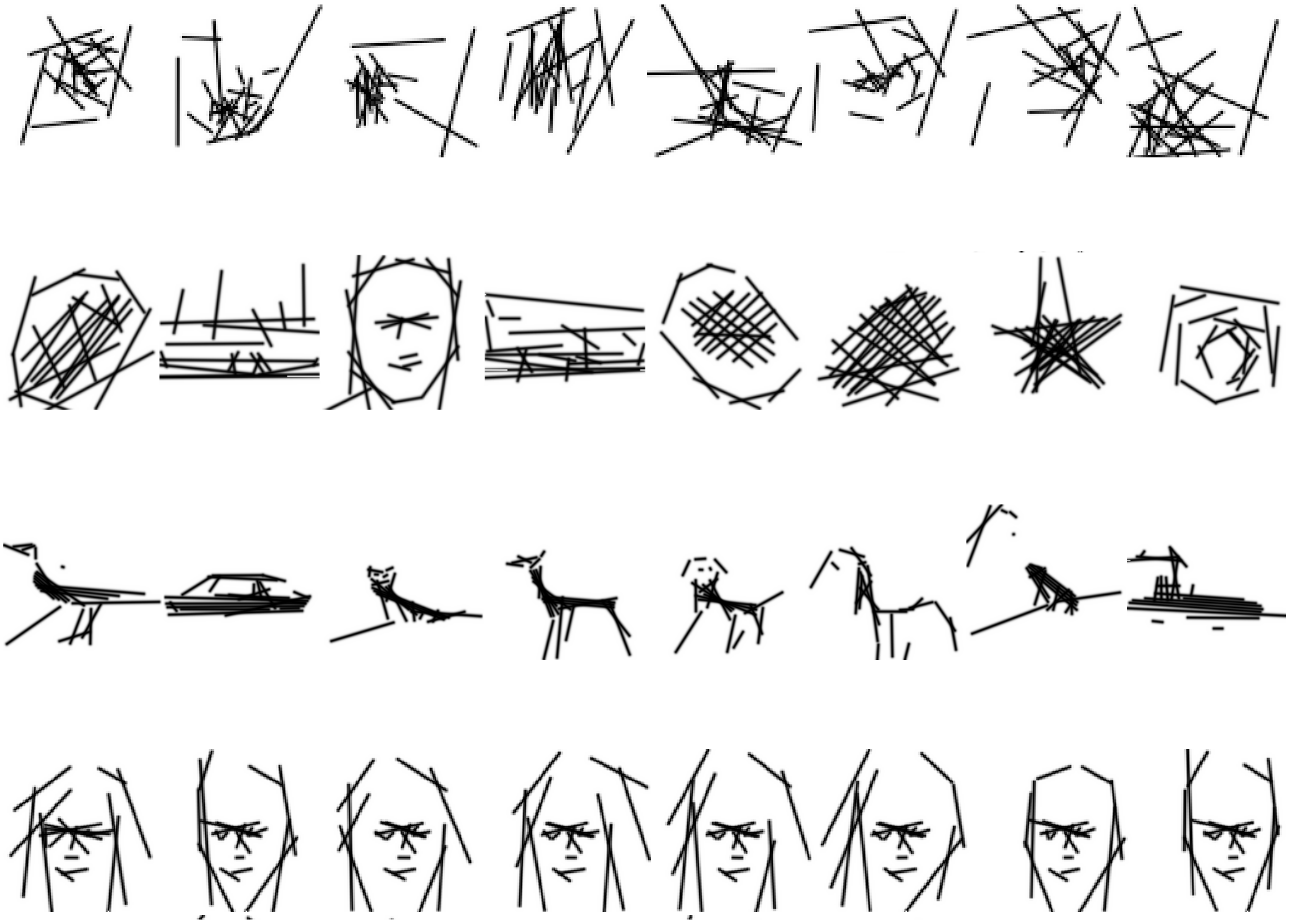


EMERGENT VISUAL COMMUNICATION

Daniela Mihai

University of Southampton

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

Emergent Visual Communication

by

Andreea Daniela Mihai
ORCiD: 0000-0003-3368-9062

A thesis for the degree of
Doctor of Philosophy

September 2022

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

Doctor of Philosophy

by Andreea Daniela Mihai

Our ability to *perceive*, *represent* and *understand* the surrounding visual world is one of the most fascinating, but equally intricate, parts of our nervous system. Supported by a sufficiently complex brain, we start to learn and develop these cognitive abilities and skills such as *communication* from the moment we are born. These capabilities are essential in numerous tasks we carry out in our daily life. The development of such intelligence is promoted by exploration of the environment and social interaction. As such, advancing artificial intelligent agents capable of *interaction* through communication with each other and with humans has been a long-standing goal. This thesis seeks to uncover how inter-agent communication about the visual world, emerging in a completely self-supervised way, can be modelled and its interpretability improved.

This research draws inspiration from how human communication developed and first compares the processes involved in transmitting meaningful information between humans and machines. In the context of referential signalling games played with realistic images, intelligent agents modelled as deep neural networks have previously been shown to develop successful token-based communication protocols to achieve a shared goal. This thesis analyses the factors which influence the emergence of *meaningful* protocols and shows that visual semantics can be learned in a self-supervised way.

Nonetheless, qualitative and quantitative insights into emergent token-based communication are not easily explainable to humans. We thus propose *drawing* as a communication channel which is a much simpler and more directly interpretable modality than language. To enable end-to-end learnable models of visual communication, a differentiable relaxation of the process of drawing vector primitives into pixel rasters is proposed. Using this approach, the physical act of drawing with a pen on paper can be modelled.

We then demonstrate that agents cooperating on a signalling game learn to communicate through sketching. An extensive analysis of the factors which influence the meaning and intent of agents' drawings is presented. The final two chapters show how interpretable sketches emerge when inducing visual perceptual similarity constraints. Through human evaluation of the emergent visual communication, we explore how, with appropriate inductive biases, artificial agents learn to draw in a fashion that humans can interpret.

Contents

Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:
    - Daniela Mihai and Jonathon Hare. Learning to draw: Emergent communication through sketching. In *Advances in Neural Information Processing Systems*, 2021.
    - Daniela Mihai and Jonathon Hare. Differentiable drawing and sketching. *arXiv preprint arXiv:2103.16194*, 2021.
    - Daniela Mihai and Jonathon Hare. Shared visual representations of drawing for communication: How do different biases affect human interpretability and intent? In *NeurIPS 2021 Workshop on Shared Visual Representations in Human and Machine Intelligence*, 2021.
    - Daniela Mihai and Jonathon Hare. Physically embodied deep image optimisation. In *NeurIPS 2021 Workshop on Machine Learning for Creativity and Design*, 2021.
    - Daniela Mihai and Jonathon Hare. Perceptions. Artwork exhibited at the *NeurIPS 2021 Workshop on Machine Learning for Creativity and Design*, 2021.

- Daniela Mihai and Jonathon Hare. The emergence of visual semantics through communication games. *arXiv preprint arXiv:2101.10253*, 2021.
- Daniela Mihai and Jonathon Hare. Avoiding hashing and encouraging visual semantics in referential emergent language games. In *NeurIPS 2019 Workshop on Emergent Communication*, 2019.

Signed:.. Date:..................

Acknowledgements

First and foremost, I want to thank Jon Hare for his continued support, guidance and friendship, without which this thesis would not have been possible. Thank you for modelling me into a competent researcher and sharing with me your passion not only for computer vision but also for human cognition and art, your ambition and zeal for perfection. Thank you for the countless meetings and long discussions over the last five years. For helping me think clearly, distil all the big ideas and place them in a wider context. Thank you for taking this journey with me!

I also have the EPSRC to thank for the support in funding this research and the VLC research group members with whom I have had meaningful discussions and made long-lasting friendships.

Thank you to my boyfriend and friends who looked through plenty of more or less terribly drawn sketches.

And above all, thank you to my parents and my sister for the infinite love and support.

Nomenclature

Network architectures

AlexNet	A GPU-implementation of a CNN by Alex Krizhevsky.
ANN	Artificial Neural Network.
BERT	Bidirectional Encoder Representations from Transformers.
CNN	Convolutional Neural Network.
DNN	Deep Neural Network.
GAN	Generative Adversarial Network.
GNN	Graph Neural Network.
GRU	Gated Recurrent Unit.
InfoGAN	An information-theoretic extension to the GAN.
LSTM	Long Short-Term Memory.
MLP	Multi-Layer Perceptron.
MT-DNN	Multi-Task Deep Neural Network.
NN	Neural Network.
RNN	Recurrent Neural Network.
SCAN	Symbol-Concept Association Network.
TCNN	Temporal Convolutional Neural Network.
VAE	Variational AutoEncoder.
VGG16	A very deep convolutional neural network with 16 weight layers.
β-VAE	A variant of the VAE which regulates the strength of the lower bound and the quality of the representations learnt.

Terms

AI	Artificial Intelligence.
BatchNorm	Batch Normalisation.
CKA	Centred Kernel Alignment.
CLIP	Contrastive Language–Image Pre-training.
CPC	Contrastive Predictive Coding.
CRS	Catmull-Rom Splines.
CTM	Computational Theory of the Mind.
DALL-E	An artificial intelligent system developed by OpenAI that can create realistic images from natural language descriptions.
DIAL	Differentiable Inter-Agent Learning.
ELBO	Evidence Lower Bound.
EoS	End of Sequence (token).
FE	Feature Extraction (network).
G-code	Computer programming language used mainly to control automated machine tools.
GEVD	Generalized Eigenvalue Decomposition.
GPU	Graphics Processing Unit.
ILSVRC	ImageNet Large Scale Visual Recognition Challenge.
img2sym	The process of generating symbols/concepts from particular visual samples.
MTL	Multi-Task Learning.
NLP	Natural Language Processing.
OCR	Optical Character Recognition.
OOV	Out-of-Vocabulary words.
RBF	Radial Basis Function.
REINFORCE	REward Increment = Non-negative Factor times Offset Reinforcement times Characteristic Eligibility. A class of reinforcement learning algorithms that falls under *Policy Gradient* methods.

RIAL Reinforced Inter-Agent Learning.

RTM Representational Theory of the Mind.

SimCLR Simple framework for Contrastive Learning of visual Representations.

SoS Start of Sequence (token).

ST-GS Straight-Through Gumbel Softmax, a reparameterisation that allows one to sample a categorical distribution from its logits.

sym2img The process of generating visual samples for particular symbols/concepts.

Datasets

Caltech-101 Image dataset of objects from 101 categories, between 40 to 800 coloured images per category. The size of each image is approximately 300 × 200 pixels.

CIFAR-10 80 million tiny images (32 × 32 pixel, coloured) dataset spanning 10 object classes.

COCO Common Objects in Context dataset by Microsoft. The release 2014 version contains 164K images from 80 classes split into training (83K), validation (41K) and test (41K) sets.

dSprites Disentanglement testing Sprites dataset contains 2D shapes generated from 6 ground truth independent latent factors (color, shape, scale, rotation, x and y positions of a sprite).

ImageNet Image dataset organised according to the WordNet concept hierarchy and designed for training large-scale object recognition models. It contains more than 14 million images spanning more than 20K categories, although the ILSVRC 2012 subset with 1.3M images and 1000 classes is most commonly used.

KMNIST Kuzushiji-MNIST dataset (28×28 pixel, greyscale). 10 classes represented by one character from each of the 10 rows of Japanese Hiragana alphabet (70K images total).

MNIST Handwritten digit dataset (28 × 28 pixel, greyscale) split into training (60K) and test (10K) sets.

Omniglot Image collection of 1623 different handwritten characters from 50 different alphabets (a training set of 30 alphabets and an evaluation set of 20 alphabets). Images were drawn online by 20 different people.

Quick Draw A collection of 50 million drawings across 345 categories, produced by players of the game ‘Quick, Draw’ by Google.

STL-10 An image recognition dataset, spanning 10 object classes from ImageNet (96 × 96 pixel, coloured), for unsupervised feature learning. 500 training images (10 pre-defined folds), 800 test images per class and 100000 unlabelled images for unsupervised learning.

Losses

BlurMSE Blurred Mean Squared Error, a single-scale version of MSE in which the input, and optionally the target are blurred by a Gaussian filter of a predetermined standard deviation.

CE Cross Entropy (loss).

LPIPS Learned Perceptual Image Patch Similarity, metric computed from deep neural network embeddings.

MM Multi-Margin loss, also known as hinge loss.

MSE Mean Squared Error.

SSIM Structural Similarity, a measure that compares local patterns of pixel intensities.

SSMSE Scale-Space Mean Squared Error, an MSE variant in which a scale-space is built for both the input and target, and the loss is accumulated over all levels.

Numbers and Arrays (scalars, vectors, matrices and tensors)

a A scalar (real or integer).

$\mathbf{a}$ A vector.

$\mathbf{A}$ A matrix.

$\mathbf{I}$ Identity matrix.

$\mathbf{I}_n$ $n \times n$ identity matrix.

$\mathbf{0}$ Vector, matrix or tensor of zeros (depending on context).

$\mathbf{1}$ Vector, matrix or tensor of ones (depending on context).

$\mathbb{1}_{[k \neq i]}$ Function that evaluates to 1 if and only if $k \neq i$ and is 0 otherwise.

Indexing

a_i Element i of vector $\mathbf{a}$, with indexing starting at 1.

$A_{i,j}$ Element i, j of matrix $\mathbf{A}$.

$\mathbf{A}_{i,:}$ Row i of matrix $\mathbf{A}$.

$\mathbf{A}_{:,i}$ Column i of matrix $\mathbf{A}$.

Sets, ranges, intervals and tuples

$\mathbb{A}$ A set.

$\mathbb{R}$ The set of all real numbers.

$\mathbb{Z}$ The set of all integer numbers.

$\mathbb{C}$ The set of all complex numbers.

$\mathbb{D}$ The set of all dual numbers.

$\{0, 1\}$ The set containing 0 and 1.

$\{0, \dots, n\}$ The set of all integers between 0 and n inclusive.

$[a, b]$ The *closed* real interval between a and b inclusive.

$[a, b)$ The *half-open* real interval between a inclusive and b exclusive.

$(a_1, \dots, a_n)$ The tuple or ordered sequence of elements a_i to a_n. Equivalent to the column vector $\begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix}^\top$.

Functions

$f : \mathbb{A} \to \mathbb{B}$ The function f with domain $\mathbb{A}$ and range $\mathbb{B}$. f maps an input taken from the set $\mathbb{A}$ to a element from the set $\mathbb{B}$.

$f(\mathbf{x}; \theta)$ or $f_\theta(\mathbf{x})$ A function of $\mathbf{x}$ parameterised by θ. Sometimes we will write $f(\mathbf{x})$, omitting θ to lighten notation.

$f \circ g$ Binary function composition of f and g. Equivalent to writing $f(g(\dots))$.

$\log(x)$ Natural logarithm of x.

$\exp(x)$ Exponential of x, e^x.

$\operatorname{argmax}(\mathbf{x})$ Arg-max or the index of the maximum value in the vector $\mathbf{x}$.

$\operatorname{ReLU}(x)$ Rectified Linear Unit activation, $\max(x, 0)$.

$\operatorname{sigmoid}(x)$ The sigmoid function, $\dfrac{1}{1 + \exp(-x)}$. Sometimes referred to as the logistic function.

$\|\mathbf{x}\|_p$ ℓ^p norm of $\mathbf{x}$.

$\|\mathbf{x}\|$ or $\|\mathbf{x}\|_2$ ℓ^2 norm of $\mathbf{x}$.

$[\mathbf{x}; \mathbf{y}]$ Concatenation of two vectors into a larger vector; can be extended to more than two vectors, e.g. $[\mathbf{x}; \mathbf{y}; \mathbf{z}]$.

Linear Algebraic Operations

$\mathbf{A}^\top$	Transpose of $\mathbf{A}$.
$\mathbf{A}^*$	Conjugate (or Hermitian) transpose of $\mathbf{A}$. If $\mathbf{A}$ is real, then $\mathbf{A}^* = \mathbf{A}^\top$.
$\mathbf{A}^{-1}$	Inverse of $\mathbf{A}$.
$\mathbf{A}^+$	Moore-Penrose inverse (pseudoinverse) of $\mathbf{A}$.
$\mathbf{A} \odot \mathbf{B}$	Hadamard product. Element-wise product of $\mathbf{A}$ and $\mathbf{B}$.
$\det(\mathbf{A})$	Determinant of $\mathbf{A}$.
$\dfrac{\mathrm{d}f(x)}{\mathrm{d}x}$	Leibniz's notation for the first derivative of $f(x)$ with respect to x.
$\dfrac{\mathrm{d}^2f(x)}{\mathrm{d}x^2}$	Leibniz's notation for the second derivative of $f(x)$ with respect to x.
f'	Langrange's notation for the first derivative of f. If f is a function of a single variable, then f' is the derivative with respect to that variable and f'' represents the second derivative.
$\dfrac{\partial y}{\partial x}$	Partial derivative of y with respect to x.
$\nabla_{\mathbf{x}} y$	Gradient (vector) of y with respect to $\mathbf{x}$.
$\nabla_{\mathbf{X}} y$	Matrix containing derivatives of y with respect to $\mathbf{X}$.
$\dfrac{\partial f}{\partial \mathbf{x}}$	Jacobian matrix $\mathbf{J} \in \mathbb{R}^{m \times n}$ of $f : \mathbb{R}^n \to \mathbb{R}^m$.

Probability

$D_{\mathrm{KL}}(P \parallel Q)$	Kullback-Liebler divergence of P and Q.
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$	Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$.
$\mathcal{N}(\mathbf{0}, \boldsymbol{I})$	Standard Normal distribution.
$a \sim P$	a is sampled from distribution P.
$\mathbb{E}_{x \sim P}[f(x)]$	Expectation of $f(x)$ with respect to $P(\mathrm{x})$.

Preface

"Of all of our inventions for mass communication, pictures still speak the most universally understood language."

— Walt Disney

"I dream my painting and I paint my dream."

