Sketchforme: Composing Sketched Scenes from Text Descriptions for Interactive Applications

Forrest Huang

University of California, Berkeley
Berkeley, U.S.A.
forrest_huang@berkeley.edu

John F. Canny

University of California, Berkeley
Berkeley, U.S.A.
canny@berkeley.edu

Figure 1: Sketchforme synthesizes sketched scenes corresponding to users' text descriptions to support interactive applications.

ABSTRACT

Sketching is an effective communication medium that augments and enhances what can be communicated in text. We introduce Sketchforme, the first neural-network-based system that can generate complex sketches based on text descriptions specified by users. Sketchforme's key contribution is to factor complex sketch rendering into layout and rendering subtasks using neural networks. The sketches composed by Sketchforme are expressive and realistic: we show in our user study that these sketches convey descriptions better than human-generated sketches in several cases, and 36.5% of those sketches were identified as human-generated. We develop some interactive applications using these generated sketches, and show that Sketchforme can significantly improve language learning applications and support intelligent language-based sketching assistants.

Author Keywords

interactive applications; natural language; sketching; generative models; deep learning; neural networks; Transformer; interactive machine learning

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UIST '19, Oct 20–23, 2019, New Orleans, LA, USA

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM 978-1-4503-6816-2/19/10 \$15.00

<https://doi.org/10.1145/3332165.3347878>

CCS Concepts

•Human-centered computing → Natural language interfaces; •Computing methodologies → Neural networks; Computer vision tasks;

INTRODUCTION

Sketching is a natural and effective way for people to communicate artistic and functional ideas. Sketches are *abstract* drawings widely used by designers, engineers, and educators as a thinking tool to materialize their vision while discarding unnecessary details. Sketching is also a popular form of artistic expression among amateur and professional artists. With the pervasive use of sketches across diverse fields, researchers in the HCI and Graphics communities developed sketch-based interactive tools to enable intuitive and rich user experiences, such as assisted sketching systems [10, 16], design tools for prototyping [14], and animation authoring tools [3].

As sketch-based user interactions pose numerous benefits, researchers have created interactive systems that couple abstract graphics with semantically meaningful text to augment user interactions across various domains, such as integrating verbal text descriptions into sketch-based UI prototyping tools [15] and synthesizing abstract icons from file names [18]. Motivated by these systems, we believe an automated system that produces a diverse set of sketches from natural language could further encourage the adoption of sketches for various applications. This system could help users rapidly generate sketches while maintaining control over the sketches' content naturally using text descriptions. These interactions could be useful for domains such as language learning and communication, where sketching can be time-consuming for users.

Recent advances in neural-network-based generative models drastically increased machines' ability to generate convincing graphical content, including sketches, from high-level concepts. The Sketch-RNN model [7] demonstrates that recurrent neural networks (RNNs) trained on crowd-sourced data can generate original sketches from various concept classes. Using these techniques, we introduce *Sketchforme*, the first system that is capable of synthesizing complex sketches conditioned on natural language descriptions.

The contribution of this paper is two-fold. First, we contribute *Sketchforme*, the system that uses a novel, automated two-step neural method for generating sketched scenes from text descriptions. *Sketchforme* first uses its *Scene Composer*, a neural network that learned *high-level composition principles* from datasets of human-annotated natural images that contain text captions, bounding boxes of individual objects, and class information of the objects, to generate composition layouts of the scenes. *Sketchforme* then uses its *Object Sketcher*, a neural network that learned *low-level sketching mechanics* to generate sketches adhering to the objects' aspect ratios in the compositions. Finally, *Sketchforme* composes these generated objects of certain aspect ratios into meaningful sketched scenes.

Second, we build and evaluate several applications, including a sketch-based language learning system and an intelligent sketching assistant. These applications illustrate the potential value of *Sketchforme* in supporting novel sketch-based interactions (Section 6). In these applications, *Sketchforme* creates new interactions and user experiences with the interplay between language and sketches. We envision *Sketchforme* to further enable engaging sketch-based human-computer interactions in various domains by allowing users to create expressive sketches interactively.

RELATED WORK

Assisted Sketching Tools and Tutorials

Prior works have augmented the creative process of sketching with automatically-generated and crowd-sourced drawing guidance. ShadowDraw [16] and EZ-sketching [21] used edge images traced from natural images to suggest realistic sketch strokes to users. The Drawing Assistant [10] extracts geometric structure guides to help users construct accurate drawings. PortraitSketch [25] provides sketching assistance specifically on facial sketches by adjusting geometry and stroke parameters. Researchers also developed crowd-sourced web applications to provide real-time feedback for users to correct and improve sketched strokes [19].

In addition to assisted sketching tools, researchers also developed sketching tutorial systems to improve users' sketching proficiency. How2Sketch [8] automatically generates multi-step tutorials of sketching 3D objects. Sketch-sketch revolution [4] provides first-hand experiences created by sketch experts for novice sketchers.

While these methods help users create refined sketches, none of them possess *Sketchforme*'s ability to synthesize sketches from text descriptions directly.

Neural Sketch Generation Model

Sketchforme builds upon the Sketch-RNN model, the first neural-network based sketch generation model [7]. Sketch-RNN is a sequence encoder-decoder model that can unconditionally generate stroke-based sketches based on object classes, or conditionally reconstruct sketches based on users' input sketches. *Sketchforme* extends Sketch-RNN's model for sketching individual objects to support sketch generation based on various aspect ratios of the objects in the composition layouts.

