2\sum_{i=1}^{n}\left(1+\left\|z_i-\mu_j\right\|^2\right)^{-1}(q_{ij}-p_{ij})(z_i-\mu_j) \quad (8)$$

Encoder and decoder parameter gradient $\frac{\partial L_r}{\partial \phi}$ and $\frac{\partial L_r}{\partial \theta}$ can be calculated by backpropagation when passing $\frac{\partial L_c}{\partial z_i}$ to the structure of our model. Then, the weights of encoder and decoder and the cluster center can be updated by mini-batch stochastic gradient decent.

2) Update target distribution. In every T epoch, the target distribution P serves as ground truth soft labels. The clustering layer is trained by predicting the soft assignment Q and then matching it to the target distribution P. At the end of T iterations, based on (5), the target distribution P is updated depending on predicted soft label Q and used for next round of T iterations. After each T iteration, the cluster label c_i assigned to x_i is obtained by:

$$c_i = \arg\max_j q_{ij} \quad (9)$$

where q_{ij} can be obtained by (3). We will stop training if cluster label assignment change (in percentage) between two consecutive T iterations is less than a threshold tol.

4. EXPERIMENTS

4.1 Datasets and Settings

Datasets. Quickdraw is a largest sketch database up to date built by Google[53]. All the data were collected by The Quick, Draw!, an online game that requires participants to draw a sketch within 20 seconds. The 75K sketches for each category have already been divided into training, validation and testing sets with sizes of 70K, 2.5K and 2.5K, respectively. It contains 345 categories of everyday objects. In this paper, the raw sequences from Quickdraw datasets are converted to monochrome png files of size 48x48, which are used as the input data for our deep neural network. These png files are binary images with pixels covered by strokes having the value one and the rest of pixels the value zero. In order to study the clustering performance of our proposed model, we conduct experiment on three datasets:

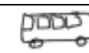
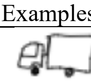
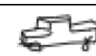
Dataset 1: it includes five categories which are van, bus, truck, pickup truck, car. All of them belong to automobiles and share some obvious shape features such as wheels and windows.

Dataset 2: it includes five categories which are speedboat, canoe, drill, pickup truck and car. Speedboat and canoe belong to boats and share some obvious shape features such as v-shaped hulls. Pickup truck and car belong to automobiles. Drill is the only one which doesn't share general shape similarity with other categories.

Dataset 3: it includes five categories which are television, canoe, drill, umbrella, car. Each of them doesn't share any general shape similarity with other categories.

Some examples of each dataset are listed in Table 1. The 15K sketches for each category are chosen in this paper. The sketches are been divided into training, validation and testing sets with sizes of 10K, 2.5K and 2.5K, respectively.

Table 1: Examples of Datasets

Dataset	Examples				
1					
	van	bus	truck	pickup truck	car
2					
	speedboat	canoe	drill	pickup truck	car
3					
	television	canoe	drill	umbrella	car

Comparing methods. We demonstrate the effectiveness of our dc-sketch-pix2seq mainly by comparing with sketch-pix2seq which can be viewed as a special case of dc-sketch-pix2seq when the constant of the clustering loss is set to zero and sketch-rnn which is the origin of sketch-pix2seq. we use the publicly available code released by the author to report the performance of sketch-pix2seq and sketch-rnn. For the sake of completeness, one of the traditional clustering algorithms, k-means is included in comparison. Since another classic clustering algorithm gaussian mixture models (GMM) performs similarly to k-means, we only report k-means results. We show qualitative and quantitative results that demonstrate the benefit of dc-sketch-pix2seq over other methods.