— Vincent van Gogh

From the very moment we are born, we have to adapt to the environment and learn from sensory stimuli such as sight, hearing, touch, smell and taste. Babies learn to communicate about their needs by exploring and acting in the physical world, but also through interaction with social partners and guidance from more mature ones [Smith and Gasser, 2005]. The very first instincts developed by babies such as crying as a form of expressing hunger or discomfort, babbling and pointing at things they see before knowing what to name them are just incipient examples of interaction and communication by signalling and referencing the environment.

This is a thesis about artificial models of emergent visual communication. The research programme, motivated by how humans make sense of the visual world and have developed written communication, seeks to improve the interpretability of inter-agent communication systems.

Context

Creating intelligent machines that can perceive, understand the surrounding visual world, and communicate with us about it in natural language has been a long-standing goal of artificial intelligence research [Turing, 1950]. Humans have developed the ability to see and perceive the natural world, form internal representations and hence attribute meaning to objects, people and actions. These representations are essential when carrying out any day-to-day tasks or activities. Although to "see" and to "make sense" of the world we live in seems natural, instinctive and instantaneous, it is in reality extremely complex and involves a vast number of processes and mental abilities, all supported by the sufficiently advanced visual and nervous system as discussed in Chapter 1.

To see is to perceive visual information in an environment through the eyes, but it can also refer to one's ability to imagine or envisage a concept/object/idea in one's mind, *i.e.* to understand. Progress in deep learning, a major sub-field of machine learning, has led to the development of artificial models of cognition (see Section 1.2.2). Such deep models have been shown to emulate, to some extent, different parts of the human visual and nervous system and the connections between them [Harris et al., 2021; Yamins et al., 2014], as well as capabilities such as object recognition [Fan et al., 2018; Krizhevsky et al., 2012].

Building upon such models of artificial intelligence, a different but equally exciting direction of research has been focusing on the emergence of communication between artificial agents. Instead of embedding intelligent agents with human knowledge from text corpora, as usually done in the Natural Language Processing (NLP) field, the *emergent communication* research attempts to train machines to develop communication through interaction and, usually, cooperation on a shared goal (see Section 1.3.2.1). If the goal is to achieve *interactive* artificial agents, this approach resembles more the environment in which humans develop intelligence and acquire skills such as communication.

Previous studies in the field of emergent communication showed that artificial agents can successfully cooperate on a task by developing a shared communication protocol [Havrylov and Titov, 2017; Lazaridou et al., 2017, 2018; Bouchacourt and Baroni, 2018]. However, the "language" developed by machines without any human intervention is nothing like the natural language, the conceptual properties of the objects are not captured in the symbols communicated by agents and, hence, can't easily be interpreted by people [Bouchacourt and Baroni, 2018; Chaabouni et al., 2019; Lowe et al., 2019].

The aim of this thesis is to investigate how semantics, meaning and intent can be learned by artificial self-supervised models of communication. With the objective of improving communication interpretability and model explainability, this research evolved from exploring a token-based communication protocol (Chapter 2) to a different modality which is more intelligible for a human observer: that of *visual communication* presented in Chapter 4. To make this possible, an approach for differentiable drawing, which actually models the physical act of marking strokes with an instrument on paper as shown in Chapter 3 has been developed.

Contributions

This thesis brings together a number of contributions, particularly in the field of emergent visual communication. These contributions are listed next:

- Analysis of the factors which influence the emergence of visual semantics in a token-based communication protocol employed between artificial agents interacting on a shared goal (*e.g.* biases introduced by the visual feature extractor's weights, task objective, data augmentations - see Chapter 2).

- Demonstration that a token-based communication system which captures visual semantics can be learned in a completely self-supervised manner by playing the right types of game. The work presented in Chapter 2 bridges a gap between emergent communication research and self-supervised feature learning.

- In Chapter 3, a differentiable relaxation of the process of drawing vector primitives (*e.g.* points, lines, curves) into pixel rasters is proposed. We demonstrate how this framework can be integrated into end-to-end learnable models such as *autotracing autoencoders* that transform rasterised images into vectors without supervision. Further, vector sketches can be generated by directly optimising against photographs. Finally, we illustrate how the optimised sketching primitives can be applied to instruct a robot to physically draw.

- With the method for differentiable rasterisation, in Chapter 4 we present a framework for artificial agents, modelled as deep neural networks, that successfully learn to communicate through sketching. Considering drawing as a modality of conveying meaning instead of the traditional symbolic channel pioneers a new and exciting direction in the research field of emergent communication.

- In Chapters 4 and 5, an extensive analysis of the factors which affect the meaning and intent of the emergent visual communication is presented.

- Through the human evaluation experiments (see Chapters 4 and 5), we demonstrate that with the appropriate inductive biases, artificial agents can communicate through drawing in a fashion that humans can interpret.

Publications

This programme of research has led to a number of publications, produced independently or in collaboration with other researchers. The publications resulting from the research presented in this thesis are listed below:

- Daniela Mihai and Jonathon Hare. Learning to draw: Emergent communication through sketching. In *Advances in Neural Information Processing Systems*, 2021.

- Daniela Mihai and Jonathon Hare. Differentiable drawing and sketching. *arXiv preprint arXiv:2103.16194*, 2021.

- Daniela Mihai and Jonathon Hare. Shared visual representations of drawing for communication: How do different biases affect human interpretability and intent? In *NeurIPS 2021 Workshop on Shared Visual Representations in Human and Machine Intelligence*, 2021.
- Daniela Mihai and Jonathon Hare. Physically embodied deep image optimisation. In *NeurIPS 2021 Workshop on Machine Learning for Creativity and Design*, 2021.
- Daniela Mihai and Jonathon Hare. Perceptions. Artwork exhibited at the *NeurIPS 2021 Workshop on Machine Learning for Creativity and Design*, 2021.
- Daniela Mihai and Jonathon Hare. The emergence of visual semantics through communication games. *arXiv preprint arXiv:2101.10253*, 2021.
- Daniela Mihai and Jonathon Hare. Avoiding hashing and encouraging visual semantics in referential emergent language games. In *NeurIPS 2019 Workshop on Emergent Communication*, 2019.

The following publications resulted from collaborations during this PhD programme, but are not directly related to this thesis:

- Ethan Harris, Daniela Mihai, and Jonathon Hare. How convolutional neural network architecture biases learned opponency and color tuning. *Neural Computation*, 33 (4):858–898, 2021.
- Ethan Harris, Daniela Mihai, and Jonathon Hare. Anatomically constrained ResNets exhibit opponent receptive fields; So what? In *NeurIPS 2020 Workshop on Shared Visual Representations in Human and Machine Intelligence*, 2020.
- Ethan Harris, Daniela Mihai, and Jonathon Hare. Spatial and colour opponency in anatomically constrained deep networks. In *NeurIPS 2019 Workshop on Shared Visual Representations in Human and Machine Intelligence*, 2019.

Thesis Structure

To conclude the preface of this thesis, the structure and content are outlined below.

Chapter 1 - Representing, Understanding and Communicating Information. To start, we introduce the background of the main components involved in the emergence of communication about visual stimuli in humans and machines. This chapter offers an overview of the complexity of achieving artificial agents that can see and communicate with each other and potentially with humans about visual information. It also describes existing research towards achieving these goals.

Chapter 2 - Communication with Tokens. Starting from previous studies on the emergence of token-based communication protocols, this chapter investigates the biases existent in the artificial models' pretraining methods, visual data augmentations and various task objectives that could prevent models from learning a hashing-like communication and encourage a more human-like protocol (*i.e.* as indicated by metrics developed to quantify semantics emergence).

Chapter 3 - Differentiable Drawing and Sketching. In this chapter, a framework for differentiable drawing is proposed. We define a bottom-up differentiable relaxation of the process of drawing points, lines and curves on a pixel raster. We demonstrate that this framework can be integrated in end-to-end differentiable programs and deep networks, and thus be learned and optimised. Moreover, we exhibit how the optimised drawing primitives can then be translated into commands which instruct a robot to physically draw images using drawing instruments such as pens and pencils on a support medium.

Chapter 4 - Communication through Sketching. In this chapter, we explore a visual communication channel between agents playing a referential game. Instead of predefined token vocabularies used in existing research, the artificial agents in this study are allowed to transmit information by drawing with simple strokes. Our agents are parameterised by deep neural networks and the drawing procedure is differentiable, allowing for end-to-end training. We demonstrate that agents can not only successfully learn to communicate by drawing, but with appropriate inductive biases, can do so in a fashion that humans can interpret.

Chapter 5 - How Do Different Biases Affect Human Interpretability and Intent? In this chapter, an investigation into how representational losses can affect the drawings produced by artificial agents playing a communication game is presented. We show that a combination of powerful pretrained encoder networks, with appropriate inductive biases, can lead to agents that draw recognisable sketches, whilst still communicating well. Further, using the technique of prompt engineering, we automatically analyse the semantic content being conveyed by a sketch and demonstrate that current approaches to inducing perceptual biases lead to a notion of objectness being a key feature despite the agent training being self-supervised.

Chapter 6 - Conclusions. The overall findings and contributions of this thesis are discussed with respect to the original aims and objectives highlighted in this preface. Then, a series of open-ended questions arising from this research programme are discussed. The chapter ends with a personal view of where the field of emergent communication is heading in the future, its applications and opportunities.

Chapter 1

Representing, Understanding and Communicating Information

"The pen is the tongue of the mind."

— Horace

"Art is a technique of communication. The image is the most complete technique of all communication."

— Claes Oldenburg

In the process of making sense of visual information one of the first and most important steps is the act of *seeing*. But what does it mean to see? According to Aristotle and many others after him studying vision, "to see" is to perceive what is where by looking. In Marr [1982]'s study, vision is, first and foremost, an information-processing task. In order for one to know what is where, that is to process the information, one first needs to be able to represent the perceived information in their mind. However, visual perception is much more than the processing of sensory inputs, it is also shaped by personal memories and experiences [von Helmholtz, 1925].

This thesis is about artificial intelligence that can learn to *represent* and *understand* visual information in order to *communicate* about it. This problem, broken down into parts, spreads across several multi-faceted fields ranging from the (human) visual system to information theory, from cognitive development and language evolution literature to artificial neural models of representation and communication. As these fields are all incredibly vast, the aim of this chapter is not to cover them in depth. Rather we discuss and compare the processes involved in the task, between humans and machines, and describe the techniques and prior art used throughout this thesis.

This chapter begins by discussing the twin strands of visual perception: representation and processing, and their duality. Section 1.3 then reviews the topic of communication, in particular the task of communicating about visual information. Finally, Section 1.3.3 addresses the importance of gameplay for learning and its influence on tasks such as communication.

1.1 Representing Information

First, the concepts of information and representation need to be established in the context of this thesis. There exists comprehensive literature on information theory [Shannon, 1948; Karnani et al., 2009; Adami, 2012]. At a broad level, information can be regarded as something which naturally resides in the external world and is involved in the creation of internal representations. It can be seen as a link between the natural world and a receptor such as the human brain [Ramos, 2014].

Any type of external stimuli such as light, sound or pressure, can act as an information source for sensory cells. These cells in turn generate action potentials also known as nerve impulses, which are essential information encoded as electrical or chemical signals transmitted to the following neurons in the pathway of the nervous system or other parts of the body [Hodgkin and Huxley, 1952]. For the purpose of this thesis, the focus of the discussion will be on *visual* stimuli, on how visual information is perceived and represented.

Representations are thought to be a special case of information which resides internally, in one's brain. The information originating from the outside world is transformed through a set of complex mental processes which give rise to various representations. Information and representations are sometimes presented as interdependent and co-varying concepts [Ramos, 2014].

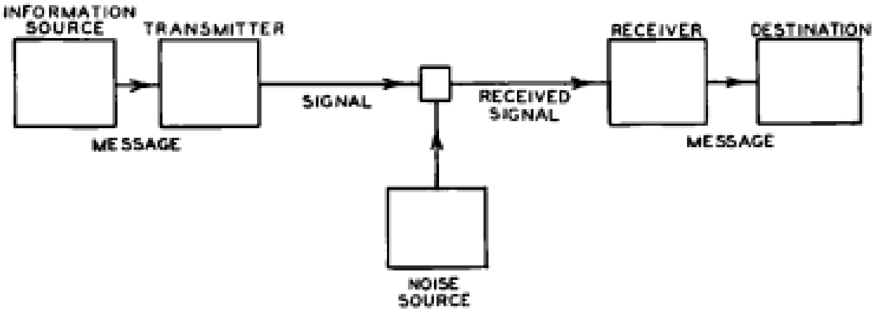


FIGURE 1.1: **Shannon's mathematical model of communication.** Image sourced from Shannon [1948].

In neuroscience, there also exists the topic of semantic information which studies the question of how information acquires meaning [Floridi, 2005]. Based on one of the most popular theories of information, Shannon [1948]'s mathematical model of communication, shown in Figure 1.1, suggests that the meaning, or what makes the information significant, occurs only when this reaches the receiver.

1.1.1 How do humans represent information?

As previously mentioned, the focus of this thesis is on visual information. The **human visual system** is responsible for perceiving and processing visual events. This is an extremely complex system. A simplified schematic of it, shown in Figure 1.2, consists of the retina, the optic nerves, the lateral geniculate nucleus (LGN) and the primary visual cortex, followed by higher-level visual areas. The information is first received by light-detecting cells which reside in the human retina. These photoreceptor cells, divided into two types, rods and cones, convert light into electrochemical signals. According to classic literature on the mammalian visual system, all the information about visual events is transported from the retina to the central nervous system by optic nerve fibres, passing through various bottlenecks and processes along the way [Harris et al., 2021].

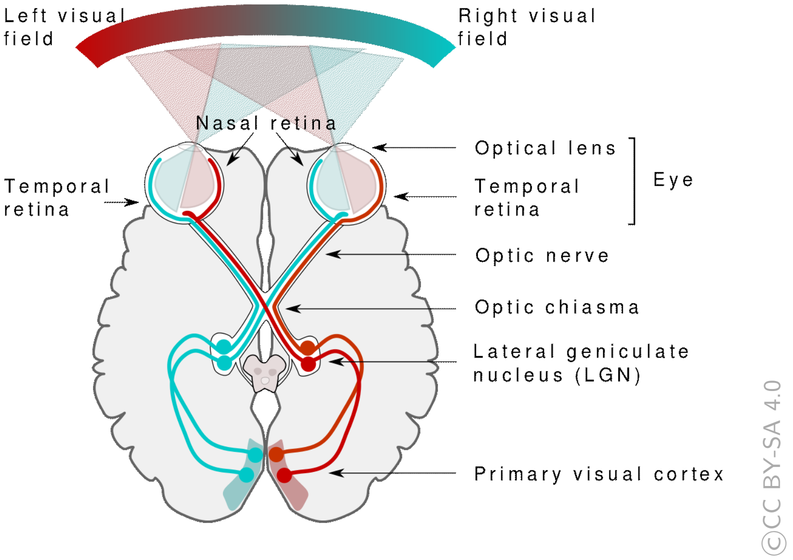


FIGURE 1.2: **A simplified schematic of the human visual pathway.** Image sourced from `https://bit.ly/3GFuP7c`.

The reception of visual information through light stimuli is just the first step. The formation of mental representations is possible because of the sufficiently complex human brain structure that allows internal mental states to associate and vary with the occurrence of outside events [Ramos, 2014]. A mechanism for constant validation against the environment is then required to ensure useful representations of the world. One such mechanism could be natural selection which helped shape human cognition over time [Tononi, 2008].

The concept of mental representations and how these are formed has been extensively studied and debated since antiquity by philosophers, and later on, reiterated by psychologists, cognitive and computational scientists. Of the many theories, it is worth mentioning two leading views which are sometimes used interchangeably, although there exists a shift in focus: the Representational Theory of the Mind (RTM) [Fodor, 1975, 1983, 1987] and the Computational Theory of the Mind (CTM) [McCulloch and Pitts, 1943; Turing, 1950; Marr, 1982]. The two theories are related in the sense that they both regard states of the mind as representations.

At a general level, a mental representation according to Pitt [2020] can be described as an object together with semantic properties such as reference (*i.e.* what that representation is about), appropriateness, truth and accuracy [Ramos, 2014; Cummins et al., 1996]. The RTM conceives of the mind as being mental states and processes, in relation to systems of internal representations. Mental states are characterised by what internal representation *are about* or *refer to*, while the mental processes are the way by which such internal representations are obtained and how they interact [Marr, 1982].

The Computational Theory of the Mind, as the name suggests, associates the brain with a computer and the mental process with computation. Then it is further divided into two branches, classical and connectionist. The former considers that mental representations are symbolic structures while the latter regards them as patterns of activation in a network of simple processing units. Likewise, mental processes are for the former the manipulation of the constituent elements according to some rules [Turing, 1950; Marr, 1982], and for the latter, they are represented by the spreading of the activation patterns [McCulloch and Pitts, 1943; Rumelhart, 1989]. The artificial neural network (ANN) is the most well-known connectionist model nowadays.

According to the CTM, mental representations have been modelled under various names and types such as “mental models” [Johnson-Laird, 1983], “retinal arrays” and “sketches” [Marr, 1982], “frames” [Minsky, 1974] or symbolic structures [Smolensky, 1989]. For example, Marr proposed the idea of a sequence of representations in the brain [Marr, 1982]. In the case of human vision, the very first representation can be modelled as an array of light intensities detected by photoreceptor cells in the retina. The subsequent representations are the primal sketch and “$2^{1}/_{2}$-dimensional ($2^{1}/_{2}D$) sketch”, which are both viewer or retinal centred representations in the sense that they describe properties such as contours, discontinuities, surface orientation and depth in relation to the viewer. Finally, the $3D$ model describes shapes and surfaces from an object-centred coordinate system. Minsky [1974], on the other hand, proposes the notion of “frame” as a data structure for describing stereotyped situations, which can be looked at as a network of nodes and connections that holds various kinds of information.