Neural Text-to-Image Synthesis

Generating graphical content from text description is a popular ongoing research problem. Recent works on Generative Adversarial Networks (GANs) [20, 26] show promising results in generating realistic images from text descriptions. GAN-CLS [20] augments the GAN architecture to consider text descriptions, and subsequently generate images based on users' text input. Extending on these works, [9] introduces multiple components to first synthesize composition and outlines from text descriptions, and subsequently generate images from these compositions and outlines. This is similar to *Sketchforme*'s multi-step approach to generate complete sketched scenes from natural language except in the domain of natural images.

Neural Style Transfer

Several prior works in the Computer Vision community focus on the research problem of transferring styles between visual content. These prior works explore image stylization by matching statistics of feature maps (i.e., filters) of pre-trained models [5] or using generative adversarial networks [24, 27]. One possible approach for sketch generation that arises from these techniques is to combine the two approaches of neural style transfer and text-to-image synthesis: stylizing synthetic images generated based on text descriptions. However, this approach likely results in realistic, detailed sketch-style images which contain distracting artifacts. *Sketchforme* focuses on synthesizing *abstract sketched scenes from scratch* that capture fundamental ideas from messages communicated by the scenes.

SYSTEM DESCRIPTION

To support applications that afford sketch and natural-language based interactions, we developed *Sketchforme*, the system that provides the core capability of synthesizing sketched scenes from natural language descriptions. *Sketchforme* implements a two-step approach to generate a complete scene from text descriptions as illustrated in Figure 2. In the first step, *Sketchforme* uses its *Scene Composer* to generate composition layouts represented by bounding boxes of individual objects. These bounding boxes dictate locations, sizes, and aspect ratios of objects in the scene. *Sketchforme*'s *Object Sketcher* then uses this information at the second step of the generation process to generate specific sketch strokes of these objects in their corresponding bounding boxes. These steps reflect a fundamental process suggested in many sketching tutorials, where the overall composition of the scene is drafted before filling in details that characterize each object [2].

Figure 2: Overall system architecture of Sketchforme which consists of two steps in its sketch generation process.

By taking this two-step approach, Sketchforme is able to model high-level object relations critical to composing the scenes, enabling a multitude of applications that require such information. Moreover, this approach overcomes the difficulty for end-to-end sketch generation methods to capture global structures of sequential inputs [7]. End-to-end scene sketch generation also requires datasets of dedicated sketch-caption pairs that are difficult for crowd-workers to create [29].

Scene Composer: Generating Composition Layouts

To generate composition layouts of scenes, we first model composition layouts as a sequence of n objects (and start/end tokens), such that each object generated by the network is represented with 8 values:

$$b_t = [x_t, y_t, w_t, h_t, l_t, \text{box}_t, \text{start}_t, \text{end}_t], t \in [1, n]$$

The first 5 values are fundamental data that describes bounding boxes of objects in the scene: x-position, y-position, width, height, and the class label. The last three values are boolean flags used as extra ‘tokens’ to mark the actual objects, the beginning of sequences and the end of sequences.

Using this sequential encoding of scenes, we designed a Transformer-based Mixture Density Network as our *Scene Composer* to generate realistic composition layouts. Transformer Networks [22] are state-of-the-art neural networks for sequence-to-sequence modeling tasks, such as machine translation and question answering. We use a Transformer Network to perform a novel task: generating a sequence of objects from a text description c , a sequence of words. As multiple scenes can correspond to the same text descriptions, we feed the outputs of the Transformer Network into Gaussian Mixture Models (GMMs) to model the variation of scenes, forming a Mixture Density Network [1].

The generation process of the composition layouts involves taking the previous bounding box b_{t-1} (or the start token) as an input and generating the current box b_t . At each time-step, the Transformer Network generates an output t_t conditioned on the text input c and previously generated boxes $b_{1...t-1}$ using self-attention and cross-attention mechanisms built into the architecture. This process is repeated for multiple bounding boxes until an end token is generated:

$$t_t = \text{Transformer}([b_{1...t-1}; c]) \quad (1)$$

t_t is then projected to the appropriate dimensionality to parameterize the GMMs with various projection layers W_{xy} and W_{wh} to model $P(x_t, y_t)$, the distribution of the bounding boxes’ positions, and $P(w_t, h_t)$, the distribution of the bounding boxes’ sizes. Sketchforme can then generate bounding boxes $[x_t, y_t, w_t, h_t]$ by sampling from these distributions in Equations 2 and 3. The GMMs use the projected values as mean and co-variance parameters for mixtures of multi-variate Gaussian distributions M . These values are passed through appropriate activation functions (Sigmoid, exp and tanh) to comply with the required range of the parameters.

$$P(x_t, y_t) = \sum_{i \in M} \Pi_{1,i} \mathcal{N}(x_t, y_t | \mu_1, \Sigma_1),$$

$$[\Pi_1, \mu_1, \Sigma_1] = a(W_{xy}(t_t)) \quad (2)$$

$$P(w_t, h_t) = \sum_{i \in M} \Pi_{2,i} \mathcal{N}(w_t, h_t | \mu_2, \Sigma_2),$$

$$[\Pi_2, \mu_2, \Sigma_2] = a(W_{wh}([x_t; y_t; t_t])) \quad (3)$$

While $P(x_t, y_t)$ is modeled only from the first projection layer W_{xy} , we consider $P(w_t, h_t)$ to be conditioned on the position of the boxes similar to [9]. To introduce this condition, we concatenate t_t and $[x_t, y_t]$ as inputs to the second projection layer as described in Equation 3. The probability of each generated bounding box being an actual object or a start/end token are generated using a softmax-activated third projection layer W_c from the Transformer output:

$$P(\text{box}_t, \text{start}_t, \text{end}_t) = \text{softmax}(W_c t_t) \quad (4)$$

In addition, Sketchforme separately uses an LSTM to generate class labels l_t because the class labels given certain descriptions are assumed to not vary across examples. The full architecture of the Scene Composer is shown in Figure 3.