Implementation Details. We conduct experiments on four methods with three data settings. The parameters used for training sketch-rnn and sketch-pix2seq models are the same as the illustration in the papers[45, 50]. For training our proposed model, we copy the trained weights of each layer in sketch-pix2seq. The coefficient τ of clustering loss in (6) is set to 0.05 which is determined by a grid search in {0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0} and batch size to 100 for all datasets. The maximum number of epochs is set to $T = 50$. In each iteration, we train the encoder for one epoch using Adam optimizer[54] with learning rate $\lambda = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. The convergence threshold is set to $tol = 0.1\%$. We implement our models end-to-end based on Python and Keras[55]. The dimension of the latent space in these three models is 128, which is the same in the papers[45, 50]. k-means is performed to cluster sketches in the latent space of pretrained sketch-pix2seq and sketch-rnn. Besides, as a baseline for comparison, k-means also runs on sketch data with original dimensions which is much larger than the latent space. k-means performs 20 times with different initialization and the result with best objective value is chosen. $k = 5$, as each dataset includes 5 categories.

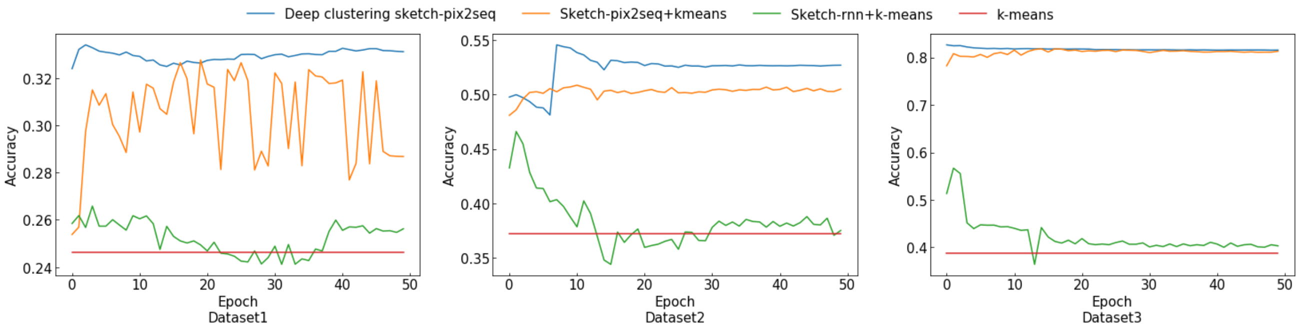


Figure 2: Clustering Accuracy for Different Datasets of Each Method

Evaluation metrics. We evaluate all clustering methods with unsupervised clustering accuracy (ACC). The ACC is defined as the best match between ground truth $\boldsymbol{y}$ and predicted cluster labels $\boldsymbol{c}$:

$$ACC(\boldsymbol{y},\boldsymbol{c}) = \max_{m\in\mathcal{M}} \frac{\sum_{i=1}^{n} \mathbf{1}\{y_i = m(c_i)\}}{n} \quad (10)$$

where n is the total number of samples, y_i is the ground truth label, c_i is the predicted cluster label of example x_i obtained by the model, and $\mathcal{M}$ is the set of all possible one to one mappings between predicted cluster labels to ground truth labels. The best mapping can be efficiently computed by the Hungarian algorithm [56].

4.2 Experiment results

Quantitative results. The order of computation time for training three deep learning methods is Dc-sketch-pix2seq > Sketch-rrn>Sketch-pix2seq. As Sketch-pix2seq doesn't need to compute $D_{KL}(\cdot)$ loss from $L(\theta,\phi;x)$ in (1), it takes less time than sketch-rnn to train. Dc-sketch-pix2seq takes the longest time for training. As it needs to pretrain and copy the weights from sketch-pix2seq, it takes almost twice as much time as Sketch-pix2seq to be trained.