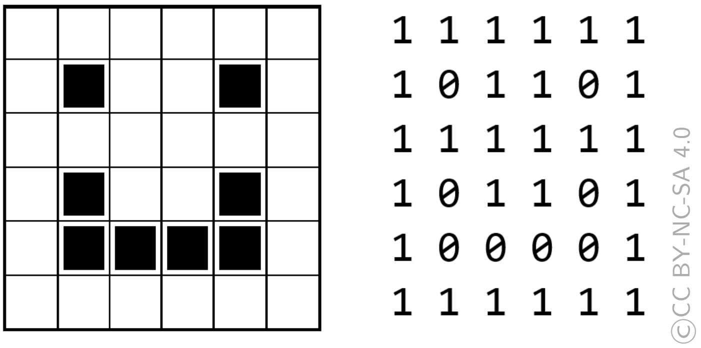
All these theories about mental representations show the diversity of ways in which information can be represented and understood. However, this is still an ongoing field of study. The differentiation that matters the most to our discussion about visual representations and how these can be computationally achieved is whether it should be as a discrete, serial, symbolic structure or as a continuous, parallel-processed, visually-analogous type. For example, should the visual object "brown horse" be represented as two consecutive discrete symbols denoting the two components? Or should it be represented as a visually similar drawing as one does, for example, when playing Pictionary?

1.1.2 How can information be modelled computationally?

In the thriving research field of Artificial Intelligence (AI), the aim over the years has been to build machines that can think, tackle and solve various tasks. Some of these are straightforward to computers such as those that involve mathematical rules, while others are more ambiguous, although intuitive for humans, such as the generation and use of natural language.

For any type of information to be useful to a machine, it needs to be in a machine-readable format. As all computers work in binary, all data such as images, videos, text and sound needs to be converted to a binary form, that is a series of 0s and 1s. A digital image, for example, is represented as a bitmapped graphic which is constructed as a grid of pixel values (*e.g.* Figure 1.3). In the case of text, for a computer to understand and process a letter, this needs to be converted to a binary number according to rules specified by a code (*e.g.* in the ASCII code, the letter "a" is assigned the binary number 0110 0001).

In the field of AI and machine learning, the various types of data have known different representations, each suited to the application or algorithm it was meant for. The remainder of this section gives an overview of some representations of the two data modalities that this thesis is focused on: language (*e.g.* text) and visual input (*e.g.* images).

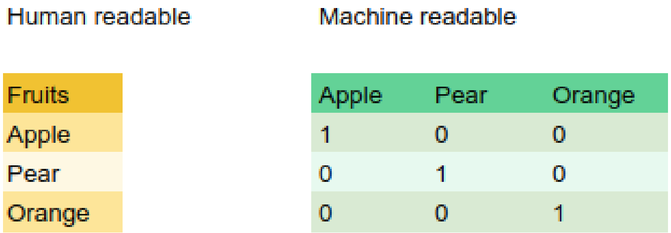


```
1 1 1 1 1 1
1 0 1 1 0 1
1 1 1 1 1 1
1 0 1 1 0 1
1 0 0 0 0 1
1 1 1 1 1 1
```

FIGURE 1.3: **An example of a black and white graphic in which each pixel is stored as one bit: black is 0 and white is 1.** Image sourced from `https://bit.ly/3zlyxBO`.

1.1.2.1 Representing language

As computers operate on numbers, written language, *i.e.* text, needs to be first converted to a numeric format. One of the classical models of word representation is the **one-hot vector encoding**. For a vocabulary of size L, a unique word in that vocabulary can be represented as an L-dimensional vector in which the element at one unique position is set to 1 and all others are set to 0 (*e.g.* Figure 1.4). This method is known to miss connections between words, as well as context information, and becomes ineffective for large vocabularies [Naseem et al., 2021].



Human readable

Fruits
Apple
Pear
Orange

Machine readable

Apple	Pear	Orange
1	0	0
0	1	0
0	0	1

FIGURE 1.4: **An example of one-hot encoding.** In a machine readable format, each fruit can be represented as a unique vector of 0s and 1s.

A **Bag-of-Words** language model represents text as an unordered set of words, discarding the original position of the word in the text, but retaining its frequency of occurrence. It is an extension of the one-hot encoding as it simply adds up the one-hot representations. This model has applications in sentiment analysis and language detection as the term frequency (TF) is an indicative factor for the task. A different language representation is given by the **N-gram** which is a sequence of N consecutive tokens. The N-grams representing a document can overlap and, depending on the level of granularity, a token can represent a word or an individual character. A word **N-gram** language model estimates the probability of the next word in a sequence of length N based on the previous words [Fürnkranz, 1998]. To represent semantics and capture similarities between words, the concept of **context** is important. The **word-document representation** takes into account the context, *i.e.* the document, in which the word occurs. This model uses a term-document matrix which keeps count of the word occurrences in each document, such that each document can then be represented as a vector (given by each column in the matrix). Other models in this line of work are the word-word model and the Term Frequency-Inverse Document Frequency (TF-IDF) model.

From the more recent approaches for learning vector space representations of words it is worth noting Word2vec [Mikolov et al., 2013], Global Vectors (GloVe) [Pennington et al., 2014] and FastText [Bojanowski et al., 2017]. Word2vec addresses the problem of word representations lacking semantic and syntactic information. This model, given a corpus of text, uses a classifier such as a neural network to produce a vector space. Simply put, it predicts an embedding (*i.e.* a dense vector) for each unique word in this space. Both its variants, CBOW and skip-gram denoting the network architecture, produce word vectors positioned close in the vector space for words which occur in common local context [Mikolov et al., 2013]. GloVe addresses Word2vec's drawback of utilising only the local context to predict word embeddings, and instead incorporates the global statistical information. It directly optimises the word embeddings according to a function which takes into account the number of times two words occur next to each other anywhere in the text corpus [Pennington et al., 2014]. Finally, FastText proposed by Bojanowski et al. [2017] addresses the problem of out-of-vocabulary words (OOV, *i.e.* words not present in the vocabulary or the training text corpus) which the previous two algorithms do not tackle. This language model breaks the text corpus into n-grams of characters instead of full words before passing them as input to a neural network. This solution allows for relationships between groups of characters, such as prefixes and suffixes, to be picked up by the network and used when encountering OOV words. As a comprehensive discussion of these models is beyond the scope of this thesis, further details can be found in the review by Naseem et al. [2021].

1.1.2.2 Representing visual input

As in the case of text, visual input (in the form of digital images or video frames) which is to be used by a computer or a machine learning algorithm to learn from first needs to be converted from the bitmapped graphic format (see Figure 1.3) to a vector of numbers representing the pixel values. This vector can then be further processed by machine learning algorithms for image classification or pattern recognition. One of the most popular such approaches is the Convolutional Neural Network (CNN) which is a specialised class of ANN that utilises the *convolution* operation in one or more of its layers [Goodfellow et al., 2016]. The layers of a CNN provide intermediate representations for a visual input, each describing various features. These intermediate representations, known as feature vectors or feature maps, are numerical vectors or tensors which capture discriminative information crucial for the task. For an image input, the feature maps from a CNN distinguish visual attributes such as objects' edges and direction, texture patterns, shapes and salient regions in numerical form.

1.1.2.3 Models of representation

Many machine learning algorithms are heavily influenced by the representation of the data they are given [Goodfellow et al., 2016; Higgins et al., 2017a]. For example, in the case of cat recognition in photographs, the presence of ears and tail is an important feature, but explaining it in terms of pixel values is a difficult problem. Bengio et al. [2012] claimed that in order for AI to understand the world around us, it must first learn to disentangle the generative factors of the data.

Representation learning algorithms are one solution to this problem. Learned representations, unlike hand-designed representations, enable AI systems to reach better performance without the expense of human time and effort. In a disentangled representation, single latent units are sensitive to changes in single generative factors and invariant to changes in other factors [Bengio et al., 2012]. Moreover, knowledge about the factors of variation in the data can boost generalisation performance [Higgins et al., 2017a].

The remainder of this section provides an overview of some of the most well-known, as well as state-of-the-art, approaches for learning, potentially interpretable and disentangled, representations. One thing that these representation learning models have in common is that they can all be viewed as a combination of an encoder module and a decoder module. Consequently, looking at this field from an encoder-decoder perspective, representation learning corresponds to learning the encoder part, while the decoder coupled with the loss function helps enforce useful representations.

Autoencoder. One of the earliest representation learning algorithms is the autoencoder [Hinton and Zemel, 1994]. This is a type of ANN which has an encoder whose function is to convert the input data into an efficient encoding of smaller dimensionality, and a decoder which converts the new representation back to the original input format and size. Traditionally, the autoencoder's reduction and reconstruction functions were used to learn meaningful representations in an unsupervised manner. More recently, connections made between autoencoders and latent variable models have brought autoencoders to the forefront of generative modelling [Kingma and Welling, 2014; Higgins et al., 2017a].

Variational autoencoder and β-VAE. The Variational Autoencoder (VAE) has a fundamental property which distinguishes it from the standard autoencoder and makes it very useful for generative modelling: the learned latent space is continuous, and hence, allows random sampling and interpolation. Instead of mapping the input to a fixed vector, the encoder network outputs two vectors, a mean vector $\boldsymbol{\mu}$ and a (typically diagonal) covariance matrix $\boldsymbol{\Sigma}$ of the distribution that the decoder is exposed to. Therefore, it learns to decode not only a specific encoding of the input but a range of variations of the same input's encoding [Kingma and Welling, 2014]. VAEs are trained to maximise a variational approximation through the use of the evidence lower bound (Equation 1.1 with $\beta = 1$). It is possible to train VAEs with gradient descent using what has become popularly known as the *reparameterisation trick* [Kingma and Welling, 2014]. Broadly speaking, the reparameterisation trick allows one to compute gradients with respect to the parameters of a distribution being sampled by factoring out those parameters from the stochastic variable (*e.g.* $\boldsymbol{y} = \boldsymbol{\mu} + \boldsymbol{\Sigma}\boldsymbol{z}$ where $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ in the case of a multivariate normal distribution).

β-VAE is a framework which improves the VAE by introducing a hyper-parameter β that regulates the strength of the lower bound and also the quality of the representations learnt by the model [Higgins et al., 2017a]. Its loss function is defined as:

$$L_{BETA}(\phi, \beta) = -\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z}) + \beta D_{\mathrm{KL}}(q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p_\theta(\mathbf{z})) \tag{1.1}$$

This maximises the probability of generating real data while keeping the distance between the real and estimated posterior distributions small. When $\beta = 1$, Equation 1.1 is equivalent to the loss function of the standard VAE, and when $\beta > 1$, it limits the representation capacity of the latent variable $\mathbf{z}$ and further encourages disentanglement. However, trading reconstruction accuracy for disentanglement also means losing information, and eventually semantics which we are interested in capturing.

InfoGAN. InfoGAN is a scalable unsupervised approach for disentangled factor learning [Chen et al., 2016]. It is based on maximising the mutual information between a small subset of latent variables and observations within the Generative Adversarial Network (GAN) framework [Goodfellow et al., 2014]. A GAN has two components, a generator network that generates fake samples from a data distribution and a discriminator network that aims to distinguish between the real and the fake samples. Unlike previous approaches, which require supervision and cannot learn a latent code for unlabelled variation, InfoGAN automatically discovers salient latent factors of variation. The generator decomposes the input vector into the source of noise ($\boldsymbol{z}$) and the latent code ($\boldsymbol{c}$), which targets the structured semantic features, and hence becomes $G(\boldsymbol{z}, \boldsymbol{c})$. The original GAN tends to learn trivial codes by finding a solution which only satisfies $P_G(\boldsymbol{x} \mid \boldsymbol{c}) = P_G(\boldsymbol{x})$ and ignores the additional latent code $\boldsymbol{c}$. InfoGAN solves this problem with an information-theoretic regularisation which encourages high mutual information between latent codes $\boldsymbol{c}$ and generator distribution $G(\boldsymbol{z}, \boldsymbol{c})$ [Chen et al., 2016]. Nevertheless, we are interested in encoders which can learn factorised and interpretable representations, rather than generators which can produce semantically disentangled samples.

Latent translation between generative models. Despite the advances of generative models such as VAEs and GANs, there has been no straightforward way to combine predefined modules. Retraining these models for each new problem becomes infeasible. Tian and Engel [2019] explore the problem of cross-modal domain transfer as a way to combine trained models to solve new tasks. Domain transfer can be enabled by deep generative models that learn a mapping between data domains such that *locality* is preserved, *i.e.* variations in one domain are reflected in the other. The approach proposed by Tian and Engel is based on a shared "bridging" VAE that can transfer between the latent spaces of pretrained models. In addition to the ELBO, the objective of this VAE includes a sliced-Wasserstein distance and a classification loss in the shared latent space which encourage locality and semantic alignment. Their method enables transfer within a modality (image-to-image), between different modalities (image-to-audio) and between latent generative models (VAE-to-GAN).

SCAN: Learning abstract hierarchical compositional visual concepts. Higgins et al. [2017b] emphasise once more that the natural world is infinitely diverse and that it is essential for intelligent systems to represent knowledge as abstract concepts which can be combined, re-combined and hierarchically organised. The Symbol-Concept Association Network (SCAN) learns such concepts in the visual world. The proposed approach is based on the β-VAE which learns a disentangled latent representation of the visual world. SCAN is then trained to extract meaningful abstractions over these disentangled primitives of the data. This is achieved by exposing it to symbol-image pairs that apply to a particular concept. Once a concept is acquired, SCAN allows bi-directional inference (sym2img and img2sym) [Higgins et al., 2017b]. The limitation of this method is that it only works on simple datasets such as dSprites [Matthey et al., 2017] which contain a limited number of objects and do not represent real-life scenes.

1.1.2.4 Self-supervised and multi-task learning approaches

Among a variety of unsupervised approaches, the self-supervised learning framework is one of the most successful as it uses pretext tasks such as image inpainting [Pathak et al., 2016], predicting image patches location [Doersch et al., 2015] and image rotations [Gidaris et al., 2018]. Such pretext tasks allow for the target objective to be computed without supervision and require high-level image understanding. As a result, high-level semantics are captured in the visual representations which are used to solve the tasks. Kolesnikov et al. [2019] provide an extensive overview of this research topic.

Over the last few years, some of the most successful self-supervised algorithms for visual representation learning have used the idea of contrasting positive pairs against negative pairs. A *contrastive loss* function minimises the distance between positive pairs of data samples, *i.e.* of the same class, and maximises it otherwise. Unlike autoencoder-based approaches, Hénaff et al. [2019] tackles the task of representation learning with an unsupervised objective, Contrastive Predictive Coding (CPC) [van den Oord et al., 2018], which extracts stable structure from still images. When applied to visual data, CPC learns by predicting the representation of patches below a certain level from those above it. This way, it overcomes the problem of representing all patches with a constant feature vector, which usually occurs in approaches with a Mean Squared Error (MSE) loss. Similarly, Ji et al. [2018] presents a clustering objective that maximises the mutual information between class assignments for pairs of images. They learn a neural network classifier from scratch which directly outputs semantic labels, rather than high dimensional representations that need external processing to be used for semantic clustering. Other

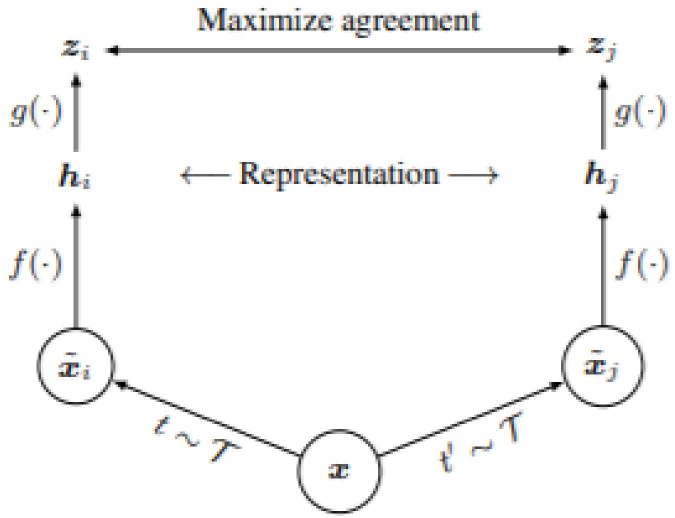


Figure 1.5: **The SimCLR framework**: two separate data augmentations, sampled from the same family, are applied to each data example to obtain two different views. Then the encoder $f()$ and the projector $g()$ network are trained to maximise the agreement between the latent vectors z_i and z_j via a contrastive loss. After training, the projection head is removed and the representations h are used for downstream tasks. Image sourced from Chen et al. [2020].

successful cases of unsupervised feature learning based on clustering methods, such as DeepCluster [Caron et al., 2018] jointly learn the parameters of a neural network and the clustering assignments of the resulting features.

Despite the surge of interest, Chen et al. [2020] have shown through the strength of their approach that self-supervised learning still remained undervalued. They proposed a simple framework, *SimCLR* illustrated in Figure 1.5, for contrastive visual representation learning. SimCLR learns meaningful representations by maximising the similarity between differently augmented views of the same image in the latent space. This is done via the following contrastive loss function:

$$l_{i,j} = -\log \frac{exp(sim(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} exp(sim(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}, \tag{1.2}$$

where $sim(\boldsymbol{z}_i, \boldsymbol{z}_j)$ represents the cosine similarity between the latent vectors, τ denotes a temperature parameter, and the function $\mathbb{1}_{[k \neq i]}$ evaluates to 1 if and only if $k \neq i$. One of the main contributions of this work is that it outlines the critical role of data augmentations in defining effective tasks to learn useful representations. We will also explore this framework in some of our experiments detailed in Section 2.5 and 2.7. Nonetheless, it is worth mentioning that trends in self-supervised learning have since shifted from contrastive methods to approaches that apply the principle of redundancy reduction [Zbontar et al., 2021; Bardes et al., 2022].

Finally, our attempt at encouraging semantically aligned image representations in this thesis is most similar to previous works which combine multiple pretext tasks into one self-supervised objective [Chen et al., 2019; Doersch and Zisserman, 2017]. Multi-task learning (MTL) rests on the hypothesis that people often apply knowledge learned from previous tasks to learn a new task. Similarly, when multiple tasks are learned in parallel using a shared representation, knowledge from one task can benefit the other tasks [Caruana, 1997]. MTL has proved itself useful in language modelling for models such as BERT [Devlin et al., 2018] which obtains state-of-the-art results on eleven natural language processing tasks. More recently, Radford et al. [2019] combine MTL and language model pretraining and propose MT-DNN, a model for learning representations across multiple natural language understanding tasks. In this thesis, we are also interested in the effect of solving multiple tasks on the semantics (see Section 2.6) captured by a fully learned feature extractor network.