Object Sketcher: Generating Individual Sketches

After obtaining scene layouts from the Scene composer, we designed a modified version of Sketch-RNN model to generate individual objects in Sketchforme according to the layouts. We adopt the decoder-only Sketch-RNN that is capable of generating sketches of individual objects as sequences of individual strokes. Sketch-RNN’s sequential generation process involves generating the current stroke based on previously generated strokes, a method commonly used in sequence modeling tasks. Sketch-RNN also uses a GMM to model variation of sketch strokes.

While the decoder-only Sketch-RNN generates realistic sketches of individual objects in certain concept classes, the aspect ratios of the output sketches generated by the original Sketch-RNN cannot be constrained. Hence, sketches generated by the original Sketch-RNN may be unfit for assembling into scene sketches guided by the layouts generated by the Scene Composer. Further, naive direct resizing of the sketches

Figure 3: Model architecture of the Scene Composer used in the first step of Sketchforme’s sketch generation process.

can produce sketches of unsatisfactory quality for complex scenes.

We modified Sketch-RNN as the *Object Sketcher* that factors in the aspect ratios of objects when generating sketches. To incorporate this information in the Sketch-RNN model, we compute the aspect ratios of the training data and concatenate the aspect ratio $r = \frac{\Delta y}{\Delta x}$ of each sketch with the previous stroke as input to our modified Sketch-RNN in the sketch generation process as shown in Figure 4. The new formulation and output of the modified Sketch-RNN for t -th stroke is:

$$[h_t; c_t] = LSTM([S_{t-1}; h_{t-1}; r]), y_t = Wh_t + b_t \quad (5)$$

Since each Sketch-RNN model only handles a single object class, we train multiple modified Sketch-RNN models based on multiple classes and use appropriate models based on class labels in the layouts generated by the Scene Composer for assembling the final sketched scene.

After generating the strokes, the Object Sketcher converts them into SVG paths and fills each object in white using the non-zero rule. The object corresponding to the first generated bounding box generated by the Scene Composer is then composed as the foreground of the sketched scene, and subsequently generated objects are placed in the background in the order of generation.

MODEL TRAINING AND DATA SOURCES

Sketchforme’s Scene Composer and Object Sketcher are trained on different datasets that encapsulate visual-scene-level knowledge and sketching knowledge separately. This relaxes the requirement for Sketchforme to be trained on natural language annotated datasets of sketched scenes that provide varied scenes corresponding to realistic scene-caption pairs.

We trained the Scene Composer using the Visual Genome dataset [13], which contains natural language region descriptions and object relations of natural images, to demonstrate its

Figure 4: Model architecture of the extended Sketch-RNN in Sketchforme’s Object Sketcher.

flexibility in utilizing various types of scene-layout datasets. Object relations in the dataset each contains a ‘subject’ (e.g., ‘person’), a ‘predicate’ (e.g., ‘on’), and an ‘object’ (e.g., ‘car’) represented by class labels and bounding boxes of participating objects in the image. Natural language region descriptions are represented by bounding boxes of the regions and description texts that correspond to the regions. We reconcile these two types of information using region graphs in the dataset that pair these two types of data. With the paired data of natural language descriptions and relations, we train the Scene Composer to generate composition layouts. We selected relations that contain subsets of the 100 most commonly used object classes and 70 predicates in the dataset. This dataset of selected object classes and predicates contains 101,968 instances. We split this dataset in the scheme of: 70% training set, 10% validation set, and 20% test set.

The Object Sketcher is trained with the Quick, Draw! [12] dataset that contains 70,000 training sketches, 2,500 validation sketches and 2,500 test sketches for each of the 345 object categories in the dataset. As mentioned in Section 3, we preprocess the data by computing the aspect ratios of all sketches as inputs to the Object Sketcher in addition to the original stroke data.

Using these data sources, we train multiple neural networks of various configurations and loss functions in Sketchforme. The LSTM architectures in the Scene Composer for generating composition layouts is stacked with 2 hidden-layers of size 512. Similarly, the Transformer Network has the configuration $(d_{model}, N_{layers}) = (512, 6)$.