We plot the clustering accuracy of all comparing algorithms on 3 datasets in Figure 2. In Table 2, the best accuracy rate in 50 epochs for each method is chosen for comparison. As it shows, we can see a rising trend of accuracy rate from Dataset1 to Dataset3 for each method. This is because it is easier to differentiate sketches from different taxonomic groups than from the same taxonomic groups. It can imply sketches in the same taxonomic group share more shape features. Deep neural network-based clustering algorithms sketch-rnn+kmeans, sketch-pix2seq+kmeans and dc-sketch-pix2seq outperform traditional clustering algorithm k-means for sketches clustering. Except for dataset3, there is a large margin between our proposed method with other methods, which indicates the fascinating potentials of dc-sketch-pix2seq in discovering underlying shape features of sketches in unsupervised clustering field. The performance gap between sketch-pix2seq+kmeans and dc-sketch-pix2seq reflects the effect of clustering loss. Especially for dataset1, sketch-pix2seq has a much larger variance than our proposed model. It means our proposed model is more robust to discover underlying shape features. The outperformance of dc-sketch-pix2seq over sketch-rnn demonstrates that the CNN encoder can help improve clustering performance. Dc-sketch-pix2seq is based on unsupervised learning and have a good performance on learning discriminative representations of sketches. We didn't compare it with a supervised learning method [44] which can reach 76% average accuracy rate for 345 categories. When sketches are from different taxonomic categories (such as Dataset3), they have distinguishing shape features. Our proposed model can have high accuracy rate. When sketches are from the same taxonomic category (such as Dataset1), they share more common shape features. The accuracy rate of the proposed method decreases a lot. It means dc-sketch-pix2seq discriminate sketches based on shape features rather than labels. In other words, it can discover and extract underlying shape features and represent them in the latent space which can differentiate sketches from different categories.

Table 2: Comparison of the Best Clustering Accuracy Rate

Method	Dataset1	Dataset2	Dataset3
k-means	0.25616	0.372	0.38752
Sketchrnn+k-means	0.26584	0.466	0.56696
Sketch-pix2seq+kmeans	0.3276	0.50856	0.81832
Dc-sketch-pix2seq(ours)	0.334	0.54568	0.82664

Qualitative results. We visualize the learned latent space of dc-sketch-pix2seq, sketch-pix2seq and sketch-rnn on these datasets. To this end, we use t-SNE[52] to reduce the dimensionality of the latent representation z from 128 to 2, and plot 2500 testing sketches from each category in Figure 3, Figure 4 and Figure 5. From these figures, we can see that the deep clustering sketch-pix2seq which uses a CNN encoder and a clustering loss performs the best in clustering tasks and sketch-rnn is the worst one. It can also be observed that the latent space learned by dc-sketch-pix2seq are better than the other two, since the sketches from different categories are more separable and the sketches from the same category is denser in all cases. For dataset1, all sketches are from the same taxonomy. It is hard for deep learning models to cluster them as they share too many shape features. In figure 3, The red, black and green clusters are