1.2 Understanding Representations

As one might think, understanding is the next natural step in the sequence of processes performed by humans when acting in the world and ceaselessly receiving information through sensations and perception of various stimuli. However, as David Marr highlighted in his study of vision, the action of processing and attributing meaning to information is indivisible from the formation of internal representation [Marr, 1982]. It is the internal representation that provides the basis for our thoughts, but it is the ability by which we extract meaningful information, that is to process and understand, that we can represent it internally and create associations between concepts.

Minsky and Papert [1972] challenged the conventional view according to which information is transformed through a sequence of stages, from the external world stimuli to sensation, then perception followed by recognition, cognition and so on. Instead of looking at a mechanism as a hierarchy of parts and processes, they argued for a “heterarchy of computational ingredients”, *i.e.* a form of organisation in which parts can depend on each other, in this case, sensations, representations and perceptions.

Considering this duality between the representation and the processing of information, this section aims to discuss the differences between humans and machines when it comes to understanding representations. The notion of understanding can be related to the ability to infer the meaning of something such as an object, a situation, an image, a sound or someone’s behaviour.

1.2.1 How do humans perceive and interpret information?

As previously discussed, sensory stimulation is raw data that enters the human body through sense organs. Some of this information we are aware of, while most of it we do not pay attention to. *Perceptions* are the sensations we consciously acknowledge, and it is the brain that organises and interprets the sensory information [Gross, 2015]. The visual experience, our ability to see and recognise objects, places, faces and so on is attributed to the brain having and developing certain activation patterns which are associated with input received from the retina. The eye acts like a camera, in terms of capturing and sending to the brain the information that matters via ganglion cells, but it is in certain areas of the brain that the conscious visual experience happens.

Empiricism and nativism were two contradicting psychological theories on human behaviour and abilities, including visual perception, which emerged in the 17th-century [Samet and Zaitchik, 2017]. On the one hand, nativism, represented by Descartes [1641], supported the idea that knowledge and certain abilities are innate to humans and require little to no learning. On the other hand, empiricism, whose key representant was Locke [1690], believed that the human mind is blank at birth and gets filled in through sensory experience and learning. These were the extremes of the nature-nurture debate, but in between these, there were many other theories. Nowadays, it is mostly agreed that human traits are a product of both nature and nurture, each with different contributions [Gross, 2015].

Another way in which theories of visual perception differ is whether this is a direct (also known as bottom-up, data-driven) or indirect (top-down, conceptually-driven) process. The bottom-up theorists argue that perception is determined directly by data reaching sensory receptors, while the top-down theorists believe that perception is a result of the inferences made about the world based on prior knowledge and experience. All conceptually-driven theorists are also empiricists, while the data-driven theorists are split between nativists and empiricists.

Most of the principles of visual perception were first identified by the Gestalt school of psychology in the early 1900s and its theory emphasised that the form or shape quality of the whole cannot be identified from examination of its individual parts [Ehrenfels, 1890]. According to this theory, humans perceive whole objects instead of individual parts, and the sensory information is organised by innate principles, making the school part of the nativism movement. One of these principles is that of *form perception* which distinguishes an object from the background. Likewise, Gestalt theorists believed in perceiving "organised wholes" or patterns of stimuli instead of unique sensations, based on laws such as proximity, similarity, continuity, closure and a part-whole relationship. For example, visual elements situated in close proximity to each other are more likely to be perceived together, similarly to elements of the same type, or which constitute a continuous pattern such as a sinusoidal wave. However, these laws were criticised by more

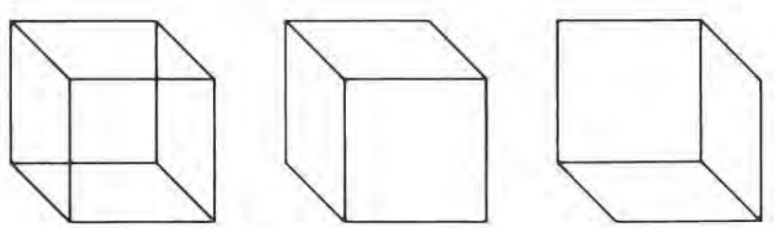


Figure 1.6: **The Necker Cube optical illusion introduced by Necker [1832].** The left most picture shows a cube followed by two potential interpretations of it. Image sourced from Marr [1982].

recent researchers as being only descriptive and difficult to apply to 3D objects [Eysenck, 1993] or whole scenes [Humphreys and Riddoch, 1987].

Depth perception is another process required in understanding visual scenes which allows us to construct 3D perceptual models from 2D retinal images. It is achieved by combining monocular cues from each eye separately. The two images received from each retina are combined by the brain in a process called stereopsis [Howard et al., 1995]. Another ability that our visual experience depends on is perceptual consistency, *i.e.* to perceive an object as unchanged despite changes in size, shape, location and colour of the sensory input [Gross, 2015].

From the top-down approaches, Gregory [1966]'s constructivist theory argues that our perception of the world is supplied by indirect inferences based on our previous knowledge, experience and expectations. He uses illusions as an example to show that our perception of what we see is actually our best guess based on how we normally interpret the world [Gregory, 1966, 1970]. When we look at the Necker cube illustrated in Figure 1.6, it happens that we switch between perspectives and that is our brain switching between potential hypotheses of the world.

On the other side of the spectrum, Gibson and Carmichael [1966] believe in the role of learning to develop visual perception. In their view, the environment provides us with all the necessary information through the optic array, which is defined as the pattern of light extended over time and spaces that reaches the retina. In particular, Gibson and Carmichael emphasises the importance of learning to differentiate between features in the optic array. Three main forms of information are contained in the optic array: optic flow patterns, texture gradients and affordances.

Neisser [1976], however, argue that both top-down and bottom-up approaches are required in the interactive, cyclic process of visual perception. As Eysenck and Keane [1995] discuss, a direct approach seems ideal in optimal viewing conditions, while the indirect approach becomes increasingly important in sub-optimal conditions, such as that of illusions occurring in the natural world.

Finally, Marr's computational theory builds upon empiricism but combines elements from top-down and bottom-up approaches. Marr [1982]'s theory is that a complex system, like the brain or a computer, should be understood at three different levels: 1) the device on which processes are to be realised physically (*e.g.* hardware), 2) what that device does and why (*e.g.* the process or the computation) and 3) the "how", the algorithm by which transformations are possible and the choice of appropriate representations for the input and output of the process [Marr, 1982]. As discussed in Section 1.1.1, according to Marr, the internal representation of an object starts fairly simple and schematic and becomes increasingly complex until it reaches object recognition. This transition, however, happens extremely fast. Marr et al. [1979] argue that the transformation from an analogous continuous representation (light intensity) to discrete symbolic representation (objects, symbols) happens almost immediately without loss of information.

What do humans perceive as 'visual semantics'? To end this section, we aim to establish what visual semantics refer to in the context of this thesis. When presented with an image of the real world, humans are capable of answering questions about any objects or beings, and about the relationships between them [Biederman, 2017]. In this thesis, we focus on the first question, the *what?*, *i.e.* the object category (or the list of categories). Research on the way humans perceive real-world scenes such as Biederman [1972] talks about the importance of meaningful and coherent context in the perceptual recognition of objects. Their study compares the accuracy of identifying a single object in a real-world jumbled scene versus in a coherent scene. On the other hand, theories such as that by Henderson and Hollingworth [1999] support the idea that object identification is independent of global scene context. A slightly more recent psychophysical study by Fei-Fei et al. [2007] shows that humans, in a single glance of a natural image, are capable of recognising and categorising individual objects in the scene and distinguishing between environments, whilst also perceiving more complex features such as activities performed or social interactions.

Despite the debate between the various theories and approaches to visual perception, it is clear that the notion of *objectness* is important in how a scene is understood by a human. Throughout this work we consider an object-based description of natural images (aligned with what humans would consider to be objects or object categories) to be suitable for the measurement of semantics captured by an emergent communication protocol.

1.2.2 How do machines interpret representations?

Because much of the human "knowledge" needed for the visual experience, be it scene or object perception and recognition, is subjective and it is sometimes impossible to convert it into a machine-readable format, it becomes essential that 'intelligent' machines automatically acquire the same knowledge in order to perform with human-comparable accuracy. In the machine learning and, especially, deep learning settings, feature vectors, weight matrices and scalar non-linear activation functions are used to perform the type of fast 'intuitive' inference that human common-sense reasoning depends on [Goodfellow et al., 2016].

As discussed in Section 1.1, differentiable neural models of representation are one of the most important advances made in the last century. Artificial Deep Neural Networks (DNNs) have been inspired by the structure and function of the brain and can: 1) produce useful representations of continuous and unstructured input, such as pixel values of an image, and 2) make inferences about the input based on these representations to solve various tasks.

This 'deep' approach to artificial intelligence allows a machine to learn complicated concepts by building them out of simpler ones [Goodfellow et al., 2016]. Each layer of the DNN extracts certain features which are then passed forward and used for specific tasks by neurons in higher layers. For image classification, for example, lower layers in the network are used to extract simpler features, such as edges, which are then forwarded to higher-level layers which detect more complex features, such as the global shape of objects [Lee et al., 2009]. DNN models are mostly trained using the backpropagation algorithm, which computes the gradient of a loss function with respect to the weights of the network for an input–output example [Rumelhart et al., 1986].

For any type of task in computer vision, information is first processed or encoded in a representation format that contains the most relevant parts for the task at hand. The stage that follows the encoding can be looked at as decoding into the desired output format. Depending on the task, the output can be an object class prediction as it is in the case of image classification, an annotation in the form of bounding boxes around different objects present in an image or labelling each pixel corresponding to an object, or a transcription of words associated with the objects in the image and so on.

The variety of deep learning models is incredibly rich and the field advances at a very fast pace, hence a comprehensive discussion is out of the scope of this thesis. However, from the most recent state-of-the-art models, one is particularly relevant to this thesis. CLIP [Radford et al., 2021] is a dual language-image encoder which allows machines to develop a better understanding of the connection between language and visual representations. The model is trained on large amounts of (image, text) pairs collected from the internet and learns to map representations of text and image in the same latent space, by predicting which caption is suitable for each image. Using the same representational space enables the comparison of similarity between these two information-bearing modalities. Therefore, CLIP provides two pretrained encoder models, one for language and one for vision, that can be integrated in other frameworks as we show in Chapter 5. CLIP's features have been tested in various tasks such as image and text retrieval, OCR, action recognition in videos and geolocalisation [Radford et al., 2021].

Along the same lines, DALL-E [Ramesh et al., 2021] and its improved version, DALL-E 2 [Ramesh et al., 2022], also make use of text and image representations to learn to generate realistic images from natural language descriptions. The most recent version, DALL-E 2 actually builds upon CLIP embeddings. The CLIP text embedding for a given caption is first passed to a prior model that produces a CLIP image embedding which is then decoded into the final image [Ramesh et al., 2022]. Concurrent with this approach, GLIDE [Nichol et al., 2021] also explores the diffusion decoder technique [Sohl-Dickstein et al., 2015] combined with CLIP or classifier-free guidance, to synthesis text-conditioned images.

1.3 Communicating Information

Having covered the representation and processing or understanding of information, the current section is looking at the ability to communicate it. But first, what is communication? There is not just one correct answer to the question. Fiske [2010] reflects on two main schools that study communication: the first one which considers it an exchange of messages, the *process* of encoding and decoding information such as in Shannon and Weaver [1949]'s model; and a second one which views communication as the production and exchange of *meanings* (see philosopher and logician Peirce [1931]). These are extensive studies which involve somewhat philosophical questions, but in essence, they reinterpret the definition of communication as the interaction through messages. Throughout this research programme, communication will be referred to as the process which emerges when two or more participants are involved and share a goal, task or incentive which can be achieved only by the transfer of information and so, is beneficial for all parties involved.

Communication is made possible by a common language or code, for example, English is a code that associates sounds with meanings and vice-versa. Studies on language origins [Nowak and Krakauer, 1999; Steels, 1997] consider cooperation to be a key prerequisite to language evolution as it implies multiple agents having to self-organise and adapt to the same convention. As Sperber [1995] discusses in his study on human communication, man has evolved an ability to communicate much more than simply encoding meaning into sound and decoding sound into meaning. Other species have their own codes and signals but are much more rudimentary in the sense that they convey essential things related to self-preservation, survival and territory ownership [Hauser, 1996]. Sperber [1995] goes on and argues that human communication, linguistic or non-linguistic, heavily relies on the ability of one's audience *to infer* the meaning from their own knowledge and understanding of a message or situation. This idea of human communication being essentially inferential simply translates to our ability to represent in our minds the mental representations of other beings [Sperber, 1995]. The concept of internal representation will be extremely relevant all throughout the programme of research in this thesis.

1.3.1 How do humans communicate?

Any exchange of information between living creatures can be considered communication and it is being displayed in all species, from primitive organisms like bacteria which employ cellular communication to plants, fungi, insects and animals. The human species, however, distinguishes itself by displaying a wide variety of means of expression and communication. Human communication is intriguingly complex, one of its most unique features being the use of abstract language and signs. Likewise, skills often encountered in today's society such as irony and sarcasm oppositely change the meaning of a sentence in the blink of an eye [Haiman et al., 1998]. Messages in real-life communication happen via various modes that can be very quickly combined to expand and/or change the overall meaning.

Humans can express their ideas, thoughts and feelings via visual, auditory and tactile channels. For example, individuals can express enthusiasm and approval by mimicry such as smiling (visual), clapping and cheering (auditory) or hand clasping (tactile). Gestures and actions can be considered as indicative as speech [McNeill, 2008; Kendon, 2004], although both these modalities are fleeting and retain no evidence of the message communicated. On the contrary, written codes, symbols or graphical depictions on various mediums like stone, wood, parchment and paper are permanent and remain as testimony of the evolution of communication. For the purpose of this thesis, the discussion will focus on the latter form of communication, that which is permanent, that is either written or marked with an instrument on a durable surface.

To reiterate, communication is only possible if there exists a common understanding, when participants share an agreed-upon language, be it spoken, written, or determined by gestures, facial expressions or actions. For any type of message to hold any significance at all, participants in the communication act must have the ability to decode and interpret the language. So, a natural question that arises is *how has language emerged in the first place?*

1.3.1.1 The evolution of human communication

Language arose as a link between the world and the people living in it, as a means of transmitting information, knowledge and emotions. The question of how humans developed language is a very controversial one. There exist countless theories on the topic of language evolution. As Fitch [2007] highlights, language evolution studied in the eighteenth and nineteenth centuries was mainly correlated with biological evolution and individual learning abilities. More recently, aspects such as cultural processes have been included in theoretical accounts of human communication systems [Kirby and Hurford, 1997; Kirby et al., 2007; Christiansen and Kirby, 2003]. From all these accounts, it is worth recounting studies on animal communication systems [Hauser, 1996], developmental

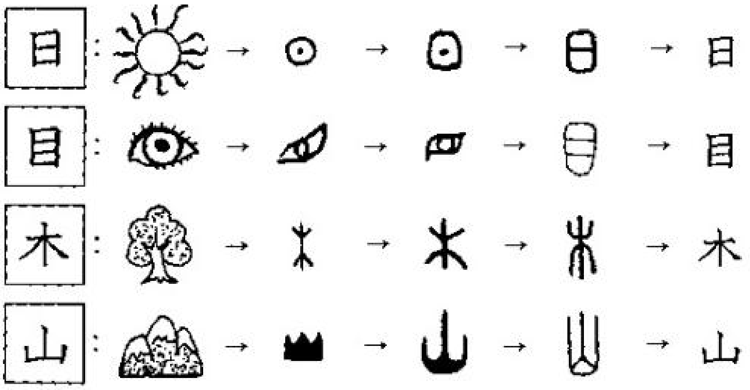


FIGURE 1.7: **Example of Chinese characters' evolution.** Image sourced from `https://mlc.ua.edu/chinese/minor/conversation-cafe/`.

intelligence and language in primates and early ancestors of humans and how their form of intelligence compares to the different stages of brain and cognitive functions in children [Parker and Gibson, 1979; Cheney and Seyfarth, 1990; Bickerton, 1990]. Some argued that language appeared as a side-effect of the development of the brain for other purposes. For example, Parker and Gibson [1979] attributed language emergence to the development of more complex behaviours in hominids such as hunting, tool-making, shelter construction and so on. On a different note, there are studies from the point of view of language development in children [Hurford, 1991; Bates, 1992; Smith and Gasser, 2005; Buhler, 2013]. Some of these studies support the idea that linguistic capacities are genetic, innate to human beings [Pinker, 2003]. However, this does not negate the existence of audible communication in the animal kingdom such as among dolphins or whales [Mann et al., 2000]. One can observe babies babbling and imitating sounds they hear. Likewise, crying is an instinct children are born with to attract attention, procure food and care for their needs. There exist lots of other approaches which explore mathematical and computational modelling [Nowak and Krakauer, 1999], symbolic thinking [Deacon, 1997] and much more.

Nevertheless, languages have been shown to evolve rapidly [Fitch, 2007; Lieberman et al., 2007; Pagel et al., 2007]. William [1994] covers the principles according to which and the levels at which linguistic change occurs, including vocabulary, syntax and phonology.

The transformation and growth of human civilisation are attributed to several activities among which written communication [Cherry, 1953]. This, in particular, can be seen as a revolution as it allowed humanity to share and pass on information without actually requiring physical presence. Ideas transmitted through speech deteriorate over time and carry inaccuracies. Hence, writing can be considered "a threshold of history" as Schmandt-Besserat [1992] describes it. Funnily enough, until the eighteenth century, the origins of writing were to be found in myths from different cultures [Schmandt-Besserat, 1992]. According to these, gods, deities or lords gifted mortals writing as a full-fledged

Figure 1.8: **Panel of Horses - Charcoal drawing on rock discovered at the Chauvet Cave in France.** Image sourced from Clottes [2008].

apparatus. All those myths, however, completely ignore evidence that alphabets evolved from a simple system to the complex state that we now know them as.