The Scene Composer is trained by minimizing the negative log-likelihoods of the position data L_{xy} and size data L_{wh} , and cross-entropy loss for categorical outputs L_P :

$$L_{xy} = - \sum_{i=1}^n \log(P(x_i, y_i)) \quad (6)$$

$$L_{wh} = - \sum_{i=1} P(\text{box}_i) \log(\text{box}_i) + P(\text{start}_i) \log(P(\text{start}_i)) + P(\text{end}_i) \log(P(\text{end}_i)) \quad (8)$$

For generating the class labels, each l_i is represented as a 100-dimensional vector in our model, with each value $l_{i,t}$ corresponding to the output probability of the class. L_{class} is thus computed as:

$$L_{\text{class}} = - \sum_{t=1}^n \sum_{i=1}^{100} l_{i,t} \log(l_{i,t}) \quad (9)$$

We combine these losses with weight hyper-parameters to obtain a general training objective L_{SC} for the Scene Composer:

$$L_{SC} = \lambda_1 L_{xy} + \lambda_2 L_{wh} + \lambda_3 L_p + \lambda_4 L_{\text{class}} \quad (10)$$

We set $\lambda_1 = 1.0, \lambda_2 = 1.0, \lambda_3 = 1 \times 10^{-5}, \lambda_4 = 1 \times 10^{-3}$. We used the Adam Optimizer with an initial learning rate of 1×10^{-5} and $\beta_1 = 0.9, \beta_2 = 0.999$ to minimize the loss function. We used 5 mixtures in each of the GMMs. We chose these hyper-parameters based on empirical experiments.

The Object Sketcher uses an HyperLSTM cell [6] of size 2048 for the modified Sketch-RNN model. The loss function of the Sketch-RNN model is identical to the reconstruction loss L_R in the original Sketch-RNN model to maximize the log-likelihood of the generated probability distribution for each of the strokes S_i . The model is trained with an initial learning rate of 0.0001 and gradient clipping of 1.0.

EXPERIMENTS AND RESULTS

Central to evaluating Sketchform’s success is assessing its effectiveness in generating realistic and relevant sketches and layouts from text descriptions. We evaluated the data generated by Sketchform at each step of the generation process qualitatively and quantitatively to demonstrate its effectiveness of generating sketched scenes. We further conducted two user studies on the overall utility of the generated sketches to explore their potential in supporting real-world applications.

Composition Layout Generation

The composition layouts generated by the Scene Composer in the first step of Sketchform’s sketch generation process are represented as bounding boxes of individual objects in the scene. While the Scene Composer already directly maximizes the log-likelihood of the data, we can evaluate the performance of the model by visualizing and comparing heat-maps created by super-positioning instances of real data and generated data.

Because Sketchform considers the text input when generating the composition layouts, we should only compare the generated bounding boxes with ground-truth bounding boxes from the dataset that are relevant to the text input. We obtain these ground-truth compositions by filtering the subjects, objects,

and predicates based on the descriptions. For instance, the composition layouts generated from ‘a person riding a horse.’ are compared to all ground-truth layouts with ‘person’ subjects, predicates that are related to riding such as ‘on’, ‘on top of’ etc. and ‘horse’ objects.

Heat-maps in Figure 5 shows the distributions of Sketchform-generated bounding boxes and ground-truth bounding boxes from the dataset. From these heat-maps, we can obtain a holistic view of the generation performance of the model by visually evaluating the similarity between the heat-maps. We observe similar distributions between the ground-truth layouts and the generated layouts based on all of the descriptions.

We can further approximate an overlap metric between the distributions using Monte-Carlo simulations to evaluate the model’s performance quantitatively. To estimate the degree of overlap between the generated data distribution and the dataset’s distribution, we generated 100 composition layouts for each description and randomly sampled 1000 data points within each bounding boxes in these layouts. We estimate the overlap between the distributions by counting the number of data points that lie within the intersections between any generated and ground-truth bounding boxes. We compare Sketchform’s performance with both a heuristic-based bounding box generator and a naive random bounding box generator. The heuristic-based bounding box generator only generates the first bounding boxes above/below the second bounding boxes for descriptions with the above-related/below-related predicates. The random bounding-box generator samples random values that describe the bounding boxes from uniform distributions, which serves as a naive baseline. Table 1 shows the percentage of the 1000 data points that lie in the intersections. The metric value for overlap between real and Sketchform-generated data is considerably higher than both the value for overlap between real and heuristic-generated data and the value for overlap between real and randomly generated data, which confirms our qualitative visual inspection of the heat-maps.

Description	Sketchform	Heuristics	Random
a dog on a chair	89.1%	64.4%	61.6%
an elephant under a tree	68.4%	40.3%	30.6%
a person riding a horse	94.0%	57.7%	51.5%
a boat under a bridge	31.8%	15.0%	6.85%

Table 1: Overlap metric from Monte-Carlo simulations for each description between real data and Sketchform-generated/heuristics-generated/random data.

Generating Individual Object Sketches at Various Aspect Ratios

The main addition of Sketchform to the original Sketch-RNN model is a new input that allows the Object Sketcher to generate sketches based on target aspect ratios ($r = \frac{\Delta y}{\Delta x}$) of completed sketches. We evaluate this approach by generating sketches of various aspect ratios. The Object Sketcher is able to adhere

Figure 5: Heat-maps generated by super-positioning Sketchforme-generated/Visual Genome (ground-truth) data. Each horizontal pair of heat-maps corresponds to an object from a description.

to input aspect ratios and generate individual object sketches coherent to the ratios. As shown in Figure 6, sketched trees generated with ratio $r = 1.0$ can be perceived as shorter than those generated with $r = 2.0$.

Complete Scene Sketches

Combining the Scene Composer and the Object Sketcher, Sketchforme generates complete scene sketches directly from text descriptions. Several examples of the sketches are shown in Figure 1 and Figure 7. In these figures, sketches that correspond to ‘a boat under a bridge’ consist of small boats under bridges, whereas sketches that correspond to ‘an apple on a tree’ consist of small apples on large trees that follow the actual sizes and proportions of the objects. Moreover, Sketchforme is able to generalize to novel concepts of ‘a cat on top of a horse,’ such that the only relations involving a cat and a horse in the Visual Genome dataset which the model was trained on correspond to ‘a horse facing a cat.’ The sizes of cats and horses in these sketches are in proportion to their actual sizes, and the cat is adequately placed on the back of the horse.