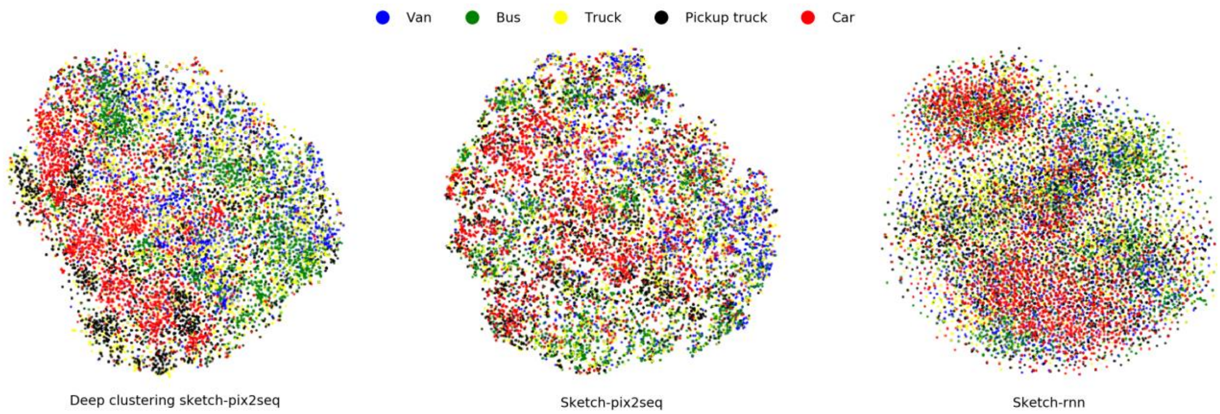


Figure 3: Clustered Latent Space of Dataset1 for Each Method

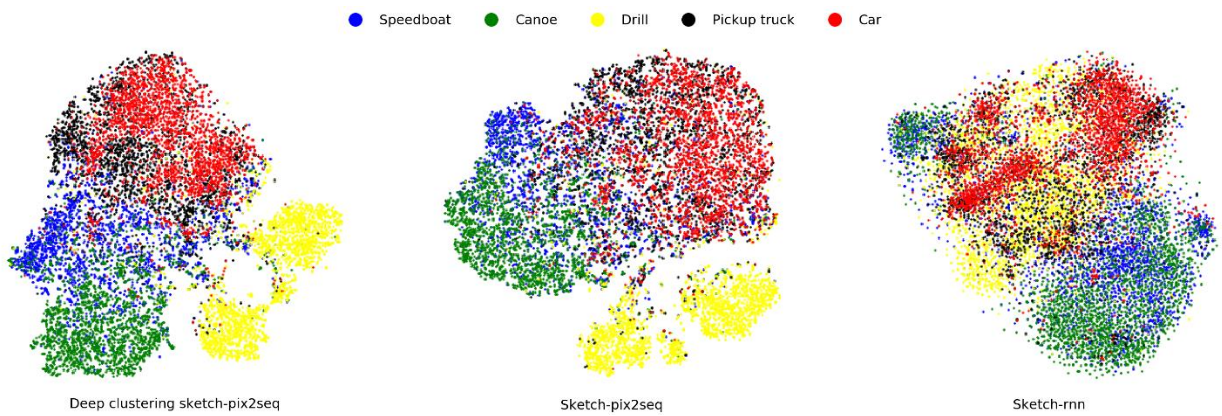


Figure 4: Clustered Latent Space of Dataset2 for Each Method

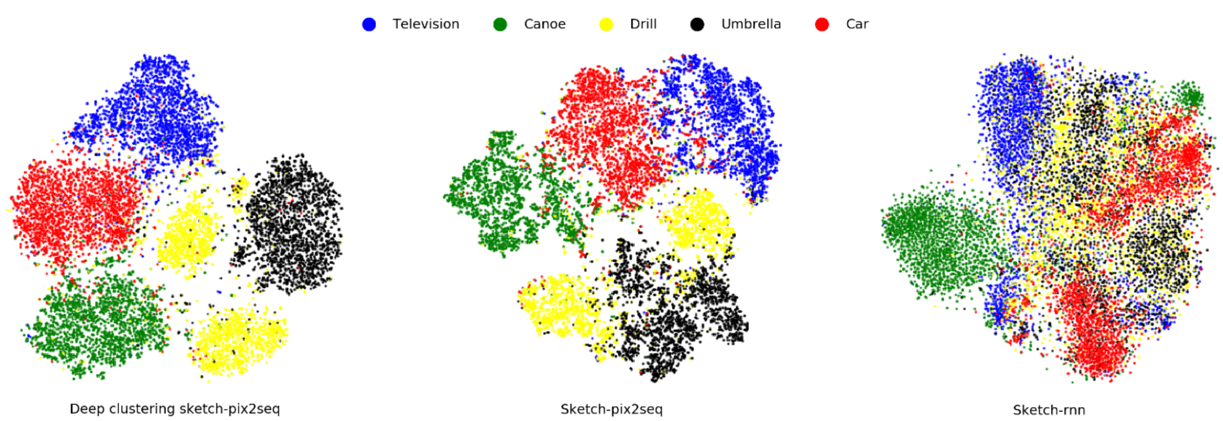


Figure 5: Clustered Latent Space of Dataset3 for Each Method

denser in dc-sketch-pix2seq than the other two as clustering loss can force sketches from the same category to be gathered together and push away sketch from different categories. For Dataset2, sketches are from three taxonomies. Sketch categories belong to the same taxonomy should be close to each other as they share more shape features and away from other taxonomies.