With the Enlightenment age, those beliefs changed as the first evolutionary theory of writing was proposed by Warburton [1742]. In essence, this theory argued that all scripts started with a pictographic stage and, over time, were simplified and abstracted. Mexican (Aztec), Egyptian and Chinese alphabets were used in this study to illustrate the stages of refinement and simplification (see Figure 1.7). We have numerous decipherments of ancient scripts, the Rosetta stone being key to decoding ancient Egyptian [Robinson, 2002]. Early writings of Mediterranean civilisation were logographic. The most well-known example is the Phaistos disc discovered in Crete at the beginning of the twentieth century and believed to be the first printed document at around 1600 BC [Robinson, 2002]. Cave paintings such as those found at Lascaux, Chauvet (*e.g.* Figure 1.8) and Coliboaia and stone engravings artefacts discovered all over the Mediterranean space stand as evidence for the pictographic theory [Clottes, 2008].

1.3.2 How can communication be achieved between AI agents?

Communication can be looked at as the process in which one agent, human or AI, encodes information into an embedding, that can be a continuous or discrete message, and a second agent decodes it. Considering this view of communication, there have been numerous attempts at modelling and studying the emergence of communication protocol among artificial intelligent agents.

1.3.2.1 Emergent communication

The emergence of language in multi-agent settings has traditionally been studied in the language evolution literature which is concerned with the evolution of communication protocols from scratch [Steels, 1997; Nowak and Krakauer, 1999]. These early works survey mathematical models and software simulations with artificial agents to explore how various aspects of language have begun and continue to evolve. One key finding of Nowak and Krakauer [1999] is that signal-object associations are only possible when the information transfer is beneficial for both parties involved, and hence that *cooperation* is a vital prerequisite for language evolution. The research presented in this thesis is inspired by a renewed interest in the field of emergent communication. Using contemporary deep learning methods, artificial agents, usually modelled as deep neural networks, learn to cooperate on a variety of tasks and develop a communication strategy in doing so. One of the most encountered tasks and also the one explored in this thesis is that of an image referential game illustrated in Figure 1.9 [Sukhbaatar et al., 2016; Evtimova et al., 2017; Havrylov and Titov, 2017; Lazaridou et al., 2017; Lee et al., 2017; Mordatch and Abbeel, 2017; Lazaridou et al., 2018; Cao et al., 2018; Chaabouni et al., 2019; Li and Bowling, 2019]. In Section 1.3.3, we discuss the importance of gameplay for learning and particularly for inducing human-like language. Other examples of cooperative tasks which require communication between multiple agents include: language translation [Lee et al., 2017], logic riddles [Foerster et al., 2016], simple dialog [Das et al., 2017] and negotiation [Cao et al., 2018; Lewis et al., 2017]. All these studies build toward the long-standing goal of having specialised agents that can interact with each other and with humans to cooperatively solve tasks and hence assist them in daily life such as going through different chores.

Alongside this goal, emergent communication research aims for the emerged *protolanguage* to receive no, or as little as possible, human supervision. However, reaching coordination between agents solving a cooperative task, while developing a human-friendly communication protocol has been shown to be extremely difficult [Lowe et al., 2019; Chaabouni et al., 2019; Kottur et al., 2017]. In these settings, the emergent language has no prior meaning, neither semantics nor syntax, and the aim is to develop these by learning to solve the task through many trials or attempts. However, it is not clear that it actually achieves these [Bouchacourt and Baroni, 2018; Kottur et al., 2017; Lowe et al., 2019]. Lee et al. [2019] proposes a translation task (*i.e.* encoding a source language sequence and decoding it into a target language) via a third pivot language. They show that auxiliary constraints on this pivot language help to best retain original syntax and semantics. Other approaches [Havrylov and Titov, 2017; Lazaridou et al., 2017; Lee et al., 2017] directly force the agents to imitate natural language by using pretrained visual feature vectors, which already encode information about objects. Lowe et al. [2020], on the other hand, discusses the benefits of combining expert knowledge supervision and self-play, with the end goal of making human-in-the-loop language learning algorithms more efficient.

Protolanguage and properties. Baroni [2020] highlights some of the priorities in current emergent language research and sketches the characteristics of a useful *protolanguage* for deep agents. It draws on the idea from linguistics that human language has gone through several stages before reaching the full-blown form it has today, and it had to start from a limited set of simple constructions [Bickerton, 2014]. By providing a realistic scenario of daily interaction between humans and deep agents, Baroni [2020] emphasises that a *useful* protolanguage first needs to use words to categorise perceptual input; then allow the creation of new words as new concepts are encountered, and only after, deal with predication structures (*i.e.* constructions of two components) which combine words referring to objects and words denoting properties or actions. The focus of this thesis is on the categorisation phase as we explore whether deep agents can develop a language which captures visual concepts whilst simultaneously learning features from natural images in a completely self-supervised way.

1.3.2.2 Continuous versus discrete communication

When communication is considered, a distinction can be made between discrete or continuous modes. As indicated by archaeological discoveries, the first written scripts have been shown to follow a photographic representation of the world. There are no two identical representations of a flower as the process of drawing is intrinsically continuous in time and so is the artefact resulting from it. However, the meaning attached to it is discrete. Therefore, sketches and drawings can be looked at both as a raster image with a discrete meaning, but can also be considered as sequences of strokes continuous in time.

With the transition and refinement of visual representations, the first hieroglyphics emerged. This represents already a transition to a token-based, or discrete, mode of written communication. In the context of this thesis, "token" refers to any symbol from a discrete vocabulary. Depending on the level of granularity, a token can be a letter in the alphabet or a word in the English vocabulary.

In the context of emergent communication studies, Lazaridou and Baroni [2020] highlight the distinction between two types of communication that can be established between multiple agents, depending on the nature of the communication bottleneck (*i.e.* of the message vector): it can be *continuous* where the agents transmit continuous vectors [Sukhbaatar et al., 2016; Foerster et al., 2016; Singh et al., 2019], or *discrete*, in which case, the agents 'talk' using a sequence of symbols [Havrylov and Titov, 2017; Lazaridou et al., 2017, 2018].

Foerster et al. [2016] proposed two approaches in the context of emergent communication protocols for both continuous and discrete communication. Their differentiable inter-agent learning (DIAL) has been one of the most influential systems. DIAL allows agents to communicate through a continuous channel, making it possible to back-propagate gradients through the whole system. Hence, the continuous vector which connects the two agent networks turns the multi-agent system into a single large network. Similarly, Sukhbaatar et al. [2016] showed that agents communicating through a continuous channel are easier to train than in a discrete case. Nevertheless, their model assumes full cooperation between agents and hence restricts its application in mixed or competitive scenarios. Singh et al. [2019] tackles this issue with a gating mechanism and individualised rewards for each agent which results in better training efficiency than simple continuous communication models and the possibility of use in various setups. For discrete communication, Foerster et al. [2016] also proposed reinforced inter-agent learning (RIAL), a model commonly used in the emergent communication literature, in which communication happens through discrete symbols. This approach treats agents as independent networks, in which each agent is conditioned on its individual hidden state and observations, as well as messages received from other agents, but does not have access to their internal states (much like in human communities). The task reward is the only learning signal received by each agent. A potential disadvantage of the RIAL system could be that the agents cannot give each other explicit feedback about their actions (*e.g.* confirming understanding in communication games). The feedback is only implicit in the reward.

This limitation has been addressed by the DIAL model, but also by methods such as REINFORCE [Williams, 1992; Lazaridou et al., 2017] or those that approximate discrete representations by continuous ones during training [Maddison et al., 2017; Jang et al., 2017]. Concretely, Maddison et al. [2017] and Jang et al. [2017] introduced what we now know as the Gumbel Softmax estimator, which is a reparameterisation that allows one to sample a categorical distribution ($t \sim \mathrm{Cat}(p_1, \ldots, p_K)\,;\, \sum_i p_i = 1$) from its logits $\boldsymbol{x}$. The first step in understanding this estimator is the Gumbel-max trick:

$$t = \operatorname*{argmax}_{i \in \{1, \cdots, K\}} x_i + z_i \tag{1.3}$$

where $z_1, \ldots z_K$ are independent and identically distributed Gumbel(0,1) variates which can be computed from Uniform variates through $-\log(-\log(\mathcal{U}(0,1)))$. Clearly argmax is not differentiable, but it can be replaced with a continuous approximation using the softargmax:

$$\mathrm{softargmax}(\boldsymbol{y}) = \sum_i \frac{e^{y_i/T}}{\sum_j e^{y_j/T}} i \tag{1.4}$$

where T is the temperature parameter. This relaxation gives us the Gumbel-softmax, a continuous approximation to sampling a categorical distribution. One way to utilise this is to use the Gumbel-softmax approximation during training, and replace it with the hard max at test time, however, this can often lead to problems because the model can learn to exploit information leaked through the continuous variables during training.

$$\mathrm{STargmax}(\boldsymbol{y}) = \mathrm{softargmax}(\boldsymbol{y}) + \mathrm{stopgradient}(\mathrm{argmax}(\boldsymbol{y}) - \mathrm{softargmax}(\boldsymbol{y})) \tag{1.5}$$

where stopgradient is defined such that $\mathrm{stopgradient}(\boldsymbol{a}) = \boldsymbol{a}$ and $\nabla\,\mathrm{stopgradient}(\boldsymbol{a}) = 0$.

Combining the Gumbel-softmax trick with the STargmax results in the Straight-through Gumbel Softmax (ST-GS) which gives discrete samples and with a usable gradient. The straight-through operator is biased but low variance; in practice, it works very well and is better than the high-variance unbiased estimates you could get through REINFORCE [Havrylov and Titov, 2017]. In short, this trick allows us to train neural network models that incorporate fully discrete sampling operations using gradient-based methods in a fully end-to-end fashion (see Chapter 2).

Besides a language-based communication protocol, with either discrete or continuous communication channels [Havrylov and Titov, 2017; Lazaridou et al., 2017, 2018; Bouchacourt and Baroni, 2018], the attention recently has shifted towards drawing or sketching as a more flexible means of expression and transmission of information among artificial agents [Fernando et al., 2020; Fan et al., 2020; Qiu et al., 2021].

The research programme in this thesis explores both language-based and visual communication. In Chapter 2, the protolanguage developed between artificial agents is formed of variable-length sequences of discrete tokens, which are chosen from a predefined, fixed vocabulary. The learned protocol is not grounded in any way, such that the messages are not forced to be similar to those of natural language. As described in Section 1.2.1, we believe it is a reasonable assumption that if the game were to be played by human agents they would capture the object's category and its properties that help distinguish the target from the distractor images. Later, in Chapter 4 an interpretable continuous form of communication is explored, that of drawing and sketching, which aims to address some of the weaknesses of discrete, token-based communication such as the lack of interpretability.

1.3.3 Multiplayer gameplay for learning

Before concluding this chapter, the concept of gameplay needs to be considered as it has been shown to provide an effective environment for learning and improving skills such as communication, memory and analytical thinking [Golinkoff and Hirsh-Pasek, 2006]. In the context of this thesis, gameplay represents the main setup for training artificial agents to communicate in a fully self-supervised way.

1.3.3.1 In humans

The importance of play and its connection to learning has long been recognised in biological and evolutionary literature [Huizinga, 2014; Caillois, 2001; Csikszentmihalyi and Bennett, 1971; Csikszentmihalyi, 2014; Prensky, 2001; Golinkoff and Hirsh-Pasek, 2006]. Diane Ackerman states in her book Deep Play [Ackerman, 2011] that "play is our brain's favourite way of learning things". Studies on animal play have led psychologist Robert Fagan to speculate that play in a relaxed environment represents the "optimal generic learning" experience [Angier, 1992; Prensky, 2001]. Other researchers like Piaget consider playing the "work of childhood" [Piaget, 2013] and its role essential in the development of academic skills [Hirsh-Pasek et al., 2003]. Play facilitates the assimilation of sensory-motor associations and cognitive representations as it involves exploration and interaction with the surrounding world, as well as learning by imitating adult behaviours and motor skills [Golinkoff and Hirsh-Pasek, 2006].

It has been shown that games can be seen as necessary tools for learning and behavioural changes [Connolly et al., 2012]. Boghian et al. [2019] looks into the importance of play and game-based learning methods in human education and development. Three types of educational games are identified: 1) board games which usually involve strategic planning and/or dice-rolling, 2) card games and 3) videos games. Their study highlights the key aspects of play that have an impact on the life of adults too, apart from mere entertainment and recreational purposes.

The advantages associated with game-based learning in adult education include enhancement of cognitive abilities such as creative, analytic and reflective thinking. Organisational abilities are improved as well as most board games usually involve solving a problem for which the planning and organisation of resources are necessary. Likewise, game playing can improve self-related abilities like self-management, self-motivation and confidence, ability to concentrate for a long time on a goal, ability to critically reflect on one's progress and flexibility. Finally, games and especially board games have an impact on adults' ability to work well in collaboration, improve social skills and communication, and lead to the development of interpersonal skills like empathy, diplomacy, negotiation and conflict management [Boghian et al., 2019]. Cultural awareness and expression can also be tackled and improved through gameplay [Roberts et al., 1959].

Most relevant to our work are games which involve cooperation and coordination of participants that results in learning and developing a certain type of shared understanding and communication protocol. Amongst such games played by humans one can think of Pictionary, "Guess-Who?" and Codenames.

Pictionary is a cooperative game in which participants split up into teams, and one member from each team takes a turn at a time. The one who takes the turn, the drawer, tries to describe to their team a word printed on a card only by sketching on a piece of paper. The team can guess the word as the drawer produces the sketch. That in turn can lead the drawer to alter the sketch so as to lead the team to make the correct guess. No writing and no verbal communication are allowed between the drawer and the team during the round.

Neuroimaging research with participants playing a Pictionary-like game showed that more brain areas are involved and interacting when performing creative activities than during usual activities [Saggar et al., 2015], and thus facilitate spontaneous improvisation and visual creativity. Pictionary has also been explored for language acquisition purposes and it has been shown to increase students' creativity, interest and participation [Hamer and Lely, 2019; Fadirsair et al., 2021]. These findings indicate that the value of the game goes beyond mere entertainment, as it involves varying levels of cognition and abstraction [Dake and Roberts, 1995], as well as social interaction and collaboration [Mäyrä, 2007].

1.3.3.2 In artificial agents

There is a long history of training and testing artificial intelligence through gameplay. Lewis's classic signalling games [Lewis, 1969] have been extensively studied for language evolution purposes [Steels, 1997; Nowak and Krakauer, 1999], but also in game theory under the name of 'cheap talk' games. These games are coordination problems in which agents must choose one of several alternative actions, but in which, their decisions are influenced by their expectations of other agents' actions. A different category of games have been studied are those that involve adversarial strategy such as chess [Silver et al., 2018], Go [Silver et al., 2016], Atari [Mnih et al., 2013], poker [Moravčík et al., 2017] or StarCraft [Vinyals et al., 2017]. Cooperative games that require a shared understanding and communication between players in order to achieve a shared goal have also been explored. Such examples include Codenames [Kim et al., 2019], 'Guess Who?' [Jorge et al., 2016] and Hanabi [Walton-Rivers et al., 2019]. These games, however, imply certain types of communication, some using single words or actions as communication means.

In emergent communication research, image-based reference games have been widely used. Similar to Lewis's games, these games are coordination problems between multiple agents that require a *limited* communication channel through which information can be exchanged to solve a cooperative task. The task usually requires one agent to transmit information about an image through a discrete or continuous bottleneck, and a second agent to guess the correct image from several others based on the received message [Lazaridou et al., 2017, 2018; Havrylov and Titov, 2017]. Figure 1.9 illustrates an example of such a referential communication game. In this example, information is communicated through language which is nothing more than a discrete communication channel. One potential constraint of a language bottleneck is that it needs to be understood by both agents. For example, a Romanian-speaking participant who does not speak English might not understand the communicated concept of "brown horse". However, the sketch of a horse is likely to be understood by any language-speaking participant.

Games that explore drawing and sketching as a richer communication modality have gained attention in recent years. Pictionary-style guessing games such as Sketch-QA [Sarvadevabhatla et al., 2018] or Stellasketch [Johnson and Do, 2009], have been proposed

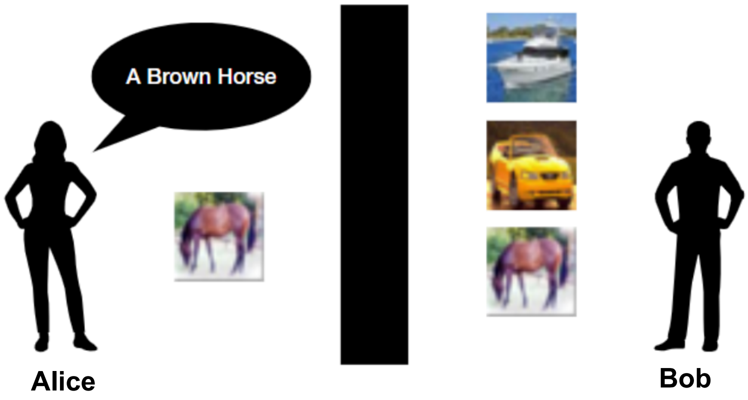


Figure 1.9: **An Image Referential Game** (see Lewis [1969]'s coordination games): Alice must communicate to Bob the image she has. Bob has that image + many distractors. Alice knows nothing about the distractors Bob has (they could all be white boats!).

for generating datasets of guess words for hand-drawn sketches produced by humans. These datasets can aid in training sketch recognition models to guess in a human-like way. More recently, Clark et al. [2021] proposed Iconary, a Pictionary-based collaborative game in which the drawer communicates an image of a phrase by combining, resizing and rotating icons on a canvas. Similar to Pictionary, the guesser in Iconary makes a series of guesses for the phrase and if it is unsuccessful, the drawer can revise the canvas by updating, removing, or adding icons so as to help the guesser. Then the cycle repeats until the guesser is correct or until time runs out. Chapter 4 of this thesis explores sketching as a communication bottleneck between artificial agents that learn to play image referential games.