We further trained Sketchforme on the Abstract Scenes dataset [28] to evaluate the approach’s generalizability on handling complex scenes. We included examples of sketches generated by Sketchforme based on a) ‘Some squirrels near the pond under the trees on a sunny day’ (a multi-object multi-relation scene) and b) ‘Living room with some paintings on the wall’

Figure 6: Generated sketches of trees with various aspect ratios by the Object Sketcher in Sketchforme.

Figure 7: Complete scene sketches generated by Sketchforme. Sketchforme is able to generalize to novel concepts such as ‘a cat on top of a horse.’

(a scene consists of an abstract setting, i.e., living room) in Figure 8.

Sketchforme was able to generate these scenes adhering to the captions: for description a), Sketchforme was able to locate each object according to inter-object relations dictated by the description; for description b), Sketchforme was able to analyze the phrase ‘living room’ and generate couches and house plants in the scenes. When Sketchforme needs to handle a large number of objects, we found that it can be further improved by making a minor modification to the Scene Composer: feeding the generated classes into the Transformer network to synchronize the generated bounding boxes with the generated class labels of the objects. Moreover, we observed that the limitation of occlusions (further discussed in Section 7.1) is more apparent as the scenes become more crowded with objects, such as the multiple overlapping trees in scenes based on description a).

Figure 8: Complete scene sketches generated by Sketchforme-trained on the Abstract Scenes dataset that contains complex multi-object scenes.

Human Perception User-Study

Sketchforme's high-level goal is to augment users' communication and learning processes by generating realistic, plausible, and coherent sketches for users to interact with. To complement the quantitative and qualitative evaluation of the sketches, we conducted a user study on Amazon Mechanical Turk (AMT) to gauge human subjects' opinions on the sketches' realism and ability to convey the descriptions used to generate them.

Study Procedure

We recruited 51 human subjects on AMT and asked them to each review 50 sketches generated by either humans or Sketchforme. These 50 sketches are generated from five descriptions. The human-generated sketches are obtained from another AMT task prior to this user study based on Quick, Draw! [12]. These human-generated sketches are shown in Figure 9. In this study, subjects are provided with complete sketched scenes and descriptions that the scenes are based on. Subjects are required to respond to the following questions:

1. Do you think this sketch was generated by a computer (AI) or a human?
2. On a scale of 1-5 (1 represents that description conveyed very poorly, 5 represents that description conveyed very well), how well did you think the message is conveyed by the sketch?

The subjects are given 10 sketches as trial questions with answers to the first question at the beginning of the task. After completing the trial tasks, the subjects' answers to the remaining 40 sketches are aggregated as the study results. This study protocol is similar to perception studies commonly used to evaluate synthetic visual content generation techniques in the deep learning community [11]. In addition, we collected comments from the users (if any) and their perceived overall difficulty of the task at the end of the task.

Results

The first question probes the realism of the sketches with a Turing-test-style question asking the subjects to determine

Figure 9: Samples of sketches produced by humans and Sketchforme used in the AMT user study.

whether the sketches are created by humans. As shown in Figure 10, subjects on average considered 64.6% of the human-generated sketches as generated by humans, while they considered 36.5% of Sketchforme-generated sketches as generated by humans. Although the percentage of Sketchforme-generated sketches considered as generated by humans is significantly lower ($p < 1.05 \times 10^{-10}$, paired t-test) than that of human-generated sketches, individual participants commented in the study that it was difficult to distinguish between human-generated and Sketchforme-generated sketches. P2 mentioned that they "really couldn't tell the difference in most images." P6 commented that they "didn't know if it was human or a computer (that generated the sketches)." These results demonstrate the potential for Sketchforme in generating realistic sketched scenes.

We hypothesize one of the possible reasons for the lower percentage of Sketchforme-generated sketches to be considered as human-drawn is that the curves of the synthetic sketches are in general less jittery than human-drawn sketches. We suggest future work explore introducing stroke variation to generate more realistic sketches.

The results for the second question reflects the ability of the sketches to communicate the underlying descriptions that they are based on. The average score for human-generated sketches is $\mu = 3.46$, whereas the average score for Sketchforme-generated sketches is $\mu = 3.21$ as shown in Figure 11. Although Sketchforme-generated sketches achieved lower scores overall, Sketchforme-generated sketches achieved statistically better average scores for sketches based on two of the descriptions: 'a boat under a bridge' and 'an airplane in front of a mountain' ($p < 0.0005$, paired t-test). There is also no significant difference between the scores of human/Sketchforme-generated sketches based on 'a cat on top of a horse.' This shows the competitive performance of Sketchforme-generated sketches in communicating the underlying descriptions for some scenes.

Figure 10: Percentage of sketches considered by users as human-generated. On average, 64.6% of human-generated sketches and 36.5% of Sketchforme-generated sketches are perceived as human-generated.

Sketch Interpretation User-Study

To further evaluate Sketchforme's ability to deliver messages through sketches, we conducted an exploratory user-study to gauge users' ability to translate Sketchforme-generated sketches back into natural language descriptions.