This assumption can be confirmed by our proposed model and sketch-pix2seq, as they both use CNN as an encoder which can discover and represent underlying shape structures in the latent space. For Figure 4, car cluster is close to pickup truck cluster and speedboat cluster is close to canoe cluster in the first plot, while this cannot be easily detected in the third plot. For Dataset3, all sketches are from different taxonomies. Three deep learning models can somehow easily cluster each category. Clusters in our proposed model are denser and with a larger margin with each other. Furthermore, incorrectly clustered sketches of our model are mostly located at the border of each cluster, where confusing sketches usually appear. In contrast, more incorrectly clustered sketches of sketch-pix2seq and sketch-rnn appear in the interior of the clusters.

5. DISUCSSION

Deep clustering based on underlying shape features. There are two ways to cluster data. One is to use traditional clustering algorithm, such as K-means or GMM, on the original data. Another one is to use deep clustering learning to reduce high-dimensional original data to low dimensional feature which can capture some pattern in the original data. Based on our experiments, clustering performance of deep clustering learning is better than traditional clustering algorithm with a large margin. Deep clustering learning can be broadly classified into two categories. The first one is to use deep neural network to learn the features of original data in the latent space by optimizing reconstruction loss and then apply clustering algorithm on the learned features. The second one is to explicitly define a clustering loss and combine it with reconstruction loss; it can obtain feature representations and cluster assignments simultaneously. Dc-sketch-pix-pix2seq falls in the second category, sketch-pix2seq+kmeans and sketch-rnn+k-means fall in the first category. As illustrated in the comparison of clustering accuracy, the large performance gap between sketch-pix2seq and sketch-rnn reflects convolutional layers is better capturing underlying shape features. As clustering loss can help improve clustering performance, dcsketch-pix2seq outperforms over sketch-pix2seq.

Shape feature extraction for various dataset. By visualizing the latent space of 3 different datasets with different levels of common shape feature sharing, we empirically validate two points: 1) if sketches are from the same taxonomy, they will share too many features. It is difficult for deep clustering models to separate them. The sketches in Dataset1 are from the same taxonomy, three models are struggling to cluster sketches. But our proposed model can somehow cluster separate red and black points from others. The sketches in Dataset3 from different taxonomies, it is easier for three models to cluster sketches. Our proposed model can separate clusters with larger margin; 2) if a deep clustering model uses CNN layers to encoder input sketches and take advantage of clustering loss, its clustering performance can be improved. Some of sketches in Dataset2 come from the same taxonomy, our model can cluster points denser than the other two models. These two facts indicate that our proposed model has higher capability to preserve the inherent and underlying shape features of sketches.

Empower visual analogy. Goldschmidt pointed out visual similarity can happen if visual representation that are rich with clues[2]. Sketch is an abstract representation of a design idea. Sketching is also useful as a way of visual imagery to generate concepts[8]. Kosslyn explains this as people can discover patterns embedded in visual representation and mentally modify the patterns which lead to visual imagery[57]. In this paper, the proposed deep learning model can construct a latent space for sketches. In that space, sketches can be abstractly represented. Therefore, some nonobvious visual similarities can be determined to help designers avoid visual fixation. In Figure 5, we can see some drills have similar shapes with some umbrellas as they have shorter distance even some overlaps. By visualizing this relationship, designers can jump out of the box to seek new concepts or modify the existing ones. The meaningfulness of our tool is providing external visual stimuli for designers to enhance their visual analogy capabilities.

