CHARACTER GENERATION THROUGH SELF SUPERVISED VECTORIZATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÖKÇEN GÖKÇEOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COGNITIVE SCIENCE

FEBRUARY 2022

Approval of the thesis:

CHARACTER GENERATION THROUGH SELF SUPERVISED VECTORIZATION

submitted by **GÖKÇEN GÖKÇEOĞLU** in partial fulfillment of the requirements for the degree of **Master of Science in Cognitive Science Department, Middle East Technical University** by,

Date: 11.02.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Gökçen Gökçeoğlu

Signature :

ABSTRACT

CHARACTER GENERATION THROUGH SELF SUPERVISED VECTORIZATION

Gökçeoğlu, Gökçen
M.S., Department of Cognitive Science
Supervisor: Prof. Dr. Cem Bozşahin
Co-Supervisor: Assist. Prof. Dr. Emre Akbaş

February 2022, 40 pages

Humans learn visual concepts rapidly and flexibly from few samples. However, this kind of learning is a challenge for the prevalent machine learning methodologies. Currently, high performing deep learning models and algorithms depend on large amounts of data and they are task-specific. In this study, we focus on the generative aspects of visual concept learning in the domain of handwritten characters. We develop an unsupervised approach that can be generalized to multiple tasks using a small number of samples. We present a drawing agent that operates on stroke-level representation of images. At each time step, the agent first assesses the current canvas and decides whether to stop or keep drawing. When a 'draw' decision is made, the agent outputs a program indicating the stroke to be drawn. As a result, it produces a final raster image by drawing the strokes on a canvas, using a minimal number of strokes and dynamically deciding when to stop. We train our agent through reinforcement learning on handwritten character datasets for unconditional generation and parsing (reconstruction) tasks. We utilize our parsing agent for exemplar generation and type conditioned concept generation without any further training. We present successful

results on all three generation tasks and the parsing task. Crucially, we do not need any stroke-level or vector supervision; we only use raster images for training.

Keywords: human concept learning, reinforcement learning, self-supervised learning, vectorization

ÖZ

GÖZETİMSİZ ÖĞRENME İLE KARAKTER VEKTÖRİZASYONU

Gökçeoğlu, Gökçen
Yüksek Lisans, Bilişsel Bilimler
Tez Yöneticisi: Prof. Dr. Cem Bozşahin
Ortak Tez Yöneticisi: Dr. Öğr. Üyesi. Emre Akbaş

Şubat 2022 , 40 sayfa

İnsanlar görsel konseptleri hızlı ve esnek bir şekilde öğrenebilirler. Ancak, bu öğrenme biçimi önde gelen makine öğrenmesi metotları için bir zorluk oluşturmaktadır. Yüksek performanslı derin öğrenme modelleri ve algoritmaları büyük veriler kullanmakta ve genellikle tek bir amaca hizmet etmektedir. Bu çalışmada biz, görsel konsept öğrenmenin üretici yönünü el yazısı karakterler üzerinde inceledik. Birden fazla amaç için kullanılabilecek, az veri ile eğitilebilen bir gözetimsiz öğrenme yöntemi geliştirdik. Görüntülerin kalem darbeleri düzeyinde temsili üzerinde çalışan bir çizim etmeni sunduk. Etmen, her zaman adımında mevcut kanvası inceler ve devam etme veya durma kararını verir. Eğer etmen çizime devam etmeye karar verirse, kalem darbesini ifade eden bir program ortaya çıkarır. Sonunda minimum sayıda darbe kullanarak bir çizim üretmiş olur. Etmeni, gözetimsiz üretim ve ayrıştırma (darbelerle yeniden oluşturma) yapabilmesi için pekiştirmeli öğrenme ile el yazısı karakter veri setleri ile eğittik. Ayrıştırma yapabilen etmenimizi örnek üretimini ve koşullu karakter üretiminde kullandık. Sözü geçen üç üretici görevde ve ayrıştırma görevinde başarılı sonuçlar sunduk. Önemli bir nokta olarak, eğitim için vektör gözetimi veya

fırça darbelerinin bilgisini kullanmadık.

Anahtar Kelimeler: insanlarda konsept öğrenimi, pekiştirmeli öğrenme, gözetimsiz öğrenme, vektörizasyon

To my family

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to Assist. Prof. Emre Akbaş. His guidance, knowledge, and insight made this work possible. At every step of the research, he constantly supported and encouraged me whenever I ran into trouble or made progress. I feel lucky to work with him.

I would like to thank Prof. Cem Bozşahin and Assoc. Prof. Cengiz Acartürk for providing helpful feedback. Their discussions widened my perspective on this study.

I would like to thank Assoc. Prof. Barbaros Yet, Assist. Prof. Burcu Can Buğlalılar, and Assist. Prof. Umut Özge for agreeing to be on my thesis committee.

I thank Ahmet Burak Baraklı and Görkay Aydemir for their renderer implementation. Their help significantly sped up months of experiments.

I cannot thank my beloved Onur enough who has been there for me at every step of this journey with his joyful and motivating attitude. His constant support has been invaluable for this thesis.

Finally, I would like express my gratitude to my parents, Kezban and Candan, for their unconditional love, support and encouragement throughout my education and life.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xiii

LIST OF FIGURES xiv

LIST OF ABBREVIATIONS xvii

CHAPTERS

1 INTRODUCTION 1

1.1 Motivation and Problem Definition 1

1.2 Proposed Methods 3

1.3 Contributions of the Study 5

1.4 Organization of the Thesis 5

2 RELATED WORK 7

2.1 Universality in Writing Systems 7

2.2 Omniglot Challenge 8

2.3 Visual Turing Test 9

2.4 Image Generation by Vectorization — With Stroke Supervision . . . 10

2.5 Self Supervised Vectorization 10
2.6 Self Supervised Program Synthesis 11
3 METHOD 13
3.1 Unconditional Generation 14
3.1.1 Policy Network 14
3.1.2 Discriminator 16
3.2 Image Reconstruction by Parsing 16
3.3 Generating New Exemplars 17
3.4 Generating Novel Concepts from Type 17
4 EXPERIMENTS 21
4.1 Datasets and Implementation Details 21
4.2 Unconditional Generation 23
4.3 Image Reconstruction by Parsing 25
4.4 Generating New Exemplars 26
4.5 Generating Novel Concepts from Type 28
5 CONCLUSIONS 31

LIST OF TABLES

Table 4.1 Comparison of the FID scores for different models on the MNIST dataset. We report the mean and the variance of FID scores from 5 simulations with different weight initializations. 23

Table 4.2 Ablation experiments for the loss term in Equation 3.1. λ_1 represents the coefficient of the term that measures the alignment of the canvas with the cluster center. λ_2 represents the coefficiet of the term that measures the distance of the current canvas to the distribution obtained from the dataset in terms of 'on' pixels. 24

Table 4.3 Reconstruction quality. L2 distance between target and the reconstructed image. (ImageVAE is taken from (Reddy et al., 2021), it indicates a purely raster-based autoencoder.) 26

Table 4.4 For each alphabet in the Omniglot evaluation set, we present the number of strokes our agent used to reconstruct the given image vs. mean number of strokes obtained from human-labeled data. The stroke count for human-labeled data is calculated using the labels within the Omniglot dataset. 27

LIST OF FIGURES

Figure 3.1 Generator model. At each time step, the policy network receives a canvas and outputs two distributions for Bezier curve parameters and stop/continue decision. When the 'continue' decision is sampled, the resulting stroke is rendered and added to the final output. 13

Figure 3.2 Forward pass of unconditional generation. The process starts with an empty canvas. At $t = 0$, the policy network generates a random stroke. Then, at each time increment, a new stroke is added to the input canvas. This process continues recursively till a 'stop' action is sampled. Partially drawn canvas is passed through the discriminator to get a 'realness' score. 14

Figure 3.3 Image reconstruction process. Given an input image, the policy network generates a stroke at each time step. Sampled stroke is masked out of the input image and the process is repeated till a 'stop' decision is made. The final canvas is a superposition of generated strokes. 17

Figure 3.4 Conditional generation process. Exemplars are generated using the policy learned by reconstruction training. In order to eliminate 'unlikely' samples, the likelihood function defined by the policy is used, i.e., a likelihood threshold is used. To achieve the alphabet (type) conditional generation task, a stroke library is formed. With this library, a set of strokes randomly sampled at each time step and obtained a new canvas. In order to obtain good (or real looking) samples, the discriminator trained for unconditional generation is employed. 19

Figure 4.1 Our drawing agent can accomplish four different tasks. From left to right: it can generate novel characters, parse a given character into its strokes, generate new exemplars for a given character, and generate novel concepts (i.e. characters) given a type (i.e. alphabet). **Ours is the first stroke-based method to tackle all of the generation and parsing tasks in the Omniglot Challenge, without requiring any stroke-level supervision.** 21

Figure 4.2 Overall architecture of the policy network. The network consists of a CNN backbone and 3 MLP braches. Each convolutional layer is followed by Batch Normalization and ReLU activation. After the last residual block, resulting tensor is flattened. Between the layers of MLP branches, ReLU activation is used. Finally, via a softmax function the action distributions are obtained. 22

Figure 4.3 Quality of generated MNIST characters as training progresses (i.e. policy is updated) from left to right. 24

Figure 4.4 Omniglot unconditional samples. For randomly sampled generations, four closest samples (in terms of pixelwise L2 distance) from the training dataset are presented. 24

Figure 4.5 Randomly sampled unconditional generations for the Omniglot dataset. 25

Figure 4.6 MNIST reconstructions. For each sample on the left hand side of the columns, parsing processes are demonstrated. Colors represent the order of the strokes. (*pink*: first stroke, *green*: second stroke, *blue*: third stroke) 25

Figure 4.7 Omniglot reconstruction. For each sample on the left hand side of the columns, resulting reconstructions are demonstrated. 26

Figure 4.8 New exemplar generation. Given an unseen character from a new alphabet (highlighted in red boxes), the model generated 9 exemplars.These metrics are calculated over all examples generated for the test set. 28

Figure 4.9 LPIPS values for each alphabet in the test set calculated from sampled exemplars (a), SSIM and L2 values for each alphabet in the test set calculated from sampled exemplars (b). 29

Figure 4.10 Novel sample generation conditioned on a type. Given 10 characters from an alphabet, our model produced 20 new samples. 30

Figure 4.11 LPIPS values for each alphabet in the test set calculated from novel samples produced (a). There is a consistency between the results obtained from these models. L2-SSIM values for each alphabet in the test set calculated from novel samples produced (b). 30

LIST OF ABBREVIATIONS

DNN	Deep Neural Network
CNN	Convolutional Neural Network
RL	Reinforcement Learning
RNN	Recurrent Neural Network
VAE	Variational Autoencoder
LSTM	Long Short Term Memory
CSG	Constructive Solid Geometry
MLP	Multi Layer Perceptron
FID	Frechet Inception Distance
LPIPS	Learned Perceptual Image Patch Similarity
SSIM	Structural Similarity Index Measure
KL	Kullback–Leibler

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Humans have an impressive ability of storing and utilizing information through concepts. Concepts are mental representations that connect our present experiences with our past knowledge base (Murphy, 2004). The way humans represent concepts enables many functions: categorization, inference, regeneration and creation of novel concepts (Spivey et al., 2012). In this study, we focus on visual concepts, specifically handwritten characters from world alphabets. Learning and leveraging visual concepts rapidly and flexibly comes naturally to humans. Additionally, humans only require one or few instances of a concept in order to learn. Through observing people's generative abilities, one can study the aspects of concept learning and the richness of representations.