Study Procedure

We recruited 10 participants on AMT. Each participant was provided with 5 sketches generated by humans and 5 sketches generated by Sketchforme from the same descriptions, but without the original text descriptions. The sketches are reused from the previous study, but the order and selection of the sketches from the set of sketches in the previous study are randomized for each participant. Participants are then asked to produce text descriptions that represent the meaning of each of the sketches. We ensured that the participants had not participated in any of our other studies.

Results

We aggregated the text descriptions produced by each of the participants and compared the subjects, objects, and predicates included in their text descriptions with the original descriptions used to generate the sketches. The percentage of user-generated descriptions that match the subjects/objects/predicates of the original descriptions are respectively 86%/72%/46% for human-generated sketches, and 86%/76%/38% for Sketchforme-generated sketches. While we did not observe any significant differences between the communication ability of both sets of sketches ($p > 0.269$, each paired t-test for subjects/objects/predicates), one novel insight brought by this study is that these sketches are particularly weak in conveying the predicates. Multiple descriptions provided by the users on both sets of sketches did not mention the predicates at all, such as 'an elephant and a tree,' or mentioned more general predicates, such as 'elephant walking near a tree,' for sketches based on 'an elephant under the tree.' Effectively

Figure 11: Average score for conveying the sketches' descriptions. Sketchforme-generated sketches perform better than human-generated sketches for sketches of 'a boat under a bridge.' and 'an airplane in front of a mountain.'

conveying the descriptions' predicates through sketches can be an interesting issue for further research investigation.

APPLICATIONS

In this section, we explore several applications that can benefit from Sketchforme's ability to generate compelling sketches from natural language descriptions.

Sketch-Assisted Language Learning

Sketches have been shown to improve memory [23]. As language learning is a memory-intensive task, Sketchforme could support language education applications based on sketches. These sketches can potentially create engaging and effective learning processes and avoid rote learning.

Language Learning Application

To explore the possibility of Sketchforme in supporting language learning, we built a basic language-learning application that aims to educate learners with a translation task from German to English. In this application, learners are presented with German phrases and asked to translate them to English in the form of multiple-choice questions similar to the process of learning term definitions from flash-cards. This application also implements the Leitner system [17] with three bins that repeats phrases which learners make the most mistakes on most frequently. Under this system, the phrases are moved to different bins depending on the participants' familiarity with the translations.

We gathered 10 pairs of German-English sentences from a native German speaker and form 2 sets of 5 translations each. In addition, deceptive English sentences are added as other choices in the multiple-choice test to be selected by the learners in the application. We deployed this application on AMT to test the improvement of learning performance by presenting

Sketchforme-generated sketches along with the phrases. The UI of the full application with a sketch presented to the users is shown in Figure 12.

Figure 12: User interface of the language-learning application powered by Sketchforme. The application augmented with sketches significantly reduced the time taken for users to achieve similar learning outcomes.

Study Procedure

The study consists of a training phase and a test phase for each participant. In the training phase, participants are presented with correct answers after answering each question. The participant can only advance to the next phase when they answer all questions correctly consecutively for all translations according to the Leitner system. In the test phase, participants are given one chance to provide their answer to all translations without seeing the correct answers. The participants are divided into two conditions, with the 'control' group only receiving phrases on their interface during training, and the 'treatment' group that receives both phrases and sketches generated by Sketchforme on their interface during the training phase. Both groups receive only the phrases on their interface during the test phase. Moreover, we use our two sets of translations for training and test phases alternatively, such that the participants will not get consecutive training and test phases for the same set of descriptions.

The performance of the participants during the study are monitored with multiple analytical metrics, including completion time of each phase, and scores in the test phase, etc. At the end of the study, we also provide surveys for them to rate the difficulty of the task and the usefulness of the sketches (if applicable) on five-point Likert scales, and ask them to provide any additional suggestions to the interface.

Results

We recruited 38 participants on AMT to participate in the study. While we did not see significant differences ($p = 0.132$, unpaired t-test) in the correctness of answers in the test phase of the phrases between the 'control' and 'treatment' groups of participants, we discovered that the *time taken to complete the learning task* for the 'treatment' group (246 seconds on average) was significantly less ($p = 0.011$, unpaired t-test) than the control group (338 seconds on average) as shown in Figure 13. The 'treatment' group also generally found the sketches to be helpful for learning (rated 4.58 out of 5 in the post-study survey).

Figure 13: Average time taken to complete the language learning task among groups. With Sketchforme-generated sketches, participants take significantly less time to complete the study.

As Sketchforme is an automated system that is capable of generating sketches from free-text descriptions, and with these promising results on sketch-assisted language learning, we envision Sketchforme to support and improve large-scale language learning applications in the future.

Intelligent Sketching Assistant

Since Sketchforme uses sequence models and a multi-step generation process to generate sketches, by design it can support interactive human-in-the-loop sketching systems. To demonstrate such capability, we built a prototype of an intelligent sketching assistant reflective of two potential use-cases:

Auto-completion of scenes

As Sketchforme's Scene Composer consists of the Transformer Network, a sequence model that attends to previous objects in the scene to generate the upcoming object, we can complete unfinished user scenes instead of starting with a blank canvas by starting the generation with both the start token b_1 and an existing object in the scene created by the user b_2 . Figure 14 shows examples of Sketchforme completing users' sketch of a horse in step a) by adding potential sketched trees involved in the scene.