6. CONCLUSION

In this paper, we propose an unsupervised deep clustering model to learn underlying shape feature presentations to detect visual similarity, which can be potentially useful to promote visual analogy in engineering design. In summary, the main contributions of this paper are:

1) An unsupervised deep clustering model is introduced which is trained with reconstruction and clustering losses to make it discover and extract underly shape features of sketches among different categories.
2) The extensive experiments have been conducted that demonstrate the effectiveness and robustness of our proposed model in using high level and low dimensional space to represent low level and high dimensional space.
3) A visualization method is introduced for understanding the visual similarity between sketches from the same and different taxonomy, which is helpful to guide designers' visual imagery.
4) A tool is provided which can potentially promote visual analogy making in conceptual design and boost idea generation.

The precondition for searching and retrieving visual analogy is a visual similarity existing between source and target domains. Hence, the future work is how to effectively and quantitatively measure visual similarity between sketches in the latent space which can support the explication and measurement of their relations of the underlying structure, despite differences in superficial features. When a given sketch from a designer is as a query, a computational tool can return sketches within the same category or from different categories as visual stimuli. Human subject-based design experiments will also be conducted to evaluate the effectiveness of the tool as the tool matures.

REFERENCES

[1] Y. Jin and O. Benami, "Creative patterns and stimulation in conceptual design," *AI EDAM,* vol. 24, no. 2, pp. 191-209, 2010.

[2] G. Goldschmidt, "Visual analogy—a strategy for design reasoning and learning," in *Design knowing and learning: Cognition in design education*: Elsevier, 2001, pp. 199-219.

[3] C. Mougenot, C. Bouchard, A. Aoussat, and S. Westerman, "Inspiration, images and design: an investigation of designers' information gathering strategies," *Journal of Design Research,* vol. 7, no. 4, pp. 331-351, 2008.