On the other hand, this kind of 'learning', i.e., generalize and utilize a concept from sparse data is a significant challenge for current machine learning methodologies. Here, we refer to the ability of achieving different tasks that are addressed by machine learning with the term 'generalization'. The prominent machine learning research often tackle different tasks with separate models/algorithms. The other problem is the models' dependence to large amounts of data. The datasets that stand on the core of deep learning research are composed of thousands of samples from each category (Antol et al., 2015; Bellemare et al., 2013; Deng et al., 2009; Johnson et al., 2017; T.-Y. Lin et al., 2014; Liu et al., 2015). This introduces a distinction between how humans learn and how human-level performing models learn.

In this study, we focus on these two problems in the context of handwritten character generation. Five years ago, Omniglot challenge was presented in order to address the gap between human intelligence and artificial intelligence (B. M. Lake et al., 2015a). With this challenge, a definition of learning that emphasizes 'concepts' and 'richness of representations', is offered. The way we represent concepts allows us to adopt a wide range of generative tasks and one-shot classification. The ultimate goal being to simulate how people learn, the research community has been encouraged to develop a single model that is able to achieve generative tasks and one-shot classification using sparse data. Here, handwritten characters are considered as cognitively natural, high-dimensional and simple enough to be an ideal testbed to test a computational model targeting this 'learning' problem (B. M. Lake, 2014).

We address the generative aspect of human-like learning: learning from sparse data,

model's adaptability to different generative tasks, sequential generation and the quality of generations. To this end, we tackle the generative tasks of the Omniglot challenge: parsing, generating exemplars, type-conditioned concept generation and unconditional generation.

We start with the question whether the compositionality and causality of character generation process in humans can be modeled with an unsupervised deep learning approach. We approach the challenge with the hypothesis that one needs to consider the properties of writing systems in order to develop compatible approaches in machine learning. This entails modeling the writing act itself, i.e, how we build characters and the properties of our writing systems need to be considered. Thus, we build a model that conforms with the following requirements:

- A writing system is composed of strokes. Hence, our model needs to use strokes while composing characters.
- The number of strokes in world alphabets are approximately 3 (Changizi & Shimojo, 2005). The probable hypothesis regarding the reason for this property is that human brain can store 3-4 objects in its visual short-term memory (Trick & Pylyshyn, 1994; Vogel et al., 2001). Therefore we need to encourage our model to form a character with as few actions as possible.
- Studies show there is a universality in how the strokes are combined and types of junctions that form a writing system (Changizi & Shimojo, 2005; Changizi et al., 2006; Dehaene & Dehaene-Lambertz, 2016; Pagliarini et al., 2017) . The ultimate goal of our novel character generation models (type-conditioned and unconditional character generation) is to capture the configuration pattern of writing systems. This determines if the outputs of our model is real-looking.
- Humans utilize the topological properties of object during recognition (Changizi et al., 2006). While the geometrical features of visual concepts can easily change, the topological properties remain the same. This also reflects on how we recognize characters. The main implication of this is we can recognize a character written in different fonts, but we can easily differentiate the similar characters such as E and F. With our exemplar generation approach (generating variants of a character), we try to model this property.
- There is a certain stroke order in how people form characters. This phenomenon is analyzed in an empirical study carried out by Lake et al. (B. Lake et al., 2012a), where subjects are asked to reproduce novel handwritten characters. It is shown that while subjects can easily reproduce a given character; they also use very similar motor strokes during drawing. Our purpose is that the parsing model complies with this feature.

The main purpose of this study is to obtain a generative model that can solve the Omniglot tasks. Indeed, we cannot make an inference regarding how people learn, but we consider that it still has a value for cognitive science. Here, we present two supporting arguments:

- Deep neural networks (DNN) can be used as scientific models for studies on cognitive science.

- It is required to understand and use the properties of human visual cognition to obtain an algorithm that can reason and learn from sparse data.

The initial argument is thoroughly investigated by Cichy et al. (Cichy & Kaiser, 2019). DNNs can be starting points for new theories, that is, we may use them as a way of exploration. Initially, DNNs offer a neutral analogy between our cognitive functionalities. A model that is open to exploration may test the viability of a given hypothesis and cause shifts in the accepted concepts. To exemplify this, two-stream visual brain hypothesis is reconsidered when it is realised that DNNs have stronger object-categorical properties in the higher layers (Peterson et al., 2016). Secondly, DNNs can trigger further exploration if a successful approach is achieved.

The latter argument states the following: It is clear that the huge gap between state-of-the-art learning models and human learning should be addressed in order to invent more efficient and interpretable algorithms. The constraints and priors that humans have, made it possible to learn efficiently and flexibly. The process of discovering more efficient models may be a step towards understanding and modeling human concept learning.

1.2 Proposed Methods

An image can be stored and displayed using pixels or vector definitions. While raster images are composed of pixels, vector images are rendered using mathematical formulas that define lines and curves. Quality of a raster image depends on the number of pixels it is formed, the resolution of the image drops as the amount of pixels is decreased. Vector images, on the other hand, are not defined in terms of resolution. They can be scaled infinitely without a reduction in their quality.

Image generation problems are typically addressed by raster-based algorithms. The introduction of generative adversarial networks (GAN) (Goodfellow et al., 2014), variational autoencoders (VAE) (Kingma & Welling, 2013) and autoregressive models (Van Oord et al., 2016) has led to a variety of applications. Style transfer (Gatys et al., 2015; Isola et al., 2017), photo realistic image generation (Brock et al., 2018; Karras et al., 2019), and super resolution (Bin et al., 2017; Ledig et al., 2017) are some of the significant instances of the advancing field. Additionally, Hierarchical Bayesian models formulated by deep neural networks are able to use the same generative model for multiple tasks such as classification, conditional and unconditional generation (Edwards & Storkey, 2016; Hewitt et al., 2018). These raster-based algorithms can produce high quality images, yet they cannot benefit from the leverage that higher level abstractions bring about.

Vector-level image representation intrinsically prevents models from generating blurry samples and allows for compositional image generation which eventually may contribute to our understanding of how humans create or replicate images (B. M. Lake et al., 2017). This idea, with the introduction of sketch-based datasets such as Omniglot (B. Lake et al., 2012b), Sketchy (Sangkloy et al., 2016), and QuickDraw (Ha & Eck, 2017) has triggered a significant body of work in recent years. Stroke based image generation and parsing has been addressed with both vector supervised models and

self-supervised generation. Of these, one prominent algorithm is Bayesian Program Learning (B. M. Lake et al., 2015b), where a single model can be utilized for 5 tasks in the Omniglot challenge: (i) parsing, (ii) unconditional generation or, (iii) generating exemplars of a given concept, (iv) generating novel concepts of a type, and (v) one-shot classification. This approach is also shown to be scalable when supported by the representative capabilities of neural networks (Feinman & Lake, 2020a, 2020b), however, it requires stroke-level or vector supervision, which is costly to obtain or simply non-existent. VAE/RNN (Aksan et al., 2020; Cao et al., 2019; Chen et al., 2017; Ha & Eck, 2017) and Transformer based models (H. Lin et al., 2020; Ribeiro et al., 2020) are other common methods applied to vector based image generation. Although impressive results have been presented, stroke-level supervision is required to train these models.

Recently, self-supervised (i.e. the absence of stroke-level supervision) stroke-based image generation has been addressed with Reinforcement Learning (RL) (Ganin et al., 2018; Huang et al., 2019; Mellor et al., 2019; Schaldenbrand & Oh, 2020). We call this approach self-supervised vectorization, since the vectorization of images is learned using only raster-images as supervision. These methods mostly focus on image reconstruction and their exploration in generation is limited. For example, none of them address the conditional generation problem, or, they need the number of actions (i.e. strokes) as input.

In this study, we propose a self-supervised reinforcement learning approach where we train a drawing agent for character generation and parsing. Our drawing agent operates on the stroke-level (i.e. vector) representation of images. At each time step, our agent takes the current canvas as input and dynamically decides whether to continue drawing or stop. When a 'continue' decision is made, the agent outputs a program specifying the stroke to be drawn. A non-differentiable renderer takes this program and draws it on the current canvas. Consequently, a raster image is produced stroke-by-stroke. We first train this agent for two tasks by formulating appropriate loss functions: (i) unconditional character generation and (ii) parsing.

Unconditional character generation is the task of generating a novel concept[1] (i.e. character) given a dataset of concepts. For this task, our loss function includes the following components: an adversarial loss produced by a discriminator to make generated characters as "real" as possible, and two data fidelity losses assessing the conformity of the current canvas with the statistical properties of the overall dataset. We also use an additional entropy loss to prevent mode collapse.

In the parsing task, the goal for our agent is to reconstruct a given character (in raster-image) by drawing it through strokes using as few of them as possible. We utilize the same action space and environment as in the unconditional generation model, only difference being the input fed to the policy is a complete canvas to be reconstructed. Our reward function in this task has two components: a fidelity reward that indicates how much of a stroke is consistent with the target image and a penalty that increases with every 'continue' action being taken. This model explicitly learns the vectorization of the input raster-image in a self-supervised manner.

Next, we show that our parsing model can be exploited for exemplar generation (i.e. a

[1] Omniglot challenge terminology.

novel drawing of a given character) and novel concept generation from type (i.e. novel character generation given an alphabet of 10 characters) *without any further training*. Given a character, the policy network of our parsing model outputs a distribution over the action space where likelihood of actions at each time step eventually allows us to generate variations of the input image. For novel concept generation conditioned on a type (i.e. alphabet), we compose a stroke library by parsing the provided inputs. As we sample strokes from this library, we observe novel samples forming, in coherence with the overall structure of the alphabet. *To the best of our knowledge, we are the first to tackle these tasks with a self-supervised approach that operates on stroke space.*

Through experiments we show that our agent can successfully generate novel characters in all three ways (unconditionally, conditioned on a given alphabet, conditioned on a given character), and parse and reconstruct input characters. For both exemplar generation and type conditioned novel concept generation, we provide LPIPS (Zhang et al., 2018a), L2 and SSIM measures between input samples and generated images.

1.3 Contributions of the Study

In this study, we present, for the first time, a drawing agent that can successfully handle all of the generation and parsing tasks in the Omniglot challenge in a self-supervised, stroke-based manner. The model trained for parsing is able to (i) generate exemplars of a given concept by capturing the overall structure of a character and (ii) produce novel concepts (characters) for an alphabet (type). Additionally, we provide qualitative benchmarks for the 'exemplar generation' and 'type conditioned novel concept generation' tasks.

1.4 Organization of the Thesis

Chapter 2 initially examines the research on universality in writing systems, then reviews the previous work on Omniglot challenge and the importance of visual Turing test in this context, vector-based image generation and program generation. Chapter 3 introduces the proposed models for unconditional generation and parsing. This is followed by an explanation of how the parsing model is utilized for exemplar generation and type-conditioned concept generation. Chapter 4 analyzes the method with experimental results. This chapter provides qualitative and quantitative results of the aforementioned tasks. Chapter 5 summarizes the thesis and proposes possible directions for the future work.

CHAPTER 2

RELATED WORK

The main purpose of this work is to present a self-supervised approach in order to solve the generation and parsing tasks in the Omniglot Challenge (B. M. Lake et al., 2015b), by capturing the stroke-level representation of images. In this chapter we initially present the domain in which we work, writing systems. Secondly, we elaborate on the purpose of Omniglot challenge and review the existing approaches. Third, we review the proposed visual Turing tests with an emphasis on its role in the Omniglot challenge. Then, we examine the work on image vectorization by categorizing them into supervised and unsupervised methods. Finally, we touch upon the research on program synthesis in the context of this study.