User-Steerable generation

Sketchforme's Scene Composer is capable of generating multiple potential candidates sketched objects at each step of composing the final sketched scenes. As such, users can select their preferred scene layout from the candidates. Figure 14 shows multiple candidates proposed by Sketchforme based on a text description in step b). Moreover, since Sketch-RNN is also capable of generating a variety of sketches, the users can select their preferred sketches of each individual objects in the scene.

LIMITATIONS

Occlusions and Layer Order

Sketchforme is trained to model scene compositions from a natural image dataset. In natural images, objects might occlude each other, hence affecting sizes and positions of the bounding boxes in the composition layouts. Figure 15a shows several boats that were inadequately placed in front of parts of the bridges that should have occluded the boats. To overcome this limitation, future systems can augment Sketchforme by including advanced vision models to determine the objects' layer order in the original natural image. The current Sketchforme system only considers a naive layer order determined by the generation sequence of the composition layout, which is 'subject' then 'object' from the dataset.

Figure 14: Intelligent sketching assistant powered by Sketchforme. The user can a) create partial sketch for Sketchforme to suggest multiple candidates for b) them to choose the adequate sketch they prefer from the description ‘a horse under a tree.’

Moreover, novel methods should be developed to handle overlapping sketched objects generated from occluded compositions. For instance, the model that generates composition layouts could enforce constraints to avoid overlaps in the sketches or follow user-specified heuristics to handle overlaps.

Aspect Ratios might be Weak Signals for Object Poses

Sketchforme uses aspect ratios of bounding boxes as the primary signal to inform the shapes of sketches of individual objects. Although these shapes can be sufficient to determine the correct poses for objects in some classes, such as the ‘tree’ class, merely constraining the shapes might be weak signals for objects of other classes. These shapes can suggest incoherent perspectives or incomplete sketches, such as examples shown in Figure 15b. In Figure 15b, only faces of the elephants were sketched due to aspect ratios provided to the Object Sketcher, which is inappropriate for composing sketched scenes. To mitigate this limitation, future work could model the poses of objects in sketches and natural images more closely using other cues, such as complete masks of the objects.

Figure 15: Limitations of Sketchforme’s sketch generation process. In a), the boats are significantly occluded by the bridges. In b), the elephants were represented with square bounding boxes which guided the system to sketch only the faces of the elephants.

FUTURE WORK

Conversational Sketch Suggestion and Tutorial System

To fully reap the benefits of the highly interactive media of sketching and natural language, and the high-level and low-level information of each element of the sketched scenes modelled by Sketchforme, we believe future work should expand beyond generating a single sketched scene from a single text description, and explore conversational interfaces that guide users to progressively create sketches coherent to dialogues. Moreover, since both components involved in Sketchforme’s

sketch generation process are capable of completing partial sketches created by users, Sketchforme can suggest possible strokes following incomplete user sketches for creating sketching tutorials.

Educational Applications in Other Domains

The interplay between natural language and sketches embodied by Sketchforme creates possibilities of building new applications that utilize the interactive properties of sketching and language. In this paper, we explored the capability of Sketchforme in supporting basic language learning. We believe future work could explore educational applications in other domains such as science and engineering education as text-annotated sketches are frequently used in these domains.

Coloring and Animation

The sketches generated by Sketchforme are binary sketch strokes without colors or animations. Future work should explore colored and/or animated sketches to enable richer user experiences. For instance, the natural image dataset that was used to train Sketchforme’s Scene Composer can be used to determine possible colors of the sketched objects.

CONCLUSIONS

This paper presents Sketchforme, a novel sketching system capable of generating abstract scene sketches that involve multiple objects based on natural language descriptions. Sketchforme adopts a novel two-step neural-network-based approach: the *Scene Composer* models *high-level* layouts of sketched scenes, and the *Object Sketcher* models *low-level* sketching mechanics of individual objects. These models can be trained without text-annotated datasets of sketched scenes. In the user study evaluating the expressiveness and realism of sketches generated by Sketchforme, human subjects considered Sketchforme-generated sketches more expressive than human-generated sketches for two of the five seeded descriptions. They also considered 36.5% of these sketches to be generated by humans. The sketches generated by Sketchforme significantly improved a sketch-assisted language learning system and enabled compelling intelligent features of a sketching assistant. Sketchforme possesses the potential to support interactive applications that utilize both sketches and natural language as interaction media, and afford large-scale applications in sketching, language education and beyond.

ACKNOWLEDGEMENT

The authors would like to thank Eldon Schoop, David M. Chan, Philippe Laban, Andrew Head, Daniel Seita, David Ha, Roshan Rao, Jeremy Warner, Jinkyu Kim, and all anonymous reviewers for their valuable feedback and comments on the paper and the project. The authors would also like to thank Patrick Barin for providing the German phrases in the language learning application, and participating workers on Amazon Mechanical Turk for completing the studies.

REFERENCES

- [1] Christopher M Bishop. 1994. *Mixture density networks*. Technical Report. Citeseer.
- [2] Jamie Combs and Brenda Hoddinott. 2011. *Drawing for dummies*. Wiley.