[4] C. Bouchard, J.-f. Omhover, C. Mougenot, A. Aoussat, and S. J. Westerman, "TRENDS: a content-based information retrieval system for designers," in *Design Computing and Cognition'08*: Springer, 2008, pp. 593-611.
[5] A. Chakrabarti, L. Siddharth, M. Dinakar, M. Panda, N. Palegar, and S. Keshwani, "Idea Inspire 3.0—A tool for analogical design," in *International Conference on Research into Design*, 2017, pp. 475-485: Springer.
[6] J. Han, F. Shi, L. Chen, and P. R. Childs, "A computational tool for creative idea generation based on analogical reasoning and ontology," *AI EDAM,* vol. 32, no. 4, pp. 462-477, 2018.
[7] D. G. Ullman, S. Wood, and D. Craig, "The importance of drawing in the mechanical design process," *Computers & graphics,* vol. 14, no. 2, pp. 263-274, 1990.
[8] M. C. Yang, "Observations on concept generation and sketching in engineering design," *Research in Engineering Design,* vol. 20, no. 1, pp. 1-11, 2009.
[9] V. Kokotovich and T. Purcell, "Mental synthesis and creativity in design: an experimental examination," *Design studies,* vol. 21, no. 5, pp. 437-449, 2000.
[10] G. Goldschmidt and M. Smolkov, "Variances in the impact of visual stimuli on design problem solving performance," *Design Studies,* vol. 27, no. 5, pp. 549-569, 2006.
[11] G. Goldschmidt, "The backtalk of self-generated sketches," *Design issues,* vol. 19, no. 1, pp. 72-88, 2003.
[12] H. Casakin and G. Goldschmidt, "Expertise and the use of visual analogy: implications for design education," *Design Studies,* vol. 20, no. 2, pp. 153-175, 1999.
[13] K. S. Marshall, R. Crawford, and D. Jensen, "Analogy seeded mind-maps: A comparison of verbal and pictorial representation of analogies in the concept generation process," in *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2016, pp. V007T06A010-V007T06A010: American Society of Mechanical Engineers.
[14] R. A. Malaga, "The effect of stimulus modes and associative distance in individual creativity support systems," *Decision Support Systems,* vol. 29, no. 2, pp. 125-141, 2000.
[15] J. Gero and M. Yan, "Shape emergence by symbolic reasoning," *Environment and Planning B: Planning and Design,* vol. 21, no. 2, pp. 191-212, 1994.
[16] R. Oxman, "The thinking eye: visual re-cognition in design emergence," *Design Studies,* vol. 23, no. 2, pp. 135-164, 2002.
[17] W. Chen, M. Fuge, and J. Chazan, "Design Manifolds Capture the Intrinsic Complexity and Dimension of Design Spaces," *Journal of Mechanical Design,* vol. 139, no. 5, p. 051102, 2017.
[18] W. Chen and M. Fuge, "Beyond the Known: Detecting Novel Feasible Domains Over an Unbounded Design Space," *Journal of Mechanical Design,* vol. 139, no. 11, pp. 111405-111405-10, 2017.
[19] J. S. Linsey, K. L. Wood, and A. B. Markman, "Modality and representation in analogy," *Ai Edam,* vol. 22, no. 2, pp. 85-100, 2008.
[20] J. F. Vincent and D. L. Mann, "Systematic technology transfer from biology to engineering," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences,* vol. 360, no. 1791, pp. 159-173, 2002.
[21] A. Chakrabarti, P. Sarkar, B. Leelavathamma, and B. Nataraju, "A functional representation for aiding biomimetic and artificial inspiration of new ideas," *Ai Edam,* vol. 19, no. 2, pp. 113-132, 2005.
[22] P. Sarkar and A. Chakrabarti, "The effect of representation of triggers on design outcomes," *Ai Edam,* vol. 22, no. 2, pp. 101-116, 2008.
[23] H. Cheong, I. Chiu, L. Shu, R. B. Stone, and D. A. McAdams, "Biologically meaningful keywords for functional terms of the functional basis," *Journal of Mechanical Design,* vol. 133, no. 2, p. 021007, 2011.
[24] L. Shu and H. Cheong, "A natural language approach to biomimetic design," in *Biologically Inspired Design*: Springer, 2014, pp. 29-61.
[25] S. Vattam, B. Wiltgen, M. Helms, A. K. Goel, and J. Yen, "DANE: fostering creativity in and through biologically inspired design," in *Design Creativity 2010*: Springer, 2011, pp. 115-122.
[26] A. K. Goel, S. Vattam, B. Wiltgen, and M. Helms, "Cognitive, collaborative, conceptual and creative—four characteristics of the next generation of knowledge-based CAD systems: a study in biologically inspired design," *Computer-Aided Design,* vol. 44, no. 10, pp. 879-900, 2012.
[27] J. K. Nagel and R. B. Stone, "A computational approach to biologically inspired design," *AI EDAM,* vol. 26, no. 2, pp. 161-176, 2012.
[28] J.-M. Deldin and M. Schuknecht, "The AskNature database: enabling solutions in biomimetic design," in *Biologically inspired design*: Springer, 2014, pp. 17-27.
[29] J. Murphy, K. Fu, K. Otto, M. Yang, D. Jensen, and K. Wood, "Function based design-by-analogy: a functional vector approach to analogical search," *Journal of Mechanical Design,* vol. 136, no. 10, p. 101102, 2014.
[30] K. Fu, J. Cagan, K. Kotovsky, and K. Wood, "Discovering structure in design databases through functional and surface based mapping," *Journal of mechanical Design,* vol. 135, no. 3, p. 031006, 2013.
[31] B. Song and J. Luo, "Mining patent precedents for data-driven design: the case of spherical rolling robots," *Journal of Mechanical Design,* vol. 139, no. 11, p. 111420, 2017.
[32] K. Fu, J. Chan, J. Cagan, K. Kotovsky, C. Schunn, and K. Wood, "The meaning of “near” and “far”: the impact of structuring design databases and the effect of distance of analogy on design output," *Journal of Mechanical Design,* vol. 135, no. 2, p. 021007, 2013.
[33] M. Gonçalves, C. Cardoso, and P. Badke-Schaub, "What inspires designers? Preferences on inspirational