2.1 Universality in Writing Systems

As a major factor influencing the human civilization, researchers have been interested in how the writing systems emerged. The studies mainly investigate the universality in writing systems and how the characters are processed by the brain.

There are significant commonalities among writing systems. Statistical analyzes show that there is a certain pattern in how the strokes are combined to form a character (Chang et al., 2015; Changizi et al., 2006; Dehaene & Cohen, 2011; Dehaene & Dehaene-Lambertz, 2016; Pagliarini et al., 2017). First, number of strokes, *length*, of a character is 3 on average, independent of the alphabet (Changizi & Shimojo, 2005). Second, approximately 50% of the strokes in a character is redundant; we can often recognize a letter even if half of the strokes are removed (Changizi & Shimojo, 2005). Last, the main difference between characters is not the number of strokes, rather the types of strokes (Changizi, 2001; Changizi & Shimojo, 2005; Changizi et al., 2006). The universality among writing systems stems the idea that there might be something innate determining the configuration of characters.

Hence, the other branch of research regarding the reading process in human brain. A specific part of the human cortex is reserved for reading and arithmetic (Dehaene & Cohen, 2007; Dehaene & Dehaene-Lambertz, 2016) and the biological process of reading is culture independent. However, the invention of writing (approximately 6000 years ago) is too recent to evolve human brain. This emerged the *neural recycling* hypothesis (Dehaene & Cohen, 2007), that is, some cultural evolutions use the cortical areas developed for other functionalities. In the case of reading, the recycling takes place in the brain circuits initially reserved for object recognition (Dehaene,

2010; Dehaene & Cohen, 2007). The use of object recognition area for the reading process might be introducing certain inductive biases in the structure of writing systems. One of them is that length of a character matches with the number of objects that can be stored in visual short-term memory (Changizi & Shimojo, 2005; Changizi et al., 2006). The other one is related to the types of junctions in a character; they match the configurations of natural signs. It is shown that the frequency of junction types in natural scenes are in line with the configuration distribution in letters (Changizi et al., 2006; Testolin et al., 2017). In this study, we consider that novel character generation requires capturing the underlying rules that form the writing systems.

2.2 Omniglot Challenge

Omniglot dataset of world alphabets was released with a set of challenges: parsing a given letter, one shot classification, generating a new letter given an alphabet, generating a novel sample of a character, and unconditional generation. Currently, there is a gap between how people learn and how machines learn. To address this, Omniglot challenge is designed for encouraging research towards developing generalizable algorithms which use very few data that are able to achieve generative tasks and one-shot classification with a single model. The ultimate goal here is to design a model that can pass a type of visual Turing test. People are asked whether a human or a machine is generated a given character, and their Bayesian model was quite successful to deceive humans.

Omniglot letters have samples that are conditionally independent based on the alphabet-character hierarchy, hence, a distinctive approach to achieve all these tasks is Hierarchical Bayesian modeling (B. M. Lake et al., 2015b), (B. M. Lake et al., 2013). As the Omniglot letters included human strokes as labels, the compositional and causal nature of letters are leveraged to model the generation process. Later, neurosymbolic models are also shown to be successful for unconditional generation (Feinman & Lake, 2020a) and conceptual compression for multiple tasks presented within the Omniglot Challenge (Feinman & Lake, 2020b). These studies has been successful in capturing the flexible representation of concepts and they are shown to achieve several tasks within the Omniglot challenge with a good performance in terms of the visual Turing test.

However, without the stroke set that generated a concept, these tasks become more difficult. The idea of sequential image generation is examined by recurrent VAE models (Rezende et al., 2016), (Gregor et al., 2015), (Gregor et al., 2016). DRAW (Gregor et al., 2015) and Convolutional DRAW (Gregor et al., 2016) were able to generate quality unconditional samples from MNIST and Omniglot datasets respectively. DRAW is proposed as an algorithm to generate images recurrently. The network is able to iteratively generate a given image by attending to certain parts of the input at each time step. Convolutional DRAW improved the idea with an RNN/VAE based algorithm that can capture the global structure and low-level details of an image separately in order to increase the quality of generations. Later, it is shown that Hierarchical Bayesian Modeling can be improved by the representational power of deep learning and attentional mechanisms in order to achieve three of the five Omniglot challenges (Rezende et al., 2016). Another novel idea to leverage Bayesian

modeling to tackle Omniglot Challenge was performing modifications on the VAE architecture to represent hierarchical datasets (Edwards & Storkey, 2016) (Hewitt et al., 2018). The significance of these studies is that they were able obtain latent variables to describe class-level features effectively. Despite the ability to utilize the same model for different problems (one-shot classification, unconditional and conditional generation), raster-based one-step generative models have two disadvantages we want to address. First, they cannot leverage the higher level abstraction and quality comes with working on a vector space. Secondly, one-step generation does not provide an interpretable compositional and causal process describing how a character is generated. In this work, we combine the advantages of two groups of aforementioned models with an agent operating on stroke representation of images that uses only raster images during training. Thus, we aim to solve all three generative and the parsing (reconstruction) tasks of the Omniglot challenge. We show that the model trained for reconstruction can also be adopted as a tool that captures the compositional structure of a given character. Without any further training, our agent can solve exemplar generation and type conditioned novel concept generation problems.

2.3 Visual Turing Test

The Turing test (Turing & Haugeland, 1950) is accepted as the gauge of machine intelligence by the research community. As intelligence is a comprehensive concept that also involves visual or audial abilities (Custer, 1998), there are studies that propose different Turing tests, such that, *"visual Turing test"* is related to the purpose of this study.

In the recent years, the idea of using visual Turing test as a metric has become widespread. For different tasks, studies proposed different visual Turing tests to test AI models. These include scene understanding (Geman et al., 2015; Qi et al., 2015), image captioning and visual question answering (H. Gao et al., 2015; Lu et al., 2019; Ma et al., 2016; Malinowski, 2017; Malinowski & Fritz, 2015; Sharma & Jalal, 2021; Väth et al., 2021; Vedd et al., 2021), video understanding (Heo et al., 2021), assessing the security of CAPTCHA images (Y. Gao et al., 2021), measuring the generation quality (Fenu & Bagwell, 2016; Q. Lin et al., 2018; Mazzone & Elgammal, 2019; Nakazawa et al., 2021; Salimans et al., 2016; Souibgui et al., 2021; Yang et al., 2021) or even their ability of being susceptible to Kuleshov effect (Olague et al., 2021). From these, image captioning, visual question answering and video understanding are multimodal tasks that also measure the linguistic abilities.

One of the goals here is to develop metrics that are able to move beyond "what is where" question (Tan et al., 2013). This is considered important in terms of measuring a model's reliability or interpretability, e.g. in medical applications (Fountoukidou & Sznitman, 2019; Nakazawa et al., 2021) or autonomous cars (Dia et al., 2021).

Another goal, that is more related to this study, is to measure the performance of generative models. Although there are metrics to assess if a generative model is able to capture the distribution of a given dataset, the quality of individual generations can be measured by asking humans. Here, we ensure the quality of our agent's generations by utilizing strokes instead of a raster-based approach.

2.4 Image Generation by Vectorization — With Stroke Supervision

Sketch-RNN (Ha & Eck, 2017) is the first LSTM/VAE based sketch generation algorithm. It is later improved to generate multiclass samples (Cao et al., 2019) and increase the quality of generations by representing strokes as Bezier curves (Song, 2020). The idea of obtaining a generalizable latent space by image-stroke mapping is studied by many (Aksan et al., 2020; Bhunia et al., 2021; Das et al., 2021; Wang et al., 2020). In CoSE (Aksan et al., 2020), the problem is articulated as 'completion of partially drawn sketch'. They achieved state of the art reconstruction performance by utilizing variable-length strokes and a novel relational model that is able to capture the global structure of the sketch. The progress in stroke representation is continued with incorporation of variable-degree Bezier curves (Das et al., 2021), and capturing Gestalt structure of partially occluded sketches (H. Lin et al., 2020).

2.5 Self Supervised Vectorization

Self-supervised vector-based image generation problem has been approached by RL based frameworks (T. Zhou et al., 2018), (Ganin et al., 2018), (Mellor et al., 2019), (Huang et al., 2019), (Schaldenbrand & Oh, 2020), and (Zou et al., 2020). In SPIRAL (Ganin et al., 2018), unconditional generation and reconstruction tasks are tackled with adversarially trained RL agents. Succeeding research enhanced the reconstruction process by a differentiable renderer, making it possible for agents to operate on a continuous space (Huang et al., 2019; Schaldenbrand & Oh, 2020). In order to avert the computational expense of RL based algorithms, end-to-end differentiable models are developed through altering the rendering process (Nakano, 2019) or formulating the generation process as a parameter search (Zou et al., 2020). More recently, a differentiable renderer and compositor is utilized for generating closed Bezier paths and the final image respectively (Reddy et al., 2021). This method led to successful interpolation, reconstruction, and sampling processes. Most related to our work is SPIRAL where both reconstruction and unconditional generation is studied through self-supervised deep reinforcement learning. However, our approach has some significant differences. First, in SPIRAL each stroke is also represented as a Bezier curve, yet, the starting point of each curve is set as the final point of the previous curve. In our model, all control points of the Bezier curve are predicted by the agent at each time step. Hence, the agent has to learn the continuity and the compositionality of the given character in order to produce quality samples. Secondly, SPIRAL provides a generative model that works through a graphics renderer without addressing the conditional generation problem. They show impressive results on both natural images and handwritten characters. While we provide a solution for multiple generative tasks, we have not explored our model in the context of natural images. Another approach that presents a similar scheme to the reconstruction problem is "Learning to Paint" (Huang et al., 2019). In Learning to Paint, the proposed model is utilized specifically for reconstruction. When reconstruction is considered, the main difference of our model is that since we try to model a human-like generation process, our agent outputs a single stroke at each time step with the environment being altered throughout this process while in Learning to Paint, 5 strokes are predicted by

the agent at each time step. As a major difference from previous studies, our agent decides whether to stop or keep drawing before generating a stroke. This enables the agent to synthesize an image with as few actions as possible when motivated with our reward formulations.

2.6 Self Supervised Program Synthesis

Our method essentially outputs a visual program that depends only on the rastered data. In that sense, studies on Constructive Solid Geometry (CSG) are also related. Different RL frameworks for reconstruction of a given CSG image, that is essentially a composition of geometric shapes, are proposed (Ellis et al., 2019; C. Zhou et al., 2020). The former considered parsing as a search problem that is solved by using a read-eval-print-loop within a Markov Decision Process. The latter adopted a Tree-LSTM model to eliminate invalid programs and the reward is considered to be the Chamfer distance between the target image and current canvas.

CHAPTER 3

METHOD

Our model consists of a policy network and a (non-differentiable) renderer. At time step t, the policy network takes the current canvas, $\mathbf{C}_t$ – a raster-image, as input and outputs two distributions, π_B and π_S. The first distribution, π_B, is for stroke (i.e. Bezier curve)-parameters and the second one, π_S, is for the continue/stop decision. From the first distribution, we randomly sample a stroke defined by its 7 parameters (x-y coordinates of start, end, control points of the quadratic Bezier curve, and a brush-width). From the second distribution, we randomly sample a decision. If the decision happens to be 'continue', we add the newly sampled stroke to the current canvas, $\mathbf{C}_t$, increment time (i.e. $t \leftarrow t + 1$) and restart. If the decision was to 'stop', then $\mathbf{C}_t$ is returned as the final output. Our model is able to handle parsing and different generation tasks, and the processing pipeline we just described is common in all these tasks. What changes among tasks is the reward functions and/or training procedures, which we explain below.