- [3] Richard C. Davis, Brien Colwell, and James A. Landay. 2008. K-sketch: A 'Kinetic' Sketch Pad for Novice Animators. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 413–422. DOI: <http://dx.doi.org/10.1145/1357054.1357122>
- [4] Jennifer Fernquist, Tovi Grossman, and George Fitzmaurice. 2011. Sketch-sketch Revolution: An Engaging Tutorial System for Guided Sketching and Application Learning. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 373–382. DOI: <http://dx.doi.org/10.1145/2047196.2047245>
- [5] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. A Neural Algorithm of Artistic Style. *CoRR* abs/1508.06576 (2015). <http://arxiv.org/abs/1508.06576>
- [6] David Ha, Andrew M. Dai, and Quoc V. Le. 2016. HyperNetworks. *CoRR* abs/1609.09106 (2016). <http://arxiv.org/abs/1609.09106>
- [7] David Ha and Douglas Eck. 2018. A Neural Representation of Sketch Drawings. In *ICLR 2018*. <https://openreview.net/pdf?id=Hy6GHpkCW> 2018.
- [8] James W. Hennessey, Han Liu, Holger Winnemüller, Mira Dontcheva, and Niloy J. Mitra. 2017. How2Sketch: Generating Easy-To-Follow Tutorials for Sketching 3D Objects. *Symposium on Interactive 3D Graphics and Games* (2017).
- [9] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. 2018. Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7986–7994.
- [10] Emmanuel Iarussi, Adrien Bousseau, and Theophanis Tsandilas. 2013. The Drawing Assistant: Automated Drawing Guidance and Feedback from Photographs. In *ACM Symposium on User Interface Software and Technology (UIST)*. ACM, St Andrews, United Kingdom. <https://hal.inria.fr/hal-00840853>
- [11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).
- [12] Jonas Jongejan, Henry Rowley, Takashi Kawashima, Jongmin Kim, and Nick Fox-Gieg. 2016. The Quick, Draw! - A.I. Experiment. (2016). <https://quickdraw.withgoogle.com/>
- [13] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *Int. J. Comput. Vision* 123, 1 (May 2017), 32–73. DOI: <http://dx.doi.org/10.1007/s11263-016-0981-7>
- [14] James A. Landay. 1996. SILK: Sketching Interfaces Like Krazy. In *Conference Companion on Human Factors in Computing Systems (CHI '96)*. ACM, New York, NY, USA, 398–399. DOI: <http://dx.doi.org/10.1145/257089.257396>
- [15] Walter S. Lasecki, Juho Kim, Nick Rafter, Onkur Sen, Jeffrey P. Bigham, and Michael S. Bernstein. 2015. Apparition: Crowdsourced User Interfaces That Come to Life As You Sketch Them. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1925–1934. DOI: <http://dx.doi.org/10.1145/2702123.2702565>
- [16] Yong Jae Lee, C. Lawrence Zitnick, and Michael F. Cohen. 2011. ShadowDraw: Real-time User Guidance for Freehand Drawing. *ACM Trans. Graph.* 30, 4, Article 27 (July 2011), 10 pages. DOI: <http://dx.doi.org/10.1145/2010324.1964922>
- [17] S. Leitner and R. Totter. 1972. *So lernt man lernen*. Herder. <https://books.google.com/books?id=sD3AAQAACAAJ>
- [18] J. P. Lewis, Ruth Rosenholtz, Nickson Fong, and Ulrich Neumann. 2004. VisualIDs: Automatic Distinctive Icons for Desktop Interfaces. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 416–423. DOI: <http://dx.doi.org/10.1145/1015706.1015739>
- [19] Alex Limpaecher, Nicolas Feltman, Adrien Treuille, and Michael Cohen. 2013. Real-time Drawing Assistance Through Crowdsourcing. *ACM Trans. Graph.* 32, 4, Article 54 (July 2013), 8 pages. DOI: <http://dx.doi.org/10.1145/2461912.2462016>
- [20] Scott E. Reed, Zeynep Akata, Xinchun Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative Adversarial Text to Image Synthesis. *CoRR* abs/1605.05396 (2016). <http://arxiv.org/abs/1605.05396>
- [21] Qingkun Su, Wing Ho Andy Li, Jue Wang, and Hongbo Fu. 2014. EZ-sketching: Three-level Optimization for Error-tolerant Image Tracing. *ACM Trans. Graph.* 33, 4, Article 54 (July 2014), 9 pages. DOI: <http://dx.doi.org/10.1145/2601097.2601202>
- [22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.
- [23] Jeffrey D. Wammes, Melissa E. Meade, and Myra A. Fernandes. 2016. The drawing effect: Evidence for reliable and robust memory benefits in free recall. *The Quarterly Journal of Experimental Psychology* 69, 9 (2016), 1752–1776. DOI: <http://dx.doi.org/10.1080/17470218.2015.1094494> PMID: 26444654.

- [24] Lidan Wang, Vishwanath A. Sindagi, and Vishal M. Patel. 2017. High-Quality Facial Photo-Sketch Synthesis Using Multi-Adversarial Networks. *CoRR* abs/1710.10182 (2017). <http://arxiv.org/abs/1710.10182>
- [25] Jun Xie, Aaron Hertzmann, Wilmot Li, and Holger Winnemöller. 2014. PortraitSketch: Face Sketching Assistance for Novices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 407–417. DOI: <http://dx.doi.org/10.1145/2642918.2647399>
- [26] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 5907–5915.
- [27] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *CoRR* abs/1703.10593 (2017). <http://arxiv.org/abs/1703.10593>
- [28] C Lawrence Zitnick and Devi Parikh. 2013. Bringing semantics into focus using visual abstraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3009–3016.
- [29] Changqing Zou, Qian Yu, Ruofei Du, Haoran Mo, Yi-Zhe Song, Tao Xiang, Chengying Gao, Baoquan Chen, and Hao Zhang. 2018. Sketchyscene: Richly-annotated scene sketches. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 421–436.