approaches during idea generation," *Design studies,* vol. 35, no. 1, pp. 29-53, 2014.
[34] B. Macomber and M. Yang, "The role of sketch finish and style in user responses to early stage design concepts," in *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2012, pp. 567-576: American Society of Mechanical Engineers Digital Collection.
[35] J. S. Linsey, E. Clauss, T. Kurtoglu, J. Murphy, K. Wood, and A. Markman, "An experimental study of group idea generation techniques: understanding the roles of idea representation and viewing methods," *Journal of Mechanical Design,* vol. 133, no. 3, p. 031008, 2011.
[36] O. Atilola, M. Tomko, and J. S. Linsey, "The effects of representation on idea generation and design fixation: A study comparing sketches and function trees," *Design studies,* vol. 42, pp. 110-136, 2016.
[37] F. L. McKoy, N. Vargas-Hernández, J. D. Summers, and J. J. Shah, "Influence of design representation on effectiveness of idea generation," *Proceedings of ASME DETC, Pittsburgh, PA, Sept,* pp. 9-12, 2001.
[38] H. Casakin, "Visual analogy as a cognitive strategy in the design process: Expert versus novice performance," *journal of Design Research,* vol. 4, no. 2, p. 124, 2004.
[39] H. P. Casakin and G. Goldschmidt, "Reasoning by visual analogy in design problem-solving: the role of guidance," *Environment and Planning B: Planning and Design,* vol. 27, no. 1, pp. 105-119, 2000.
[40] G. Stiny, "Emergence and continuity in shape grammars," in *CAAD futures*, 1993, vol. 93, pp. 37-54.
[41] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Transactions on Graphics (TOG),* vol. 34, no. 4, p. 98, 2015.
[42] Y. Pu *et al.*, "Variational autoencoder for deep learning of images, labels and captions," in *Advances in neural information processing systems*, 2016, pp. 2352-2360.
[43] Q. Yu, Y. Yang, Y.-Z. Song, T. Xiang, and T. Hospedales, "Sketch-a-net that beats humans," *arXiv preprint arXiv:1501.07873,* 2015.
[44] P. Karimi, M. L. Maher, N. Davis, and K. Grace, "Deep Learning in a Computational Model for Conceptual Shifts in a Co-Creative Design System," *arXiv preprint arXiv:1906.10188,* 2019.
[45] D. Ha and D. Eck, "A neural representation of sketch drawings," *arXiv preprint arXiv:1704.03477,* 2017.
[46] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114,* 2013.
[47] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing,* vol. 45, no. 11, pp. 2673-2681, 1997.
[48] S. Reed *et al.*, "Parallel multiscale autoregressive density estimation," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 2912-2921: JMLR. org.
[49] R. E. Bellman, *Adaptive control processes: a guided tour*. Princeton university press, 2015.
[50] Y. Chen, S. Tu, Y. Yi, and L. Xu, "Sketch-pix2seq: a model to generate sketches of multiple categories," *arXiv preprint arXiv:1709.04121,* 2017.
[51] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*, 2016, pp. 478-487.
[52] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research,* vol. 9, no. Nov, pp. 2579-2605, 2008.
[53] H. R. J. Jongejan, T. Kawashima, J. Kim, and N. Fox-Gieg, "The Quick, Draw! - A.I. Experiment.," 2016.
[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.
[55] F. Chollet. (2015). *Keras*. Available: https://github.com/fchollet/keras
[56] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval research logistics quarterly,* vol. 2, no. 1‐2, pp. 83-97, 1955.
[57] S. M. Kosslyn, *Image and brain: The resolution of the imagery debate*. MIT press, 1996.