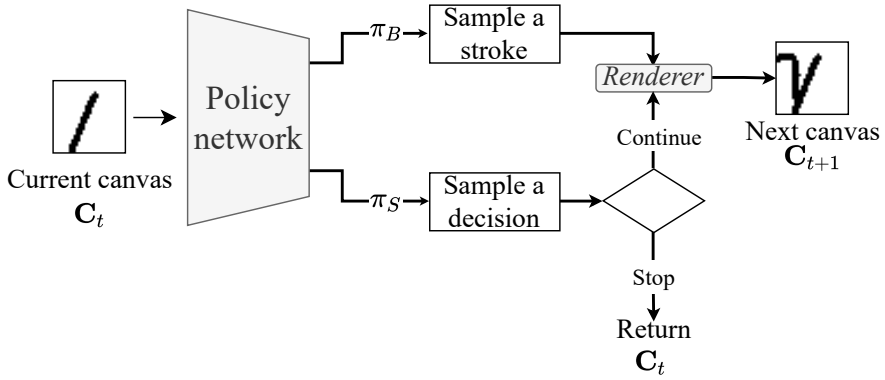


Figure 3.1: Generator model. At each time step, the policy network receives a canvas and outputs two distributions for Bezier curve parameters and stop/continue decision. When the 'continue' decision is sampled, the resulting stroke is rendered and added to the final output.

3.1 Unconditional Generation

Unconditional generation or 'generating new concepts' refers to sampling from a distribution approximated during the training process. In this study, we use the whole Omniglot dataset for training, where the purpose is to obtain a model that can generate novel characters. The process starts with an empty canvas and allows the policy network to draw a character freely. Figure 4.2 demonstrates one forward pass of the unconditional generation process.

3.1.1 Policy Network

For our agent, an episode starts with an empty canvas and finishes when 'stop' action is sampled. During the episode agent acts on the partially drawn canvas, $\mathbf{C}_t$, that is updated at each time step with a new stroke.We adopt an adverserial approach to train the agent. At each time step t, we calculate an instantaneous reward, r_t, that has three components:

$$r_t = D(\mathbf{C}_t) + \lambda_1 \text{align}(\mathbf{C}_t, \mathbf{I}) + \lambda_2 \mathcal{N}(|\mathbf{C}_t|; \mu, \sigma). \tag{3.1}$$

The first term is a reward based on a discriminator to make generated characters as 'real' as possible. $D(\cdot)$ is a discriminator that outputs the "realness" score of its input canvas. We train it in an adversarial manner by using the generated examples as negatives and the elements of the input dataset as positives.

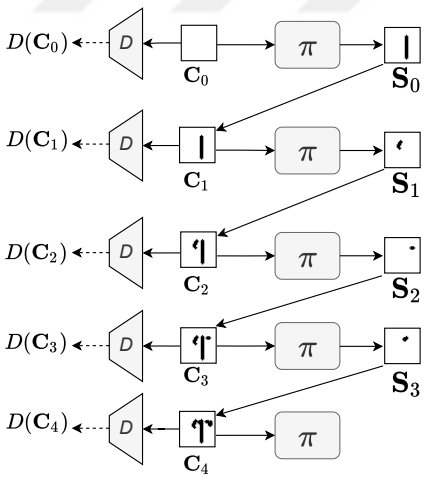


Figure 3.2: Forward pass of unconditional generation. The process starts with an empty canvas. At $t = 0$, the policy network generates a random stroke. Then, at each time increment, a new stroke is added to the input canvas. This process continues recursively till a 'stop' action is sampled. Partially drawn canvas is passed through the discriminator to get a 'realness' score.

The second and third terms are auxilary rewards that accelerate learning as they guide the exploration within the vicinity of real characters.

Second term is a is a clustering-based data fidelity reward. This reward encourages the diversity within generated samples while rewarding the actions that conform with the dataset. To calculate this reward, initially, we apply k-means on all characters in the input dataset to obtain cluster centers. Throughout the episode we quantify the alignment of the current canvas, $\mathbf{C}_t$, and a cluster center $\mathbf{I}$ that is randomly selected at the begining of each episode. align basically counts the number of intersecting on-pixels (between the two canvases) minus the number of non-intersecting on-pixels in $\mathbf{C}_t$, and divides this quantity by the number of on-pixels in $\mathbf{I}$.

$$x_1(\mathbf{C}_t, \mathbf{I}) = \sum_{i,j} (\mathbf{C}_t \cdot \mathbf{I})_{i,j} \tag{3.2}$$

$$x_2(\mathbf{C}_t, \mathbf{I}) = -\sum_{i,j} 1[(\mathbf{C}_{t,(i,j)} \neq 0) \wedge (\mathbf{I}_{(i,j)} = 0)] \tag{3.3}$$

$$\mathrm{align}(\mathbf{C}_t, \mathbf{I}) = \frac{x_1 + x_2}{\sum_{i,j} 1[\mathbf{I}_{(i,j)} \neq 0]} \tag{3.4}$$

$$\tag{3.5}$$

The final term assesses the current canvas, $\mathbf{C}_t$, by counting the number of on-pixels. $\mathcal{N}(|\mathbf{C}_t|; \mu, \sigma)$ evaluates a normal distribution with (μ, σ) at $|\mathbf{C}_t|$ which is the number of on-pixels in the current canvas. We obtain (μ, σ) by fitting a normal distribution to the on-pixel counts of characters in the training set. We update the model through temporal credit assignment; at each time t, we use the difference of successive rewards, i.e. $r_t - r_{t-1}$.

Finally, in order to encourage exploration and avoid mode collapse, we use an entropy penalty term L_H as

$$L_H = \alpha \max(0, \mathrm{KL}([\pi_B, \pi_S], \mathrm{U}) - \tau). \tag{3.6}$$

Here, KL indicates KL-divergence and U is the uniform distribution. This term first measures the divergence between the uniform distribution and π_B, π_S, the distributions output by the policy network. Then, through the hinge function, if the divergence exceeds a threshold (τ), this term activates and increases the penalty. For a sampled action a given the state of the canvas s_t the resulting policy loss becomes:

$$L_{\pi_\theta} = -\log \pi(a|s_t; \theta)(r_t - r_{t-1}) + L_{H,t} \tag{3.7}$$

We employ the REINFORCE algorithm (Williams, 1992) to update the weights of the policy network.

3.1.2 Discriminator

We train the discriminator using hinge loss. In order to stabilize the discriminator and keep the Lipschitz constant for the whole network equal to 1, Spectral Normalization is applied at each layer (Miyato et al., 2018). Given the generator G, discriminator D, real image x, and the fake sample $D(G(z))$, the discriminator loss L_D is as follows:

$$L_D(G, D) = \mathbf{E}_{x \sim p_{real}(x)}[\max(0, 1 - D(x))] + \mathbf{E}_{z \sim p_z}[\max(0, 1 + D(G(z)))] \quad (3.8)$$

Throughout the training, we kept the balance ratio between generated and real samples at 3. Each update of the discriminator is preceded by the generation of *BATCH_SIZE* samples. After each epoch, additional fake samples are obtained through hard negative sampling. The hard negatives are obtained using the generations in the previous epoch and the current hard negatives.

3.2 Image Reconstruction by Parsing

In the "parsing" task, the goal is to reconstruct the given input image by re-drawing it through strokes as accurately as possible. To this end, we formulate a new reward function with two terms: a fidelity reward that indicates how much of a stroke is consistent with the input image (using the "align" function introduced above) and a penalty that is added with every time increment represented by t as 'continue' decisions being made:

$$r_t = \text{align}(\mathbf{S}_t, \mathbf{C}_t) - \lambda_1 t, \quad (3.9)$$

where $\mathbf{S}_t$ is the newly sampled stroke and $\mathbf{C}_t$ is the current canvas (input). Second term simply acts as a penalty for every 'continue' action. The first term ensures the sampled stroke to be well-aligned with the input and the second term forces the model to use as few strokes as possible. There is no need for a discriminator. This model explicitly learns the vectorization of the input raster-image in a self-supervised manner.

Apart from the different reward function, another crucial difference between the training of the unconditional generation model and the parsing model is how the input and output are handled. In unconditional generation, the newly-sampled stroke is added to the current canvas, whereas in parsing, we do the opposite: the sampled stroke is removed (masked out) from the current canvas, and the returned final canvas is the combination of all sampled strokes until the 'stop' decision. Figure 3.3 demonstrates this process.

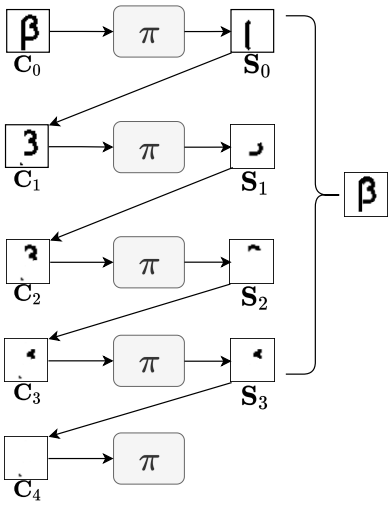


Figure 3.3: Image reconstruction process. Given an input image, the policy network generates a stroke at each time step. Sampled stroke is masked out of the input image and the process is repeated till a 'stop' decision is made. The final canvas is a superposition of generated strokes.

3.3 Generating New Exemplars

In this task, a model is required to generate a new exemplar (i.e. a variation) of an unseen concept (i.e. character). We utilize our parsing network described in the previous section to capture the overall structure of a given letter. In order to produce new exemplars, we randomly sample different parsings (a set of strokes) from the distribution generated by the agent. In order to eliminate 'unlikely' samples, we compute the likelihood of the parsing using the resulting policy, and apply a threshold.

3.4 Generating Novel Concepts from Type

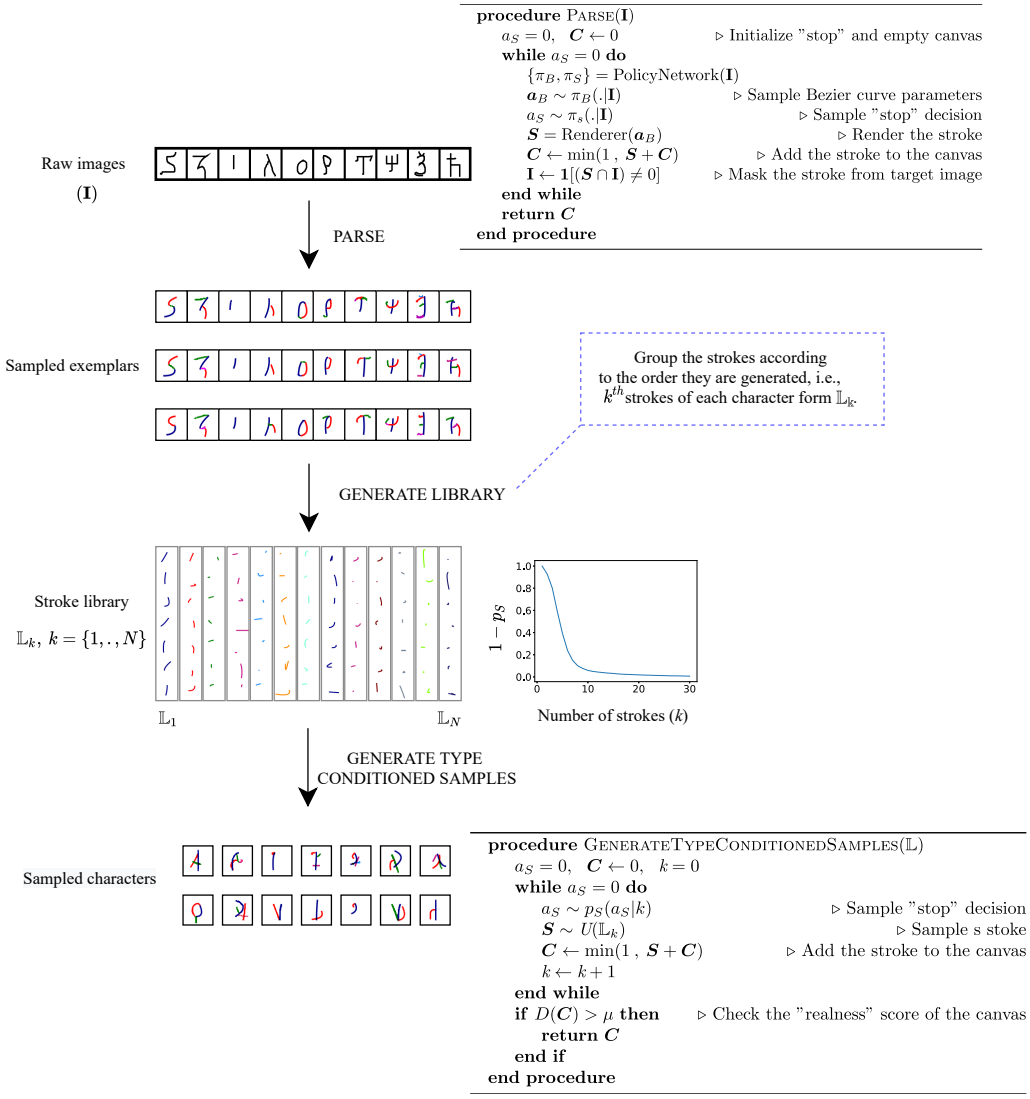
In this task, the goal is to to generate a novel concept (i.e. character) given a previously unseen type (i.e. alphabet) consisting of 10 concepts. The novel concepts should conform to the overall structure, that is, the stroke formulation and composition of the given type (alphabet). We, again, tackle this challenge using our parsing network without any further training.