1.4 Summary

This chapter has provided an overview of the main components of this research programme: perception of visual scenes, learning meaningful representations from visual information and how to communicate them efficiently in the setting of visual reference games. The goal of the research in this thesis is to contribute toward improving human-agent communication by leveraging knowledge about how humans learn to see and communicate in real life. The following chapters explore two forms of written communication, with tokens and through drawing, and extend the discussion of some topics covered in this chapter.

Chapter 2

Communication with Tokens

"Each character is full of life"

— Shinagawa Tetsuzan

"The single biggest problem in communication is the illusion that it has taken place."

— George Bernard Shaw

Emergent language research aims to develop agents that can cooperate with each other, and ultimately with humans. To achieve this goal, these agents necessarily communicate with particular protocols through communication channels. In emergent-communication research, communication protocols are learned by the agents, and researchers often investigate how these protocols compare to natural human languages (see Section 1.3.2). This chapter studies the emergence of semantics in token-based communication protocols learned by playing visual referential signalling games [Lewis, 1969], also discussed in Section 1.3.3.

Previous research has looked into how pre-linguistic conditions, such as the input representation (either symbolic or raw pixel input), affect the nature of the communication protocol [Lazaridou et al., 2018]. In this chapter, we explore and highlight the features of a referential game that can improve the *semantics* of the conveyed messages towards a more naturally interpretable form. More specifically we focus on how to bias agents away from learning image-hashing solutions that can naïvely solve the game perfectly without actually capturing any notion of the *meaning* of the input images. We then explore the effects of linking language learning with feature learning in a completely self-supervised setting where no information on the objects present in a scene is provided to the model at any point. We thus seek to build a bridge between recent research in self-supervised feature learning with recent advances in self-supervised gameplay through emergent communication channels.

2.1 Motivation and Contributions

The idea that agents might learn language by playing visually grounded games has a long history [Cangelosi and Parisi, 2002; Steels, 2012]. Research in this space has recently had something of a resurgence with the introduction of a number of models that simulate the play of *signalling* games [Lewis, 1969] using realistic visual inputs [Lazaridou et al., 2017; Havrylov and Titov, 2017; Lee et al., 2017]. On one hand, these works have shown that the agents can learn to successfully communicate to play these games; however, on the other hand, there has been much discussion as to whether the agents are really learning a communication system grounded in what humans would consider being the semantics of visual scenes. Bouchacourt and Baroni [2018] highlight this issue in the context of a pair of games designed by Lazaridou et al. [2017] which involved the sender and receiver agents being presented with pairs of images. They show that the internal representations of the agents are perfectly aligned, which allows them to successfully play the game but does not enforce capturing conceptual properties. Moreover, when the same game is played with images made up of random noise, the agents still succeed at communicating, which suggests that they agree on and rely on incomprehensible low-level properties of the input which drift away from human-interpretable properties. This finding should perhaps not be so surprising; it is clear to see that one easy way for agents to successfully play these visual communication games would be by developing schemes which create hash-codes from the visual content at very low levels (perhaps even at the pixel level).

Havrylov and Titov [2017] explored a different, and potentially harder, game than that proposed by Lazaridou et al. [2017]. In their game (see Section 2.2 for full details), the sender sees the target image and the receiver sees a batch of images formed of a number of distractor images plus the target one. The sender agent is then allowed to send a variable-length message, up to a maximum length, from a fixed vocabulary to the receiver. The latter then needs to use that message to identify the target. As opposed to Lazaridou et al. [2017]'s game in which both agents see only a pair of images, this setting requires the message to include information that will allow the receiver to pick the target image from a batch of 128 images. In their work, they show some qualitative examples in which it does appear that the generated language does in some way convey the visual semantics of the scene (in terms of 'objectness' — correlations between the sequences of tokens of the learnt language and objects, as perceived by humans, known to exist within the images). There are however many open questions from this analysis; one of the key questions is to what extent the ImageNet-pretrained VGG16 CNN [Simonyan and Zisserman, 2015] used in the model is affecting the language protocol that emerges.

In this chapter, we explore visual semantics in the context of Havrylov and Titov [2017]'s referential game by carefully controlling the visual feature extractor that is used and augmenting the gameplay in different ways. We seek to explore what factors encourage the emergent token-based language to convey visual semantics rather than falling back to a communication system that just learns hashes of the input images. More concretely, we:

- Study the effect of different weights in the CNN used to generate the features: fixed (pretrained on ImageNet and frozen as in Havrylov and Titov [2017]), initialised randomly and frozen, and learned end-to-end in the model. We find that models with a feature extractor pretrained in a supervised way capture the most semantic content in the emergent protocol.
- Investigate the effect of augmentations that make the game harder by changing the image given to the sender (adding noise and/or random rotations), but not the receiver. Overall, adding noise seems to only make the game slightly harder as the communication success drops, while rotation improves the visual semantics metrics.
- Explore the effect of independently augmenting the images given to the sender and the receiver (random cropping and resizing to the original image size, random rotations and colour distortion), so they do not see the exact same image. We show that it is possible to get a fully learned model that captures similar amounts of semantic notions as a model with a pretrained feature extractor.
- Extend the game to include a secondary task (guessing the rotation of the sender's input) in order to assess whether having agents perform more diverse tasks might lead to stronger visual semantics emerging. We find that without a complex sequence of data augmentation transforms and any supervision, a more meaningful communication protocol can emerge between agents that solve multiple tasks.
- Analyse the effect of pretraining the feature extractor network in a self-supervised framework before engaging in the multi-task game. We show that solving such a self-supervised task helps ground the emergent protocol without any human supervision and is even more beneficial for the semantic content captured by a fully learned model.

We draw attention to the fact that other than in the cases where we use pretrained feature extractors, our simulations are completely self-supervised, and there is no explicit signal of what a human would understand as the 'visual semantics' (as discussed in Section 1.2.1) given to the models at any point. If our models are to communicate visual semantics through their communication protocols, then they must learn how to extract features that provide suitable information on those semantics from raw image pixel data.

Section 1.3.2 and Section 1.3.3 looked at related work, which necessarily covers a wide range of topics relevant for this chapter. The remainder of the chapter is structured as follows: Section 2.2 describes our baseline game and model, building upon the work of Havrylov and Titov which is reproduced for comparison in Section 2.3 and discusses different learning strategies and hyperparameters. Sections 2.4 to 2.7 present a range of investigations into the factors that can make the emergent communication protocol convey more semantically meaningful information. Finally, Section 2.9 summarises our findings and discusses possible research directions to be explored in the future based on these results. A detailed discussion of the future of emergent communication is presented in Chapter 6.

2.2 Baseline Experimental Setup

In this section we provide the details of our experimental setup; we start from Havrylov and Titov [2017]'s image reference game. The objective of the game is for the sender agent to communicate information about an image it has been given to allow the receiver agent to correctly pick the image from a set containing many (127 in all experiments) distractor images.

2.2.1 Model architecture

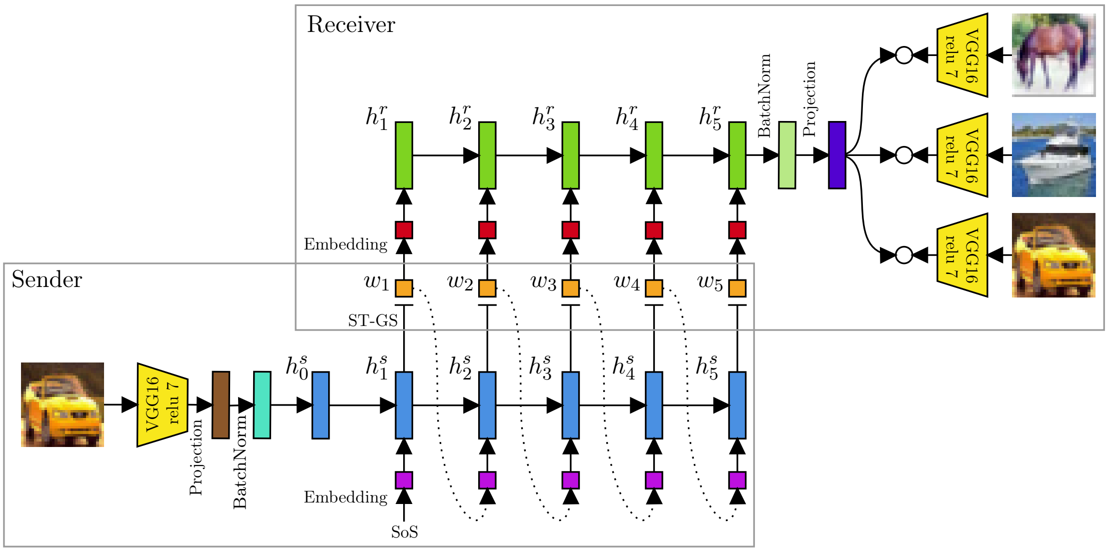


FIGURE 2.1: **Havrylov and Titov [2017]'s game setup and model architecture.**

Havrylov and Titov [2017]'s model and game are illustrated in Figure 2.1. The sender agent utilises an LSTM to generate a sequence of tokens given a hidden state initialised with visual information and a Start of Sequence (SoS) token. To ensure that a sequence of only discrete tokens is transmitted, the output token logits produced by the LSTM cell at each timestep are sampled with the Straight-Through Gumbel Softmax operator (ST-GS).[1] The ST-GS provides a one-hot vector at each time step but uses the gradients of the relaxed Gumbel Softmax during the backward pass to circumvent the problem

[1] Havrylov and Titov [2017] experimented with ST-GS, the relaxed Gumbel Softmax and REINFORCE in their work, however, we focus our attention on ST-GS here.

that the sampling is non-differentiable [Maddison et al., 2017; Jang et al., 2017]. The receiver agent uses an LSTM to decode the sequence of tokens produced by the sender, from which the output is projected into a space that allows the receiver's image vectors to be compared using a dot product. Havrylov and Titov [2017] use a fixed VGG16 CNN pretrained on ImageNet to extract image features in both agents. The model is trained using a hinge-loss objective (see Equation 4.3) to maximise the probability of the correct image being chosen. The sender can generate messages up to a given maximum length; shorter codes are generated by the use of an End of Sequence (EoS) token. Although not mentioned in the original paper, we found that the insertion of a BatchNorm layer in the sender between the CNN and LSTM, and after the LSTM in the receiver, was critical for learnability and reproduction of the original experimental results.

2.2.2 Training details

Our experiments use the model described above with some modifications under different experimental settings. In all cases, we perform experiments using the CIFAR-10 dataset rather than the COCO dataset used in the original work (to replicate the original results requires multiple GPUs due to the memory needed, as well as considerable training time[2]). In light of the smaller resolution images and lower diversity of class information, we choose a word embedding dimension of 64, a hidden state dimension of 128, and a total vocabulary size of 100 (including the EoS token). We also limit the maximum message length to 5 tokens. The training data is augmented using color jitter ($p_{bri} = 0.1, p_{con} = 0.1, p_{sat} = 0.1, p_{hue} = 0.1$), random grayscale transformation ($p = 0.1$), and random horizontal flipping ($p = 0.5$), so there is very low probability of the model seeing exactly the same image more than once during training. The batch size is set to 128, allowing for the receiver to see features from the target image plus 127 distractors. Most simulations converge or only slowly improve after about 60 epochs, however for consistency, all results are reported on models trained to 200 epochs where convergence was observed to be guaranteed for well-initialised models[3].

2.2.3 Metrics

Our key objective is to measure how much visual semantic information is being captured by the emergent language. If humans were to play this game, it is clear, as discussed in Section 1.2.1, that a sensible strategy would be to describe the target image by its

[2] We found that about 32GB of RAM spread across four RTX-2080Ti GPUs was required with the sender, receiver and feature extractor each being placed on a different GPU, and the loss being computed on the forth. Each epoch of 74624 games (for each batch of 128 images we played the 128 possible games by taking each image in turn as the target) took around 7 minutes to complete. The convergence of the communication rate to a steady level took at least 70 epochs.

[3] Certain model configurations were more sensitive to initialisation; this is discussed further in Section 2.4

semantic content (*e.g.* “a yellow car front-on” in the case of the example in Figure 2.1). It is also reasonable to assume in the absence of strong knowledge about the make-up of the dataset (for example, that the colour yellow is relatively rare) that a semantic description of the object in the image (a “car”) should have a strong part to play in the communicated message if visual semantics are captured. Work such as Hare et al. [2006] considers the semantic gap between object/class labels and the full semantics and significance of the image. However, in the case of the CIFAR-10 dataset in which most images have a single subject, “objectness” can be considered a reasonable measure of semantics.

With this in mind, we can measure to what extent the communicated messages capture the object by looking at how the target class places in the ranked list of images produced by the receiver. More specifically, in the top-5 ranked images guessed by the receiver, we can calculate the number of times the target object category appears, and across all the images we can compute the average of the ranks of the images with the matching category. In the former case, if the model captures more semantic information, the number will increase; in the latter, the mean-rank decreases if the model captures more semantic information. A model which is successful at communicating and performs almost ideal hashing would have an expected top-5 number of the target class approaching 1.0 and an expected average rank of 60 (resulting from $\frac{1\times1+11.8\times65}{12.8}$), whilst a model that completely captures the “objectness” (and still guesses the correct image) would have an expected top-5 target class count of 5 and expected mean rank of 6.9 (resulting from $\frac{1\times1+11.8\times7.4}{12.8}$). In addition to these metrics for measuring visual semantics, we also measure the top-1 and top-5 communication success rate (receiver guesses correctly in the top-1 and top-5 positions) and the message length for each trial. On average across all games, there are 12.8 images with the correct object category in each game (on the basis that the images are uniformly drawn without replacement from across the 10 classes and the correct image and its class are drawn from within this). If the message transmitted only contained information about the object class, then the communication success, when considering the top-1 and top-5 choices of the receiver, would be on average 0.078 (resulting from $\frac{1}{12.8}$), and 0.39 (resulting from $5 \times \frac{1}{12.8}$) respectively. Since we observe that throughout the experiments there is a significant trade-off between the semantics measures and the top-1 communication rate, we consider the top-5 rate a better indication of the capacity of the model to succeed at the task while learning notions of semantics. If the communication rate in top-5 is higher than the average, it means that the message must contain additional information about the correct image, beyond the type of object. However, we do not easily have the tools to find out what that extra information might be; it could be visual semantics such as attributes of the object, but it could also be some robust hashing scheme.

2.3 Re-implementing the Original Game: Exploring Datasets and Learning Stability

We first implemented Havrylov and Titov [2017]'s model described in Section 2.2.1 and attempted to replicate their results on Microsoft's COCO dataset consisting of 80 different object classes [Lin et al., 2014]. We follow their setup and use 90% of COCO 2014 training set for training, while the rest of 10% randomly selected images are used as validation data. The learned protocol is evaluated on COCO 2014 validation set. In this model configuration, the vocabulary size is 10000, embedding dimensionality is 256, LSTM layer dimensionality is 512 and the batch size is 128 (127 distracting images + target image). For further details, please refer to Havrylov and Titov [2017].

It is worth mentioning that following their exact details of the architecture and setup was not sufficient. It was only after we added a batch normalisation layer in the sender, between the feature extractor CNN (after the affine transformation of the features) and the LSTM, and in the receiver, after the last hidden layer of the LSTM (before the affine layer), that we reached a communication success similar to that claimed in their paper. Therefore, batch normalisation is a critical feature for the learning stability of this game.

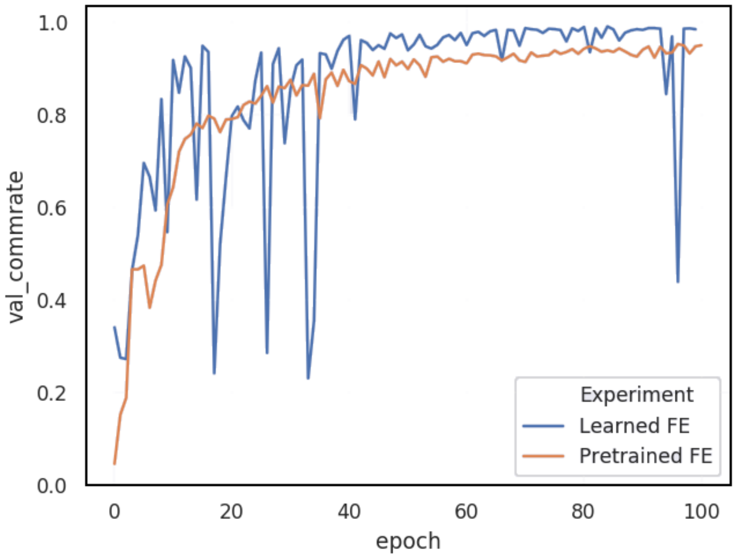


Figure 2.2: **Validation communication success rate for two variants of the original model [Havrylov and Titov, 2017]** trained with a maximum message length of 14: with a pretrained and a learned feature extractor.

Our results from the reproduction of the original experiments on COCO (with the aforementioned BatchNorm layer) are depicted in Figure 2.2. We replicated the model with a maximum message sequence length of 14. The model with a pretrained feature extractor reaches a communication success of 94% on the validation set, whereas the same model which also trains the VGG16 CNN during the gameplay reaches a communication success of 98%, but is more unstable. The models have been trained for 100 epochs which seem to be enough for good communication to be established.

As previously mentioned in Section 2.2.3, using COCO for the experiments is very expensive in terms of GPU memory and running time. The training set has around 74500 images, while the validation has 8000 images. More than 30 gigabytes of GPU memory are needed for training, so we had to split the model over 4 GPUs to train one model only. Moreover, training for 100 epochs and following the original model configuration with ST-GS takes over 10 hours to complete, which makes the reproducibility of experiments a very big challenge. Because of these constraints, further experiments will explore the CIFAR-10 dataset. For comparison, the same experiment ran for the same number of epochs, takes slightly over an hour and requires only one GPU.

2.3.1 Qualitative analysis

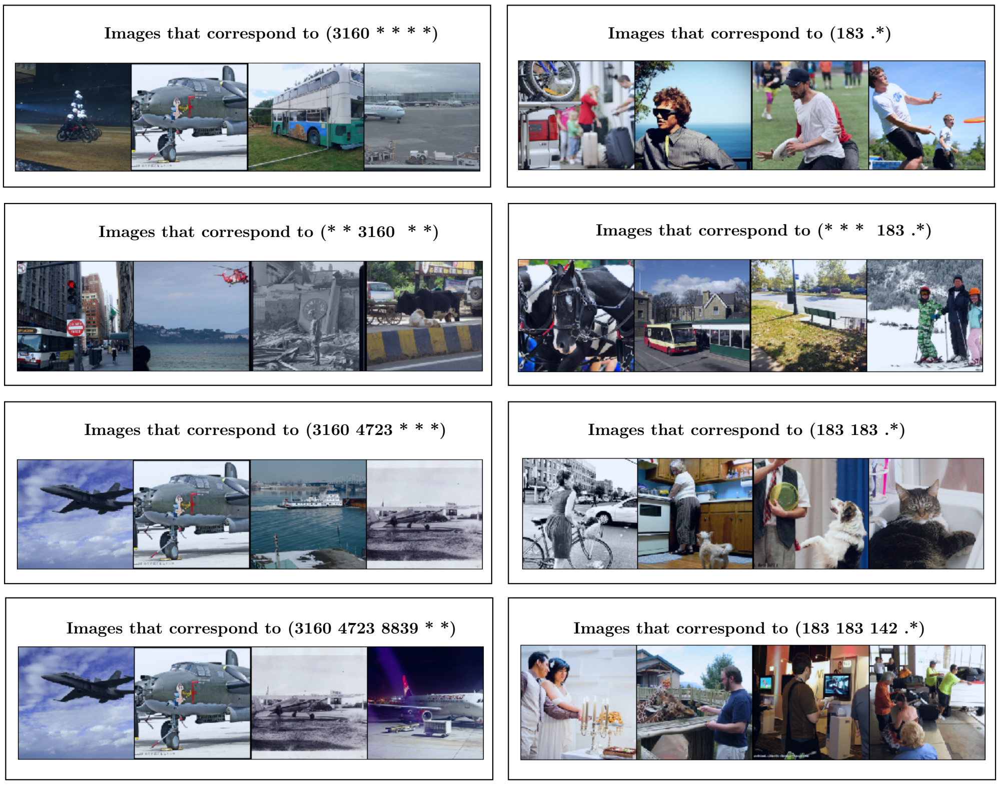

Images that correspond to (3160 * * * *)
Images that correspond to (183 .*)
Images that correspond to (* * 3160 * *)
Images that correspond to (* * * 183 .*)
Images that correspond to (3160 4723 * * *)
Images that correspond to (183 183 .*)
Images that correspond to (3160 4723 8839 * *)
Images that correspond to (183 183 142 .*)

FIGURE 2.3: **Samples from COCO test set that share similar codes.** The left column shows results from a model trained with maximum sequence length of 5. The right column shows results from the model trained with 14 maximum symbol message.

We continue with the qualitative analysis of the communication protocol as proposed by Havrylov and Titov [2017], in which the nature of the learned language is investigated. From the validation set, a random image is selected and a corresponding message is generated by the model. We then selected other images whose messages share the first 1,2 and 3 symbols with the previously selected image. Two such examples are illustrated in Figure 2.3. On the left column, we present the results from a model trained with a maximum sequence length of 5, and on the right column, with the sequence up to 14 symbols. The first row shows images that correspond to messages that start with 3160, and 183 respectively. As it can be seen, images correspond to vehicles, on the left-hand side, and people, on the right-hand side. However, images whose codes contain these symbols at arbitrary positions in the sequence, such as in the third place (* * 3160 * *), and fourth place (* * 183 .*), do not correspond to any specific category and do not seem to have any similarities. These results resemble Havrylov and Titov [2017]'s findings and suggest that the first symbol in the message encodes a predefined category, hence the order is important. The third row of Figure 2.3 shows images which have the same first two symbols in their codes: mostly airplanes, with the exception of a boat whose shape might be mistaken for one and has a blue background. For the other model, the results are more diverse because it has more freedom in the number of symbols it can use. Hence the images show both people with and without animals, as well as animals alone. The last row shows that the message (3160 4723 8839 * *) becomes more specialised and corresponds only to airplanes. Similarly, for the other model, (183 183 142 .*) seems to correspond to humans who have a common feature, they all reach their hands out and use them in some activity.

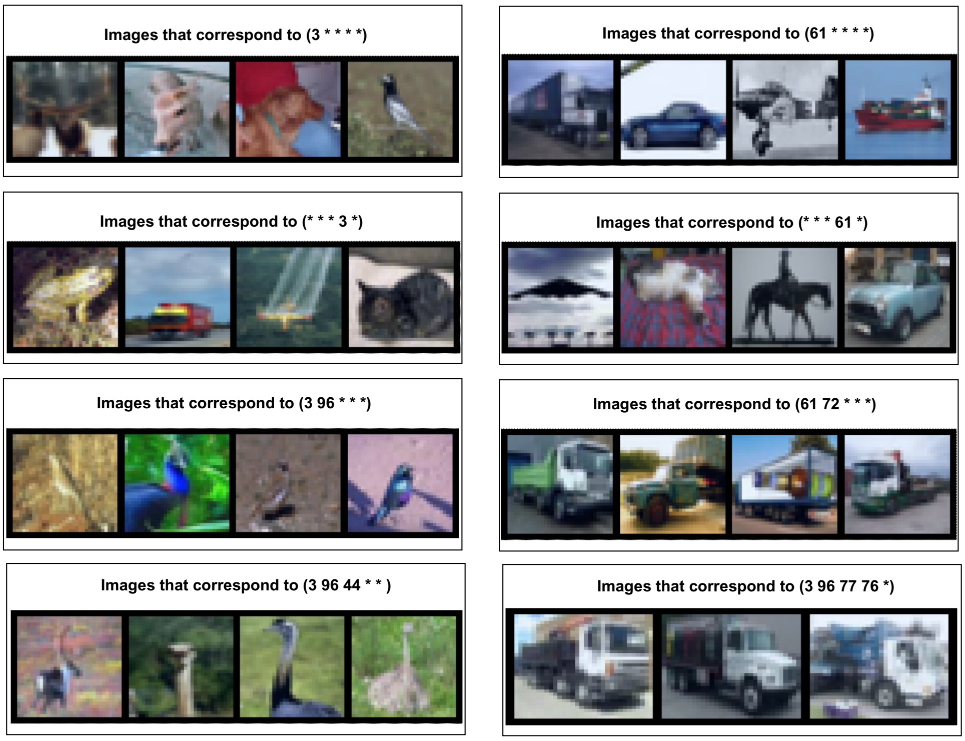

Images that correspond to (3 * * * *)
Images that correspond to (61 * * * *)
Images that correspond to (* * * 3 *)
Images that correspond to (* * * 61 *)
Images that correspond to (3 96 * * *)
Images that correspond to (61 72 * * *)
Images that correspond to (3 96 44 * *)
Images that correspond to (3 96 77 76 *)

FIGURE 2.4: **Samples from CIFAR-10 test set that share similar codes.** The two columns come from two versions of CIFAR-10 models, with differently pretrained feature extractors. LEFT: with supervised pretraining on ImageNet classification task; RIGHT: self-supervised pretraining with the SimCLR framework [Chen et al., 2020].

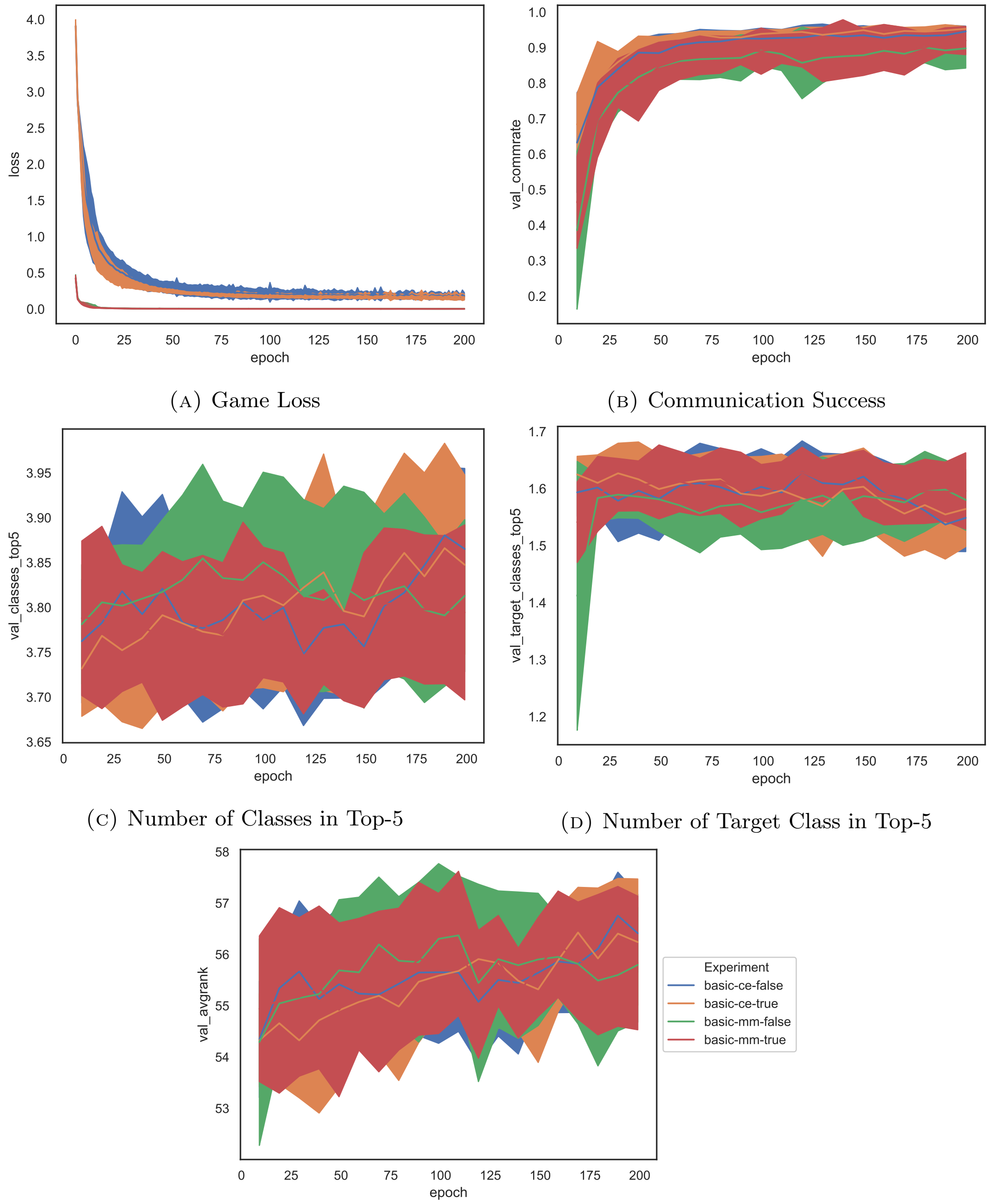
The same experiment was performed with some of our models trained on the CIFAR-10 dataset, following the setup detailed in Section 2.2.2. Figure 2.4 shows qualitative results of two models which have different pretrained feature extractor CNNs: one with a fully-supervised VGG16 network pretrained on ImageNet, and one pretrained with the SimCLR method [Chen et al., 2020] (see Section 1.1.2.4) in a completely self-supervised way. The results show that both models develop a “language” similar to that found by Havrylov and Titov [2017] in their original ImageNet setup. Hence, this once more assures us that proceeding with further experiments on a smaller dataset such CIFAR-10 should not compromise the learned communication protocol. Similar to the original model, the language developed by either variant presented in Figure 2.4 seems to follow some hierarchical encoding scheme.

2.3.2 Experimenting with different learning strategies

A further step in our experiments after having switched to CIFAR-10 dataset was to investigate the effect of the game loss. The decision taken in the original paper to use the hinge-loss objective seems somehow arbitrary and was not justified by the authors. Hence, we ran the original game with a multi-margin loss and with a cross-entropy loss on CIFAR-10 and show the comparison of different metrics in Figures 2.5a, 2.5b, 2.5c, 2.5d. The training details for our CIFAR-10 experiments are presented in Section 2.2.2. Similarly, the different metrics we compare in this experiment are detailed in Section 2.2.3.

Another factor we found very important for learning stability is gradient clipping. This method is commonly used when training recurrent networks and involves clipping the derivatives of the loss function to a given maximum or minimum value. For the chosen game objective, the two LSTMs can sometimes get the message wrong and hence mistake the correct image; however, this should not be allowed to overthrow all the good decisions made so far. Following common practices, we tried clipping the derivative of the loss function to a maximum value of 1.0. We tested this method for the models trained with the two different losses (multi-margin abbreviated as ‘mm’, and cross-entropy as ‘ce’ in the figure legends). In the plot legends, the true/false in the model name indicates if this technique was used or not.

We find that the multi-margin loss decreases much faster than cross-entropy (see Figure 2.5a). However, the communication success metric (which measures if the target image was guessed correctly by the receiver) shows a similar rate of improvement for both losses. Although the cross-entropy models reach a slightly higher communication rate, the difference is not significant and both variants seem to plateau after 75 epochs (see Figure 2.5b). For the metrics which quantitatively measure the amount of semantic information captured in the learned protocol, we provide results for the average rank (Figure 2.5e), the number of times the target class appears in top-5 choices (Figure 2.5d), and the number of different classes in top-5 (Figure 2.5c). Considering the semantic performance,



(A) Game Loss

(B) Communication Success

(C) Number of Classes in Top-5

(D) Number of Target Class in Top-5

(E) Average rank of Target Class

FIGURE 2.5: **Results for the basic game played on CIFAR-10 with different configurations.** Models were trained with either a multi-margin loss objective (mm) or cross-entropy (ce). True/false indicate if gradient clipping of the loss derivative was performed. For the models which employed this technique, a maximum value of 1.0 was used.

the models which use a multi-margin loss get better results. However, the model trained with a multi-margin loss and no gradient clipping seems more unstable than the rest. On the whole, the models which use gradient clipping prove to always perform slightly better.

2.3.3 Exploring other hyperparameters

In this section, we look at other parameters which could impact the learned communication protocol: the vocabulary size and the maximum message length.

2.3.3.1 The effect of the vocabulary size

In the original game played on COCO dataset, the suggested vocabulary size is 10000, and the maximum message length is 14 [Havrylov and Titov, 2017]. We tested if reducing the number of available symbols for the message (*i.e.* the sequence of symbols the sender generates to describe the image) influences the agents' performance. Figure 2.6 shows a comparison of two models trained on COCO, with the same maximum message length (*i.e.* L=14), but different vocabulary sizes, 10000 and 200. As can be seen, the model which has a smaller number of message symbols available ends up reaching a similar communication rate as the original model.

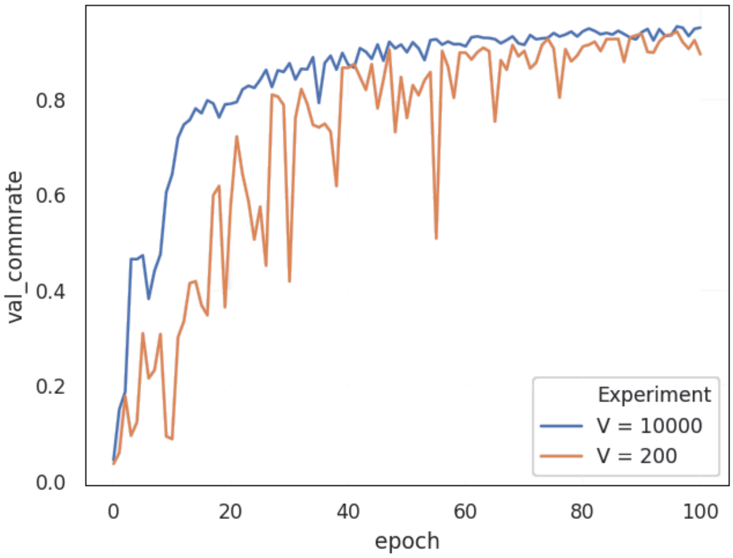


FIGURE 2.6: **Validation communication success measure for two configurations of the original model [Havrylov and Titov, 2017] tested on COCO in which the vocabulary size is varied.**

2.3.3.2 The effect of the message length

We then compared the performance of models trained with the same vocabulary size, but with various maximum possible lengths (L) of messages. Havrylov and Titov [2017] claim that for all models trained with L greater than 4, the communication success is very similar. However, they also show that the number of updates needed to converge for a model trained with L less than 5 is considerably higher (over 50k updates). Hence, because training the model on COCO dataset, with $L = 14$, was already resource-intensive, we

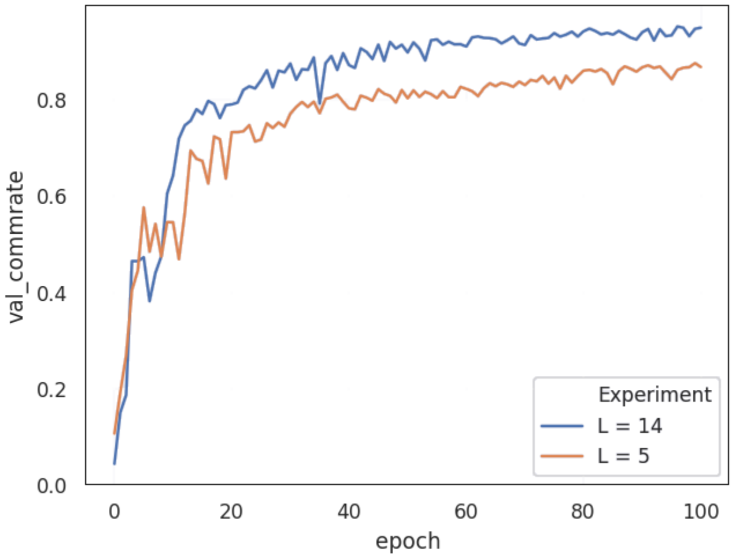


Figure 2.7: **Validation communication success measure for two configurations of the original model tested on COCO in which the maximum message length (L) is varied.**

only tested this hypothesis for a maximum message length of 5 and show the comparison with the initial model configuration in Figure 2.7. The validation communication rate of this model is indeed approaching the one achieved by the original model.