To do so, we first parse all input images into its strokes. For each input image, we sample five stroke sets from the stroke-parameters distribution output by the policy network. During the sampling process, we again use the likelihood-based quality function described in the previous section.

We add all the strokes sampled during this process to form a *stroke library*. Here the

strokes are stored with the time steps they are generated. Noting that the number of strokes sampled for a given character is not constant, we approximate a distribution for stopping actions. This process provides a stroke set representing the structure of letters and the way they are composed, that is, we can exploit the compositionality and causality of an alphabet.

Throughout the character generation process, a stroke is sampled at each time step belonging to that particular group of the library. The sampled strokes are summed together to obtain the final canvas.



procedure PARSE($\mathbf{I}$)
  $a_S = 0, \quad \boldsymbol{C} \leftarrow 0$ ▷ Initialize "stop" and empty canvas
  while $a_S = 0$ **do**
    $\{\pi_B, \pi_S\} = \text{PolicyNetwork}(\mathbf{I})$
    $\boldsymbol{a}_B \sim \pi_B(.|\mathbf{I})$ ▷ Sample Bezier curve parameters
    $a_S \sim \pi_s(.|\mathbf{I})$ ▷ Sample "stop" decision
    $\boldsymbol{S} = \text{Renderer}(\boldsymbol{a}_B)$ ▷ Render the stroke
    $\boldsymbol{C} \leftarrow \min(1\ ,\ \boldsymbol{S} + \boldsymbol{C})$ ▷ Add the stroke to the canvas
    $\mathbf{I} \leftarrow \mathbf{1}[(\boldsymbol{S} \cap \mathbf{I}) \neq 0]$ ▷ Mask the stroke from target image
  end while
  return $\boldsymbol{C}$
end procedure

procedure GENERATETYPECONDITIONEDSAMPLES($\mathbb{L}$)
  $a_S = 0, \quad \boldsymbol{C} \leftarrow 0, \quad k = 0$
  while $a_S = 0$ **do**
    $a_S \sim p_S(a_S|k)$ ▷ Sample "stop" decision
    $\boldsymbol{S} \sim U(\mathbb{L}_k)$ ▷ Sample s stoke
    $\boldsymbol{C} \leftarrow \min(1\ ,\ \boldsymbol{S} + \boldsymbol{C})$ ▷ Add the stroke to the canvas
    $k \leftarrow k + 1$
  end while
  if $D(\boldsymbol{C}) > \mu$ **then** ▷ Check the "realness" score of the canvas
    return $\boldsymbol{C}$
  end if
end procedure

Figure 3.4: Conditional generation process. Exemplars are generated using the policy learned by reconstruction training. In order to eliminate 'unlikely' samples, the likelihood function defined by the policy is used, i.e., a likelihood threshold is used. To achieve the alphabet (type) conditional generation task, a stroke library is formed. With this library, a set of strokes randomly sampled at each time step and obtained a new canvas. In order to obtain good (or real looking) samples, the discriminator trained for unconditional generation is employed.

CHAPTER 4

EXPERIMENTS

In this chapter, we present the experimental results. Initially, we explain the details of the dataset and implementation. Then, we provide qualitative and quantitative results of the experiments. For unconditional generation and parsing, we tested our approach on MNIST and Omniglot datasets. For examplar generation and type-conditioned concept generation, we used Omniglot dataset, as it has a hierachical structure that conform with these tasks.

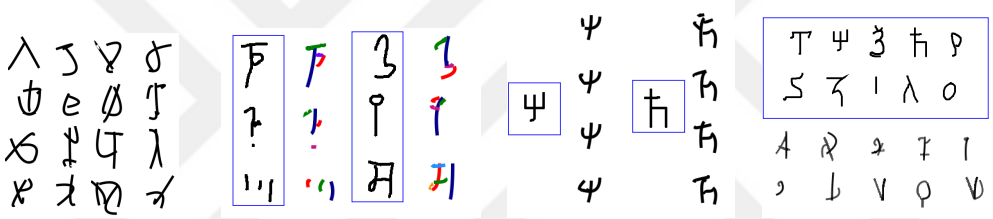


Figure 4.1: Our drawing agent can accomplish four different tasks. From left to right: it can generate novel characters, parse a given character into its strokes, generate new exemplars for a given character, and generate novel concepts (i.e. characters) given a type (i.e. alphabet). **Ours is the first stroke-based method to tackle all of the generation and parsing tasks in the Omniglot Challenge, without requiring any stroke-level supervision.**

4.1 Datasets and Implementation Details

We report generation and reconstruction (parsing) results on the Omniglot dataset (B. M. Lake et al., 2015b), which includes 1623 characters from 50 different alphabets, with 20 samples for each character. 30 alphabets are used for training and the remaining 20 are used for evaluation. For unconditional generation and reconstruction, we also report results on the MNIST dataset (LeCun, 1998). For both datasets, we rescale input images to 32x32.

The backbone is a ResNet with 3 convolutional layers and 8 residual layers. The first

convolutional layer has 32 filters of size 5x5. The second and third convolutional layers have 32 filters of size 4x4 and stride of 2, resulting in a tensor with dimensions 8x8x32. Then, we use standard residual layers described in (He et al., 2016). The output of the final residual layer is flattened to a 2048x1 vector to be processed by the MLPs. The first MLP outputs a set of distributions for each control point of the Bezier curve. It has 1 fully connected layer that outputs a 192x1 vector. This vector is reshaped to a 32x6 matrix where each 32x1 vector defines a distribution over the possible coordinates. The MLPs used for selecting the brush width and sampling the stop/continue decision consist of 2 layers with 64 and 2 neurons. Architectural details can be found in Figure 4.2.

For the Omniglot dataset, we take brush width as a constant and omit the corresponding MLP branch.

We tune the learning rate and weight decay of the generator, λ hyperparameters in Equation 3.1 and Equation 3.9, α and τ hyperparameters in Equation 3.6, using the Tree-structured Parzen Estimator algorithm (Bergstra et al., 2011) in the RayTune library (Liaw et al., 2018).

For unconditional generation, we use the discriminator architecture proposed by (Miyato et al., 2018). In order to stabilize the discriminator and keep the Lipschitz constant for the whole network equal to 1, Spectral Normalization is applied at each layer. Discriminator is trained using the hinge loss. Throughout the training, we set the balance ratio between fake and real samples as 3. We performed hard-negative mining to speed up convergence during this process.

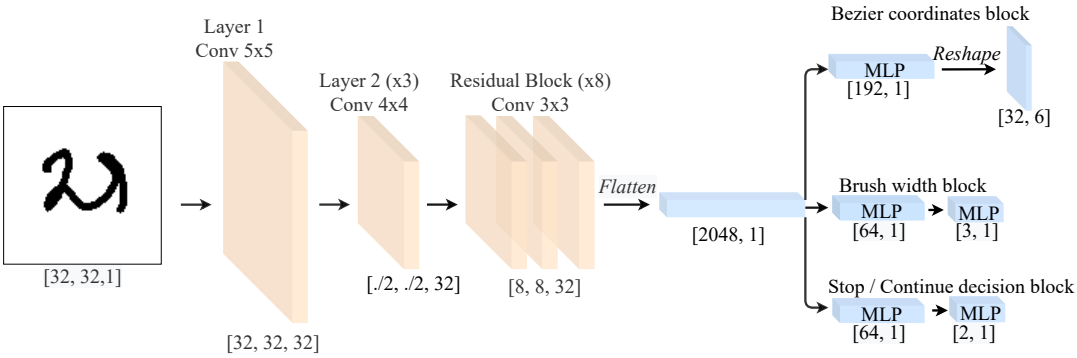


Figure 4.2: Overall architecture of the policy network. The network consists of a CNN backbone and 3 MLP braches. Each convolutional layer is followed by Batch Normalization and ReLU activation. After the last residual block, resulting tensor is flattened. Between the layers of MLP branches, ReLU activation is used. Finally, via a softmax function the action distributions are obtained.

4.2 Unconditional Generation

We initially tested our approach on the MNIST dataset. Figure 4.3 presents the improvement in the quality of samples generated throughout the policy network updates. We use FID score (Heusel et al., 2017) for quantitative evaluation. FID score provides a distance measure between the training dataset and the fake samples. Initially, each image from the fake and real datasets are processed through a pretrained InceptionV3 (Szegedy et al., 2016) model and feature vectors are obtained. Then, these features are fit to two multidimentional Gaussian distributions, one for fake samples and one for real samples. FID is the Wasserstein distance between these two distributions.

At the beginning, generated characters are mostly random scribbles. Towards the end, they start to look like real digits. Table 4.1 shows that our method achieves an acceptable FID score given the scores of other prominent GAN and VAE methods. Presented FID values are taken from (Lucic et al., 2017).

Table 4.1: Comparison of the FID scores for different models on the MNIST dataset. We report the mean and the variance of FID scores from 5 simulations with different weight initializations.

	FID (Less the better)
LSGAN (Mao et al., 2017)	9.8 ± 0.9
NSGAN (Goodfellow et al., 2014)	6.8 ± 0.5
WGAN (Arjovsky et al., 2017)	6.7 ± 0.4
WGAN GP (Gulrajani et al., 2017)	20.3 ± 5.0
DRAGAN (Kodali et al., 2017)	7.6 ± 0.4
VAE	23.8 ± 0.6
Ours	17.3 ± 3.0

Figure 4.4 shows sample generations for the Omniglot dataset. To show that our generations are not duplicates of the characters in the training set, we present the four most similar characters from the training set to our generations. Similarity is computed using pixelwise L2 distance.

Finally, Figure 4.5 presents more generated characters, which demonstrate the variability and the quality of generated concepts. The agent was able to capture the type of strokes, number of strokes a character has and letter structures without any stroke supervision.

We examine the effect of the terms in Equation 3.1. As the terms of the reward function are removed, the overall performance decrease. In Table 4.2 we provide the resulting FID scores.

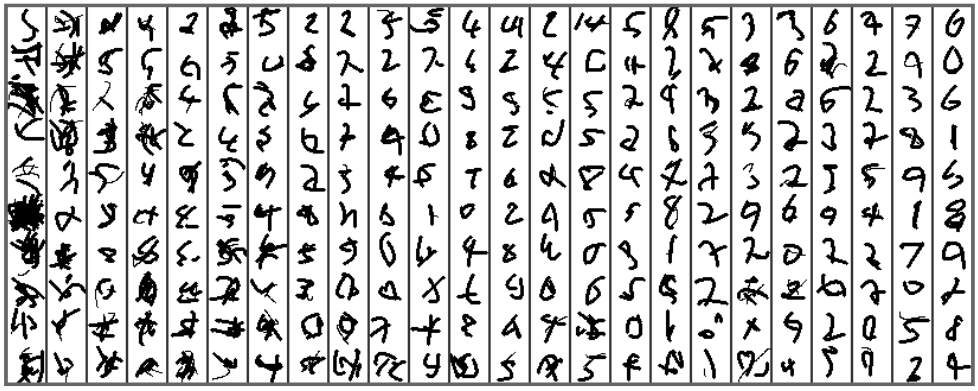


Figure 4.3: Quality of generated MNIST characters as training progresses (i.e. policy is updated) from left to right.



Figure 4.4: Omniglot unconditional samples. For randomly sampled generations, four closest samples (in terms of pixelwise L2 distance) from the training dataset are presented.

Table 4.2: Ablation experiments for the loss term in Equation 3.1. λ_1 represents the coefficient of the term that measures the alignment of the canvas with the cluster center. λ_2 represents the coefficiet of the term that measures the distance of the current canvas to the distribution obtained from the dataset in terms of 'on' pixels.

λ_1	λ_2	Balance Ratio	Hard Negative Mining	FID (Less the better)
0	0	3	✓	65.71
0	1	3	✓	52.55
1	0	3	✓	52.26
1	1	1	✓	52.87
1	1	1		57.43
0	0	1		60.95
1	**1**	**3**	✓	**45.04**

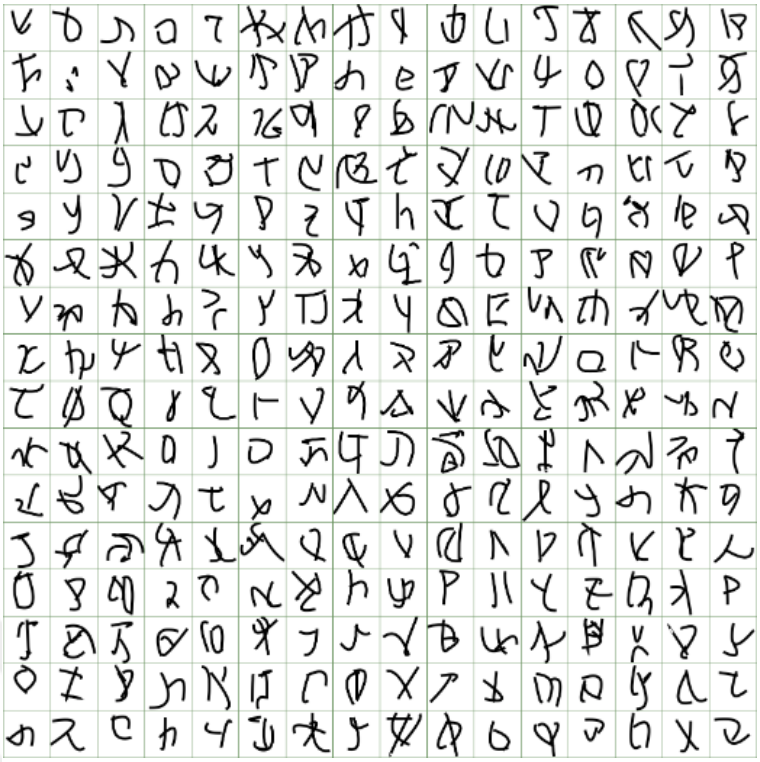


Figure 4.5: Randomly sampled unconditional generations for the Omniglot dataset.

4.3 Image Reconstruction by Parsing

We initially tested our approach for reconstruction on the MNIST dataset. Figure 4.6 presents sample parsing and reconstruction results on MNIST. Our agent can reconstruct a character from the test set in a minimal number of actions within the abilities of quadratic Bezier curves. Selected brush widths also conform with the stroke heterogeneity of the dataset. We have also observed that there is usually a consistency in the order of strokes.

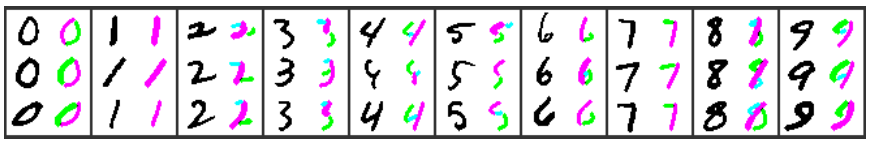


Figure 4.6: MNIST reconstructions. For each sample on the left hand side of the columns, parsing processes are demonstrated. Colors represent the order of the strokes. (*pink*: first stroke, *green*: second stroke, *blue*: third stroke)

Then, we train our model with the characters in the Omniglot training set. For evaluation, we utilize the evaluation set with completely novel characters from unseen alphabets. Thereby, we can see that our agent has learned how to parse a given char-

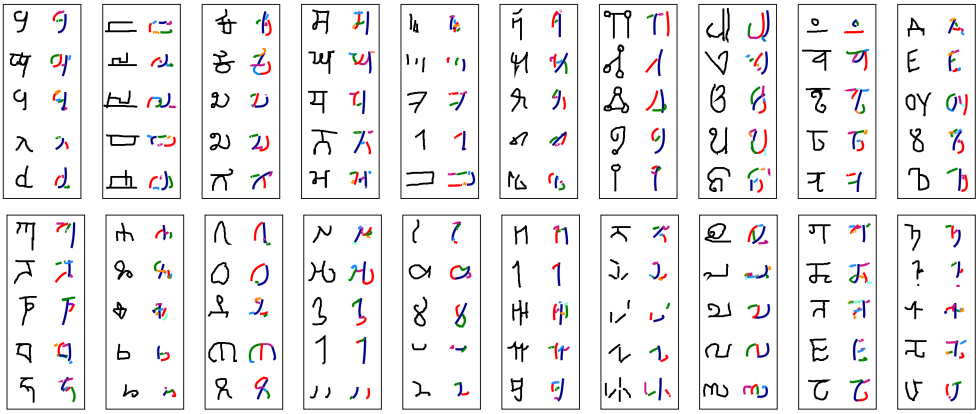


Figure 4.7: Omniglot reconstruction. For each sample on the left hand side of the columns, resulting reconstructions are demonstrated.

Table 4.3: Reconstruction quality. L2 distance between target and the reconstructed image. (ImageVAE is taken from (Reddy et al., 2021), it indicates a purely raster-based autoencoder.)

Method	MNIST	Omniglot
ImageVAE	0.0033	N/A
Im2Vec (Reddy et al., 2021)	0.0036	N/A
Learning to Paint (Huang et al., 2019)	0.006	N/A
SPIRAL (Training distance) (Ganin et al., 2018)	0.01	0.02
StrokeNet (Training distance) (Zheng et al., 2018)	0.015	0.02
Ours	0.04	0.06

acter. Due to the penalty term that increases with the number of strokes, there is a tradeoff for the agent to replicate a character exactly and replicate it in a small number of actions. This indirectly demotivates the agent from retouching the image with small strokes to minimize the difference to the target. Results in Figure 4.7 show that overall structure of the target images are preserved, however, small details are lacking in some of the examples. This reflects on the distance measures (Table 4.3). We provide the mean number of strokes our agent used for parsing a the characters of different test alphabets in Table 4.4.

4.4 Generating New Exemplars

For this task, we use the evaluation set of the Omniglot dataset. For each character in the test set, we sample 500 different parses from the policy. In Figure 4.8, it can

Table 4.4: For each alphabet in the Omniglot evaluation set, we present the number of strokes our agent used to reconstruct the given image vs. mean number of strokes obtained from human-labeled data. The stroke count for human-labeled data is calculated using the labels within the Omniglot dataset.

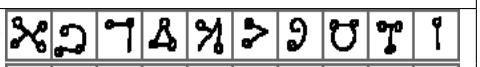
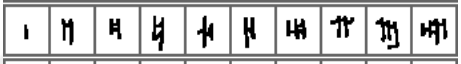
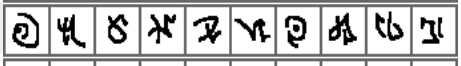
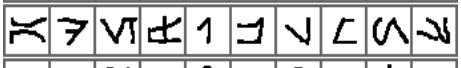
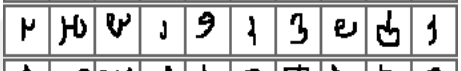
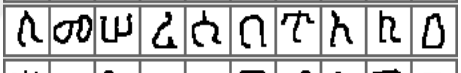
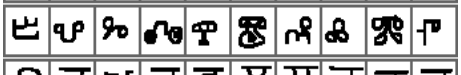
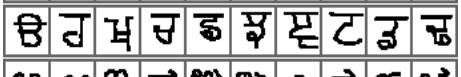
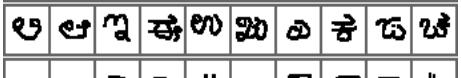
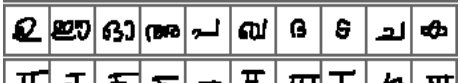
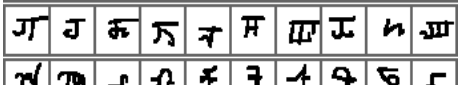
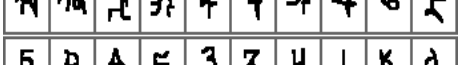
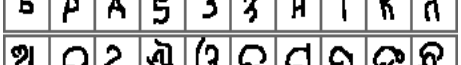
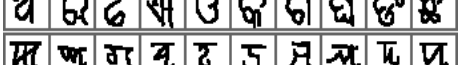
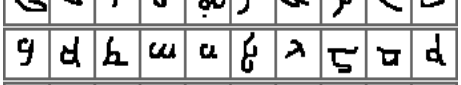
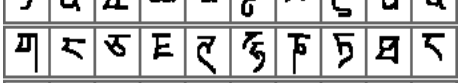
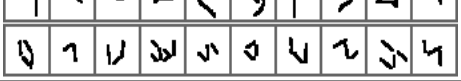
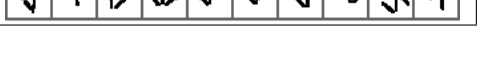
Alphabet	**Number of Strokes**		**Sample Image**
	Our Model	**Human Labeled Data**	
Angelic	3.935	4.49	
Atemayar Qelisayer	10.15	3.571	
Atlantean	6.209	2.078	
Aurek-Besh	7.6	2.565	
Avesta	9.511	1.52	
Ge_ez	10.112	1.984	
Glagolitic	5.24	2.88	
Gurmukhi	6.080	3.09	
Kannada	4.217	2.33	
Keble	8.573	4.140	
Malayalam	7.215	1.453	
Manipuri	10.676	2.82	
Mongolian	8.93	2.405	
Old Church Slavonic	5.171	2.954	
Oriya	5.59	2.82	
Sylheti	11.38	2.84	
Syriac	6.35	2.206	
Tengwar	8.088	2.492	
Tibetan	11.69	3.62	
ULOG	6.417	3.253	

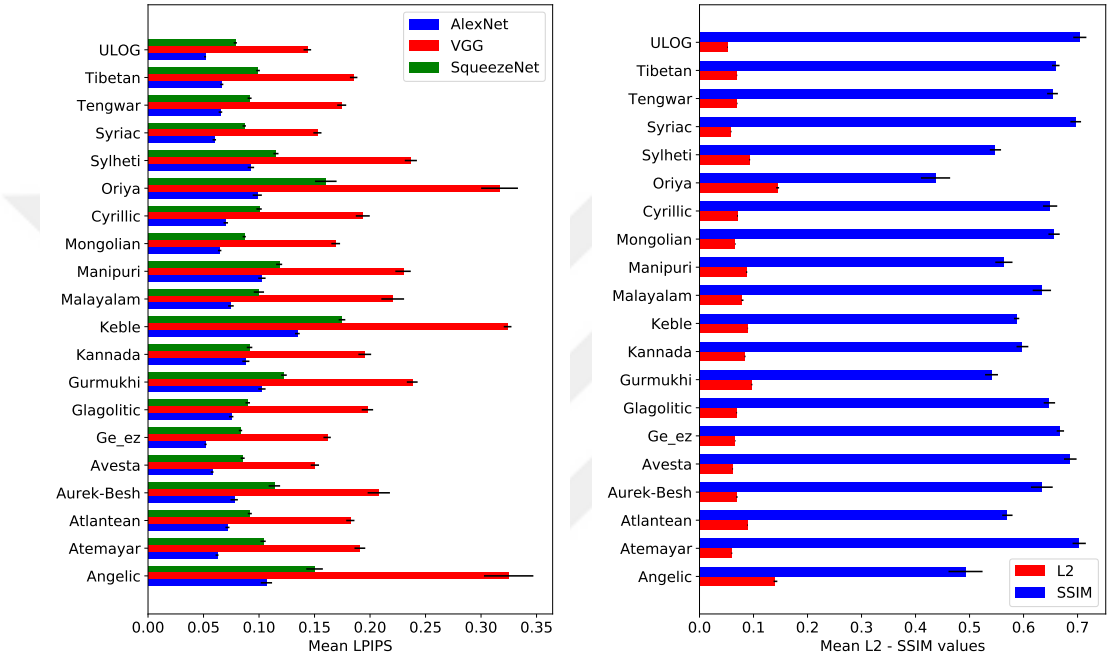


Figure 4.8: New exemplar generation. Given an unseen character from a new alphabet (highlighted in red boxes), the model generated 9 exemplars.These metrics are calculated over all examples generated for the test set.

be observed that given an unseen letter from a novel alphabet, our agent can sample from the resulting distribution, and output quality variations. The major indications of variation are structures of the strokes, number of actions to generate a sample and the fine details of certain characters. We compare each produced character with its corresponding input image using Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018b), SSIM and L2 distance values. LPIPS is the squared difference between the feature vectors of two images. Feature vectors can be calculated using AlexNet (Krizhevsky et al., 2012), VGG (Simonyan & Zisserman, 2014), or SqueezeNet (Iandola et al., 2016). The mean and standard deviation of these values for the whole evaluation set are 0.078 ± 0.002 (for features obtained through AlexNet), 0.616 ± 0.018 and 0.08 ± 0.016, for LPIPS, SSIM, and L2 respectively. Results per alphabet can be found in Figure 4.9.

4.5 Generating Novel Concepts from Type

In order to generate a concept that is likely to belong in a given alphabet, we again leverage our reconstruction model. Given 10 different characters of an unseen alphabet, we are able to generate novel images with similar structural features. Results presented in Figure 4.10 show that our algorithm can model the compositional pattern of an alphabet in stroke space. In order to obtain quantitative results, we produce 10000 images conditioned on each input set and randomly sample characters by utilizing the discriminator trained for the unconditional generation model, assuming it has learned what features of a given input imply a real character. We generate a sampling distribution according to the discriminator scores of generated samples and repeat the sampling process multiple times for each input to obtain a set of outputs to be considered. For a sample generated, we calculate performance metrics with respect to all characters in the input. In order to report final metrics presented in supplemental figures 4.11b and 4.11a, we consider the most similar input-output pairs. The mean and standard deviation of LPIPS, SSIM and L2 values for the whole evaluation set are 0.08 ± 0.003, 0.5 ± 0.07 and 0.12 ± 0.0009 respectively.

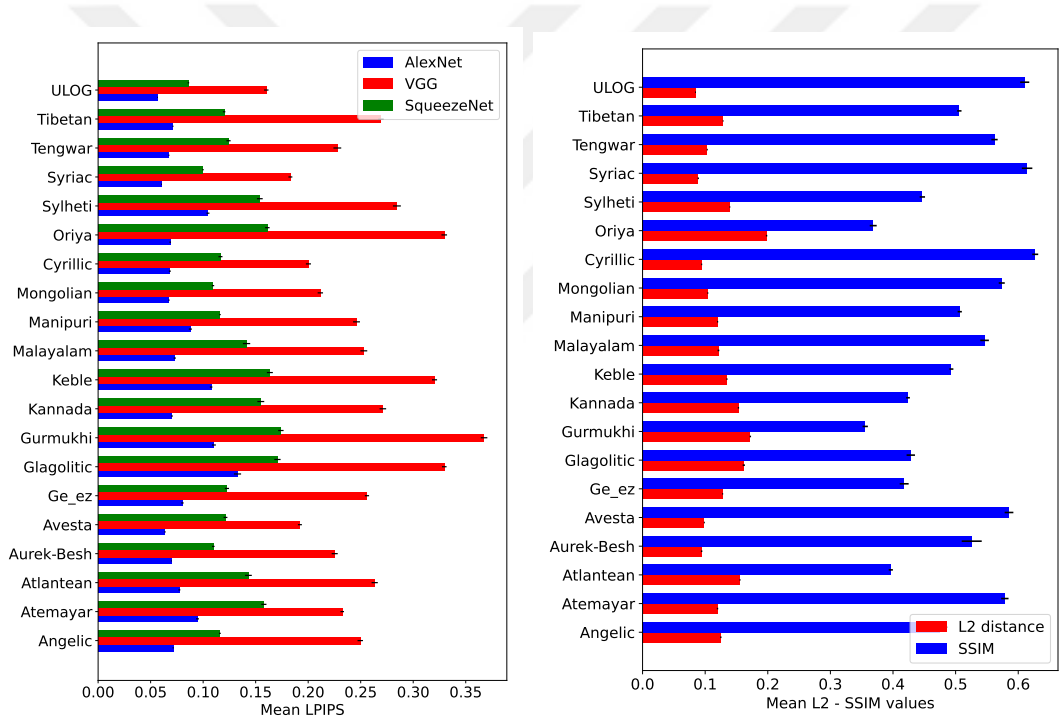


(a) (b)

Figure 4.9: LPIPS values for each alphabet in the test set calculated from sampled exemplars (a), SSIM and L2 values for each alphabet in the test set calculated from sampled exemplars (b).



Figure 4.10: Novel sample generation conditioned on a type. Given 10 characters from an alphabet, our model produced 20 new samples.



(a) (b)

Figure 4.11: LPIPS values for each alphabet in the test set calculated from novel samples produced (a). There is a consistency between the results obtained from these models. L2-SSIM values for each alphabet in the test set calculated from novel samples produced (b).

CHAPTER 5

CONCLUSIONS

On the purpose of modeling compositionality and causality in human character generation, we propose a self-supervised reinforcement learning approach. We trained our model for unconditional generation and parsing on handwritten character datasets by defining a single action space and environment. Through experiments, we showed that, given the whole training set, our agent is able to capture the overall distribution and generate quality novel samples for the challenging Omniglot dataset. Then, we trained our agent for the parsing task; given a raster image, the goal is to reconstruct it through as few strokes as possible. We demonstrated that the parsing agent can be utilized for generating exemplars of a concept and creating novel samples conditioned on a type, without any further training, only difference being how it is called among tasks. To the best of our knowledge, we are the first to tackle these tasks with a self-supervised approach that operates on a stroke level.

As future work, we will consider two directions of improvement: (i) in order to increase performance and (ii) model a more realistic generation process. One drawback of our study is that as the smallest unit of sketching we use quadratic Bezier curves. However, for human-level generations, the stroke representations should be enhanced to capture more complex structures. To model a more human-level stroke generation process, the action space of the agent, hence the generator architecture, needs to be slightly modified. For instance, if we use B-spline curves with a varying length of control points to represent strokes, we can adopt an autoregressive structure where each control point is conditioned to the previous point till the model decides to 'finish' the curve. We anticipate that this curve formulation will increase the overall performance and introduce a further novelty to our current approach.

Bibliography

Aksan, E., Deselaers, T., Tagliasacchi, A., & Hilliges, O. (2020). Cose: Compositional stroke embeddings. *arXiv preprint arXiv:2006.09930*.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., & Parikh, D. (2015). Vqa: Visual question answering. *Proceedings of the IEEE international conference on computer vision*, 2425–2433.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. *International conference on machine learning*, 214–223.

Bellemare, M. G., Naddaf, Y., Veness, J., & Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, *47*, 253–279.

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, *24*.

Bhunia, A. K., Chowdhury, P. N., Yang, Y., Hospedales, T. M., Xiang, T., & Song, Y.-Z. (2021). Vectorization and rasterization: Self-supervised learning for sketch and handwriting. *arXiv preprint arXiv:2103.13716*.

Bin, H., Weihai, C., Xingming, W., & Chun-Liang, L. (2017). High-quality face image sr using conditional generative adversarial networks. *arXiv preprint arXiv:1707.00737*.

Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Cao, N., Yan, X., Shi, Y., & Chen, C. (2019). Ai-sketcher: A deep generative model for producing high-quality sketches. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 2564–2571.

Chang, C. H., Pallier, C., Wu, D. H., Nakamura, K., Jobert, A., Kuo, W.-J., & Dehaene, S. (2015). Adaptation of the human visual system to the statistics of letters and line configurations. *NeuroImage*, *120*, 428–440.

Changizi, M. A. (2001). Universal scaling laws for hierarchical complexity in languages, organisms, behaviors and other combinatorial systems. *Journal of Theoretical Biology*, *211*(3), 277–295.

Changizi, M. A., & Shimojo, S. (2005). Character complexity and redundancy in writing systems over human history. *Proceedings of the Royal Society B: Biological Sciences*, *272*(1560), 267–275.

Changizi, M. A., Zhang, Q., Ye, H., & Shimojo, S. (2006). The structures of letters and symbols throughout human history are selected to match those found in objects in natural scenes. *The American Naturalist*, *167*(5), E117–E139.

Chen, Y., Tu, S., Yi, Y., & Xu, L. (2017). Sketch-pix2seq: A model to generate sketches of multiple categories. *arXiv preprint arXiv:1709.04121*.

Cichy, R. M., & Kaiser, D. (2019). Deep neural networks as scientific models. *Trends in cognitive sciences*.

Custer, B. B. (1998). Position paper on appropriate audio/visual turing test. *New Methods in Language Processing and Computational Natural Language Learning*.

Das, A., Yang, Y., Hospedales, T., Xiang, T., & Song, Y.-Z. (2021). Cloud2curve: Generation and vectorization of parametric sketches. *arXiv preprint arXiv:2103.15536*.

Dehaene, S. (2010). *Reading in the brain: The new science of how we read*. Penguin Group USA.

Dehaene, S., & Cohen, L. (2007). Cultural recycling of cortical maps. *Neuron*, *56*(2), 384–398.

Dehaene, S., & Cohen, L. (2011). The unique role of the visual word form area in reading. *Trends in cognitive sciences*, *15*(6), 254–262.

Dehaene, S., & Dehaene-Lambertz, G. (2016). Is the brain prewired for letters? *Nature neuroscience*, *19*(9), 1192–1193.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, 248–255.

Dia, H., Tay, R., Kowalczyk, R., Bagloee, S., Vlahogianni, E., & Song, A. (2021). Artificial intelligence tests for certification of autonomous vehicles. *Academia Letters*, 3.

Edwards, H., & Storkey, A. (2016). Towards a neural statistician. *arXiv preprint arXiv:1606.02185*.

Ellis, K., Nye, M., Pu, Y., Sosa, F., Tenenbaum, J., & Solar-Lezama, A. (2019). Write, execute, assess: Program synthesis with a repl. *arXiv preprint arXiv:1906.04604*.

Feinman, R., & Lake, B. M. (2020a). Generating new concepts with hybrid neuro-symbolic models. *arXiv preprint arXiv:2003.08978*.

Feinman, R., & Lake, B. M. (2020b). Learning task-general representations with generative neuro-symbolic modeling. *arXiv preprint arXiv:2006.14448*.

Fenu, S., & Bagwell, C. (2016). Image colorization using residual networks.

Fountoukidou, T., & Sznitman, R. (2019). Concept-centric visual turing tests for method validation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 254–262.

Ganin, Y., Kulkarni, T., Babuschkin, I., Eslami, S. A., & Vinyals, O. (2018). Synthesizing programs for images using reinforced adversarial learning. *International Conference on Machine Learning*, 1666–1675.

Gao, H., Mao, J., Zhou, J., Huang, Z., Wang, L., & Xu, W. (2015). Are you talking to a machine? dataset and methods for multilingual image question answering. *arXiv preprint arXiv:1505.05612*.

Gao, Y., Gao, H., Luo, S., Zi, Y., Zhang, S., Mao, W., Wang, P., Shen, Y., & Yan, J. (2021). Research on the security of visual reasoning {captcha}. *30th {USENIX} Security Symposium ({USENIX} Security 21)*.

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

Geman, D., Geman, S., Hallonquist, N., & Younes, L. (2015). Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, *112*(12), 3618–3623.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Gregor, K., Besse, F., Rezende, D. J., Danihelka, I., & Wierstra, D. (2016). Towards conceptual compression. *arXiv preprint arXiv:1604.08772*.

Gregor, K., Danihelka, I., Graves, A., Rezende, D., & Wierstra, D. (2015). Draw: A recurrent neural network for image generation. *International Conference on Machine Learning*, 1462–1471.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*.

Ha, D., & Eck, D. (2017). A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Heo, Y.-J., Lee, M., Choi, S., Choi, W. S., Shin, M., Jung, M., Ryu, J.-K., & Zhang, B.-T. (2021). Toward a human-level video understanding intelligence. *arXiv preprint arXiv:2110.04203*.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, *30*.

Hewitt, L. B., Nye, M. I., Gane, A., Jaakkola, T., & Tenenbaum, J. B. (2018). The variational homoencoder: Learning to learn high capacity generative models from few examples. *arXiv preprint arXiv:1807.08919*.

Huang, Z., Heng, W., & Zhou, S. (2019). Learning to paint with model-based deep reinforcement learning. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8709–8718.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*.

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125–1134.

Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., & Girshick, R. (2017). Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2901–2910.

Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4401–4410.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kodali, N., Abernethy, J., Hays, J., & Kira, Z. (2017). On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

Lake, B., Salakhutdinov, R., & Tenenbaum, J. (2012a). Concept learning as motor program induction: A large-scale empirical study. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *34*(34).

Lake, B., Salakhutdinov, R., & Tenenbaum, J. (2012b). Concept learning as motor program induction: A large-scale empirical study. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *34*(34).

Lake, B. M. (2014). *Towards more human-like concept learning in machines: Compositionality, causality, and learning-to-learn* (Doctoral dissertation). Massachusetts Institute of Technology.

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2013). One-shot learning by inverting a compositional causal process.

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015a). Human-level concept learning through probabilistic program induction. *Science*, *350*(6266), 1332–1338.

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015b). Human-level concept learning through probabilistic program induction. *Science*, *350*(6266), 1332–1338.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and brain sciences*, *40*.

LeCun, Y. (1998). The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4681–4690.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., & Stoica, I. (2018). Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*.

Lin, H., Fu, Y., Xue, X., & Jiang, Y.-G. (2020). Sketch-bert: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6758–6767.

Lin, Q., Liang, L., Huang, Y., & Jin, L. (2018). Learning to generate realistic scene chinese character images by multitask coupled gan. *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, 41–51.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *European conference on computer vision*, 740–755.

Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep learning face attributes in the wild. *Proceedings of International Conference on Computer Vision (ICCV)*.

Lu, J., Ye, X., Ren, Y., & Yang, Y. (2019). Good, better, best: Textual distractors generation for multi-choice vqa via policy gradient. *arXiv preprint arXiv:1910.09134*.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., & Bousquet, O. (2017). Are gans created equal? a large-scale study. *arXiv preprint arXiv:1711.10337*.

Ma, L., Lu, Z., & Li, H. (2016). Learning to answer questions from image using convolutional neural network. *Thirtieth AAAI Conference on Artificial Intelligence*.

Malinowski, M. (2017). Towards holistic machines: From visual recognition to question answering about real-world images.

Malinowski, M., & Fritz, M. (2015). Hard to cheat: A turing test based on answering questions about images. *arXiv preprint arXiv:1501.03302*.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., & Paul Smolley, S. (2017). Least squares generative adversarial networks. *Proceedings of the IEEE international conference on computer vision*, 2794–2802.

Mazzone, M., & Elgammal, A. (2019). Art, creativity, and the potential of artificial intelligence. *Arts*, *8*(1), 26.

Mellor, J. F., Park, E., Ganin, Y., Babuschkin, I., Kulkarni, T., Rosenbaum, D., Ballard, A., Weber, T., Vinyals, O., & Eslami, S. (2019). Unsupervised doodling and painting with improved spiral. *arXiv preprint arXiv:1910.01007*.

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Murphy, G. (2004). *The big book of concepts*. MIT press.

Nakano, R. (2019). Neural painters: A learned differentiable constraint for generating brushstroke paintings. *arXiv preprint arXiv:1904.08410*.

Nakazawa, S., Han, C., Hasei, J., Nakahara, R., & Ozaki, T. (2021). Bapgan: Gan-based bone age progression of femur and phalange x-ray images. *arXiv preprint arXiv:2110.08509*.

Olague, G., Olague, M., Jacobo-Lopez, A. R., & Ibarra-Vazquez, G. (2021). Less is more: Pursuing the visual turing test with the kuleshov effect. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1553–1561.

Pagliarini, E., Scocchia, L., Vernice, M., Zoppello, M., Balottin, U., Bouamama, S., Guasti, M. T., & Stucchi, N. (2017). Children's first handwriting productions show a rhythmic structure. *Scientific reports*, *7*(1), 1–10.

Peterson, J. C., Abbott, J. T., & Griffiths, T. L. (2016). Adapting deep network features to capture psychological representations. *arXiv preprint arXiv:1608.02164*.

Qi, H., Wu, T., Lee, M.-W., & Zhu, S.-C. (2015). A restricted visual turing test for deep scene and event understanding. *arXiv preprint arXiv:1512.01715*.

Reddy, P., Gharbi, M., Lukac, M., & Mitra, N. J. (2021). Im2vec: Synthesizing vector graphics without vector supervision. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7342–7351.

Rezende, D., Danihelka, I., Gregor, K., Wierstra, D., et al. (2016). One-shot generalization in deep generative models. *International Conference on Machine Learning*, 1521–1529.

Ribeiro, L. S. F., Bui, T., Collomosse, J., & Ponti, M. (2020). Sketchformer: Transformer-based representation for sketched structure. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14153–14162.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. *Advances in neural information processing systems*, *29*, 2234–2242.

Sangkloy, P., Burnell, N., Ham, C., & Hays, J. (2016). The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)*, *35*(4), 1–12.

Schaldenbrand, P., & Oh, J. (2020). Content masked loss: Human-like brush stroke planning in a reinforcement learning painting agent. *arXiv preprint arXiv:2012.10043*.

Sharma, H., & Jalal, A. S. (2021). Image captioning improved visual question answering. *Multimedia Tools and Applications*, 1–22.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Song, Y.-Z. (2020). Bézier sketch: A generative model for scalable vector sketches. *Computer Vision–ECCV 2020*.

Souibgui, M. A., Biten, A. F., Dey, S., Fornés, A., Kessentini, Y., Gomez, L., Karatzas, D., & Lladós, J. (2021). One-shot compositional data generation for low resource handwritten text recognition. *arXiv preprint arXiv:2105.05300*.

Spivey, M., Joanisse, M., & McRae, K. (2012). *The cambridge handbook of psycholinguistics*. Cambridge University Press.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Tan, C., Leibo, J. Z., & Poggio, T. (2013). Throwing down the visual intelligence gauntlet. *Machine learning for computer vision* (pp. 1–15). Springer.

Testolin, A., Stoianov, I., & Zorzi, M. (2017). Letter perception emerges from unsupervised deep learning and recycling of natural image features. *Nature human behaviour*, *1*(9), 657–664.

Trick, L. M., & Pylyshyn, Z. W. (1994). Why are small and large numbers enumerated differently? a limited-capacity preattentive stage in vision. *Psychological review*, *101*(1), 80.

Turing, A. M., & Haugeland, J. (1950). *Computing machinery and intelligence*. MIT Press Cambridge, MA.

Van Oord, A., Kalchbrenner, N., & Kavukcuoglu, K. (2016). Pixel recurrent neural networks. *International Conference on Machine Learning*, 1747–1756.

Väth, D., Tilli, P., & Vu, N. T. (2021). Beyond accuracy: A consolidated tool for visual question answering benchmarking. *arXiv preprint arXiv:2110.05159*.

Vedd, N., Wang, Z., Rei, M., Miao, Y., & Specia, L. (2021). Guiding visual question generation. *arXiv preprint arXiv:2110.08226*.

Vogel, E. K., Woodman, G. F., & Luck, S. J. (2001). Storage of features, conjunctions, and objects in visual working memory. *Journal of experimental psychology: human perception and performance*, *27*(1), 92.

Wang, A., Ren, M., & Zemel, R. (2020). Sketchembednet: Learning novel concepts by imitating drawings. *arXiv preprint arXiv:2009.04806*.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, *8*(3), 229–256.

Yang, H., Xue, C., Yang, X., & Yang, H. (2021). Icon generation based on generative adversarial networks. *Applied Sciences*, *11*(17), 7890.

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

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018b). The unreasonable effectiveness of deep features as a perceptual metric. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zheng, N., Jiang, Y., & Huang, D. (2018). Strokenet: A neural painting environment. *International Conference on Learning Representations*.

Zhou, C., Li, C.-L., & Poczos, B. (2020). Unsupervised program synthesis for images using tree-structured lstm. *arXiv preprint arXiv:2001.10119*.

Zhou, T., Fang, C., Wang, Z., Yang, J., Kim, B., Chen, Z., Brandt, J., & Terzopoulos, D. (2018). Learning to sketch with deep q networks and demonstrated strokes. *arXiv preprint arXiv:1810.05977*.

Zou, Z., Shi, T., Qiu, S., Yuan, Y., & Shi, Z. (2020). Stylized neural painting. *arXiv preprint arXiv:2011.08114*.