Next, in our CIFAR-10 experiments, since there is fewer data and the images contain only one class of objects, we reduced the size of the vocabulary to 200 and used a maximum message length of 5. More details of the setup of our CIFAR-10 experiment can be found in Section 2.2.2. We explored the influence of message length in our variant of the game with an additional task, which is discussed in detail in Section 2.7. Results (Figure 2.8) show that decreasing the maximum number of symbols which can be transmitted to identify the target image causes the communication success to drop. It is also worth noting that this modified game, which has two objectives, is definitely harder and hence allowing for a higher L improves the agents' performance on the image guessing task. This, however, seems to decrease the semantic information captured in the learned protocol which is shown by the semantics measures such as the average rank of the target image (Figure 2.9). Overall, this metric seems particularly sensitive to the choice of L. At the beginning of the game, using a longer message ($L = 10$), causes the model to have more difficulty in distinguishing images which correspond to the target class, but after 100 epochs it starts approaching the same values as the model trained with $L = 5$. On the other hand, using shorter message sequences seems to have the opposite effect: the model with $L = 3$ starts the game with a better average rank, but as it learns to play the game, the semantic metric values become worse which indicates that a hashing approach is being used instead.

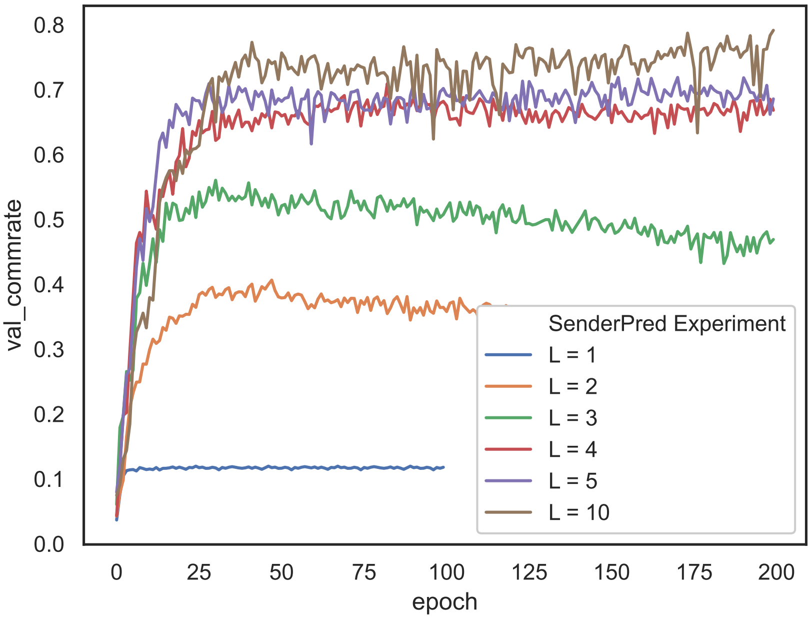


FIGURE 2.8: **Validation communication success for various configurations of our multiple-task game (the sender also has to predict the rotation of the input image) on CIFAR-10.** The models shown have been trained with different maximum message lengths (L). Results show that decreasing L causes the communication rate to drop.

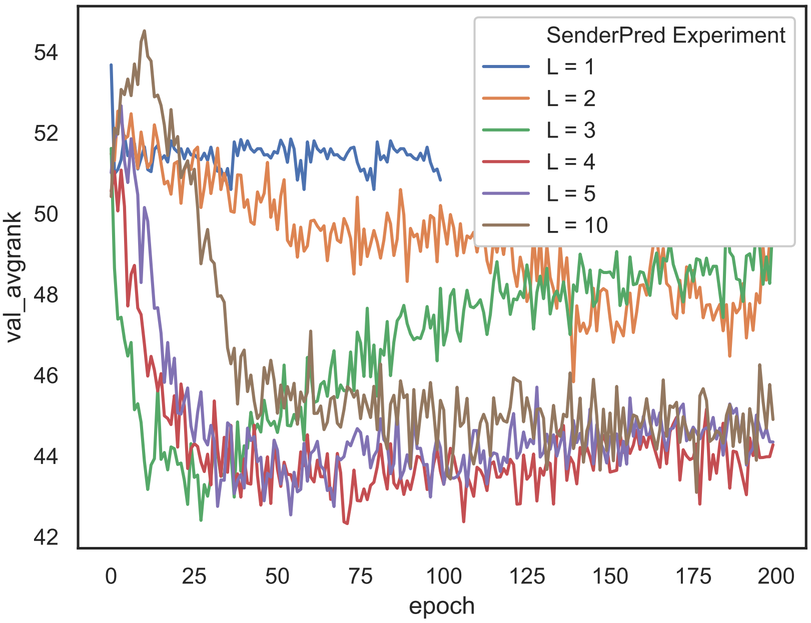


FIGURE 2.9: **Target average rank metric for multiple configurations of our multiple-task game in which the maximum message length is varied.** Increasing L also increases the average rank which suggests that, overall, it is more difficult to guess the target class when longer messages are used.

2.4 What is the Baseline Level of Semantics Under Different Conditions?

Generating and communicating hash codes is very clearly an optimal (if very unhuman) way to play the image guessing game successfully. However, in Havrylov and Titov [2017]'s original work there was qualitative evidence that did not happen when the model was trained, and that visual semantics were captured (see Section 2.3.1). To what extent is this caused by the pretrained feature extractor?

We attempt to answer this question by exploring different model variants: the original model with the CNN fixed and initialised with ImageNet weights; the CNN fixed, but initialised randomly; and, the CNN initialised randomly, but allowed to update its weights during training. In addition, we also tested the CNN pretrained on the SimCLR self-supervised task [Chen et al., 2020], fixing the weights or allowing them to modify during the game. Results from these experiments are summarised in Table 2.1. The first observation relates to the visual semantics measures; it is clear (and unsurprising) that the pretrained model captures the most semantics of all the models. It is also reasonable that we observe less semantic alignment with the end-to-end model; without external biases, this model should be expected to move towards a hashing solution. It is perhaps somewhat surprising however that the end-to-end model and the random model have a similar communication success rate, however, it is already known that a randomly initialised CNN can provide reasonable features [Saxe et al., 2011]. During training, the sender and receiver convergence had particularly low variance with both the end-to-end and random models, allowing the agents to quickly evolve a successful strategy. This is in contrast to the pretrained model which had markedly higher variance as can be seen from the plots in Figure 2.10.

Table 2.1: **The effect of different weights in the feature extractor CNN.** Measures are averaged across 7 runs of the game for each model on the CIFAR-10 validation set. Communication rate values in brackets are standard deviations across games, which show the sensitivity to different model initialisations and training runs. The message length standard deviation is measured across each game and averaged across the 7 runs, and shows how much variance there is in transmitted message length.

Feature extractor	Comm. rate	Message length	Top-5 comm. rate	#Target class in top-5	Target class avg. rank
Pretrained & fixed	0.90 (±0.02)	4.93 (±0.34)	1	1.86	46.25
Random & fixed	0.93 (±0.03)	4.90 (±0.39)	1	1.69	51.65
Learned end-end	0.94 (±0.02)	4.90 (±0.39)	1	1.5	57.14
Models with self-supervised pretrained feature extractor:					
Pretrained SS & fixed	0.84 (±0.02)	4.98 (±0.13)	0.99	2.24	39.50
Pretrained SS end-end	0.83 (±0.02)	4.97 (±0.21)	0.99	2.23	39.71

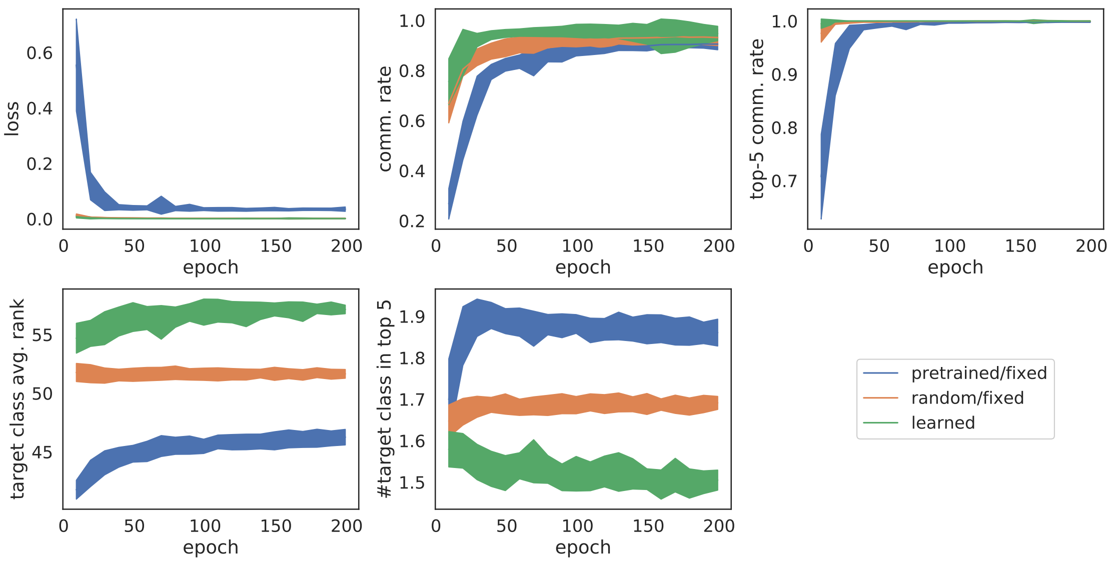


FIGURE 2.10: **The gameplay and semantic performance over the training epochs of the three model variants using a: pretrained, random or fully learned feature extractor CNN.** The loss plot shows that the learned and random models converge much faster than the pretrained one, and have lower variance allowing the agents to evolve a successful game strategy.

One might question if the end-to-end model would be handicapped because it had more weights to learn in the same number of epochs (200 for all models), however, as the results show, the end-to-end model actually has the best performance. We also investigated if the models required more training time, however training all the models for 1000 epochs yielded only a 2% improvement in communication rate across the board.

The additional two models which use a CNN pretrained on a self-supervised task seem to capture the most semantics, although there is not much difference if the network weights remain fixed or not during the game. The visual-semantics measures suggest that Chen et al. [2020]'s self-supervised approach yields more meaningful features which clearly capture more of the object (*i.e.* class) related information without compromising the game objective. The role of self-supervision in emergent communication environments is discussed in more detail in Section 2.7.

2.5 Making the Game Harder with Augmentation

We next investigate the behaviour of the same three model variants while playing a slightly more difficult game. The input image to the sender is randomly transformed, and thus will not be pixel-identical with any of those seen by the receiver. For the model to communicate well it must either capture the semantics or learn to generate highly-robust hash codes.

2.5.1 Noise and rotation

We start by utilising transformations made from random noise and random rotations. The added noise is generated from a normal distribution with mean 0 and variance 0.1, and the rotations applied to the input images are randomly chosen from {0°, 90°, 180°, 270°}.

The first part of Table 2.2 shows the effect of adding either noise or rotations, or both. In general, noise results in a slight increase in the communication success rate. More interestingly, for randomly rotated sender images the augmentation tends to increase the visual semantics captured by all the models, although this is most noticeable in the pretrained variant. At the same time, the communication success rate of the pretrained model drops; it is an open question as to whether this could be resolved by sending a longer message. Finally, the models augmented with both noise and rotations do not show any improvement over the rotation-only game in terms of the semantics measure. As one might guess, noise only makes the game harder, a fact which is reflected in the slight drop in the communication success, but does not explicitly encourage semantics.

TABLE 2.2: **The effect of different weights in the feature extractor CNN when the model is augmented** by adding noise and/or random rotations to the sender agent's input images, and when independently augmenting both agent's inputs images following the SimCLR framework [Chen et al., 2020]. Measures as per Table 2.1.

Feature extractor	Comm. rate	Message length	Top-5 comm. rate	#Target class in top-5	Target class avg. rank
Sender images augmented with Gaussian noise:					
Pretrained & fixed	0.89 (±0.02)	4.93 (±0.33)	0.99	1.86	46.39
Random & fixed	0.94 (±0.01)	4.90 (±0.38)	1	1.66	52.45
Learned end-end	0.94 (±0.02)	4.92 (±0.33)	1	1.51	57.33
Sender images augmented with random rotations:					
Pretrained & fixed	0.8 (±0.05)	4.94 (±0.32)	0.99	2.03	42.9
Random & fixed	0.80 (±0.12)	4.87 (±0.45)	0.98	1.7	51.43
Learned end-end	0.92 (±0.04)	4.92 (±0.32)	1	1.59	55.8
Sender images augmented with Gaussian noise and random rotations:					
Pretrained & fixed	0.76 (±0.02)	4.92 (±0.38)	0.98	2.01	42.85
Random & fixed	0.67 (±0.26)	4.77 (±0.57)	0.92	1.62	51.37
Learned end-end	0.90 (±0.06)	4.94 (±0.29)	1	1.58	55.8
Sender & receiver images independently augmented (SimCLR-like):					
Pretrained & fixed	0.48 (±0.03)	4.90 (±0.41)	0.86	2.14	38.08
Random & fixed	0.42 (±0.10)	4.92 (±0.33)	0.85	1.68	47.94
Learned end-end	0.72 (±0.05)	4.91 (±0.39)	0.98	2.00	42.37
Pretrained SS & fixed	0.43 (±0.49)	4.82 (±0.54)	0.86	2.11	39.00
Pretrained SS end-end	0.50 (±0.5)	4.95 (±0.27)	0.90	2.19	39.18

2.5.2 More complex transformations

We continue by adding a more complex composition of data augmentations to the game. Chen et al. [2020] have recently shown that combinations of multiple data augmentation operations have a critical role in contrastive self-supervised learning algorithms and improve the quality of the learned representations. We implement their transformation setup in our game, with sender and receiver having differently augmented views of the same image. We follow the combination proposed by Chen et al. for the CIFAR-10 experiment which consists in sequentially applying: random cropping (with flip and resize to the original image size) and random colour distortions[4]. We test if the combination does improve the learned representations in a self-supervised framework as ours, which however does not use a contrastive loss in the latent space, but the aforementioned hinge-loss objective (see Section 2.2.1). It is also worth noting that we continue using a VGG16 feature extractor, as opposed to the ResNet [He et al., 2016] variants used by Chen et al. [2020]. The game is played as described in Section 2.2, but this time each image is randomly transformed twice, giving two completely independent views of the same example, hence, making the game objective harder than with the noise and rotation transformations[5].

The bottom part of Table 2.2 shows the results of the newly-augmented game for the different configurations of feature extractors used previously (pretrained with ImageNet and fixed; random and fixed; and, learned end-to-end) and two additional models, which have a VGG feature extractor pretrained with the self-supervised SimCLR framework. The results show that, indeed, by extending the augmentations and composing them randomly and independently for sender and receiver, the communication task becomes harder, hence the communication success is lower than in the previous experiments. However, as Chen et al. [2020]'s results have also shown, the quality of the representations improves considerably, especially for the model 'Learned end-end', and this is reflected in the improvement of our measures for the amount of semantic information captured in the learned communication protocol. Specifically, the number of times the target class appears in top-5 predictions increases by almost half a point for the pretrained and learned model, and the average rank of the target class lowers (over 10 units for the learned model) which indicates that the protocol captures more content information and is less susceptible to only hashing the images. Using this approach, the learned model achieves the highest communication success while also getting semantic results close to the model with an ImageNet pretrained feature extractor. In this setup, however, the two additional models with self-supervised feature extractors do no provide any improvement

[4] The details of the data augmentations are provided in the appendix of Chen et al. [2020] and available at `https://github.com/google-research/simclr`

[5] In the noise and rotation case only the sender's image was transformed. It is conceivable in this case that the sender might learn to de-noise or un-rotate the feature in order to establish a communication protocol. If images are transformed on both sides of the model, the agents won't have an easy way of learning a 'correct' inverse transform.

over the model with a fully supervised feature extractor CNN, as was the case in the previous experiment presented in Section 2.4. This could be influenced by the capacity of the bottleneck; further experiments are needed to investigate this.

It is particularly interesting to observe that by the relative simplicity of applying the same transformations to the images as Chen et al. [2020] we encourage semantic alignment in a completely different model architecture and loss function. This suggests that the value of Chen et al.'s proposal for contrastive learning is more towards the choice of features rather than the particular contrastive loss methodology.

2.6 Making the Game Harder with Multiple Objectives

The experimental results with the model setups shown in Tables 2.1 and 2.2 clearly show that the fully-learned models always collapse towards gameplay solutions which are not aligned with human notations of visual semantics. Conversely, the use of a network that was pretrained in a supervised fashion to classify real-world images has a positive effect on the ability of the communication system to capture visual semantics. On the other hand, using a different experimental setup involving a complex set of independent transformations of the images given to the sender and receiver helps the learned model acquire and use more of the visual-semantic information, similar to the pretrained model. However, this improvement comes at the cost of reducing the communication success rate as the game becomes much harder when using the proposed augmentations. We continue by exploring if it might be possible for a communication protocol with notions of visual semantics to emerge directly from pure self-supervised gameplay. To achieve this, we propose that the agents should not only learn to play the referential game, but they should also be able to play other games (or solve other tasks). In our initial experiments, we formulate a setup where the agents not only have to play the augmented version of the game described in Section 2.5 (with both noise and rotations randomly applied to the image given to the sender, but not the receiver) but also one of the agents has to guess the rotation of the image given to the sender as shown in Figures 2.11 and 2.12.

This choice of the additional task is motivated by Gidaris et al. [2018] who showed that a self-supervised rotation prediction task could lead to good features for transfer learning, on the premise that in order to predict rotation the model needed to recognise the object. The rotation prediction network consists of three linear layers with Batch Normalisation before the activation functions. The first two layers use ReLU activations, and the final layer uses a Softmax to predict the probability of the four possible rotation classes. With the exception of the final layer, each layer outputs 200-dimensional vectors. Cross-Entropy is used as the loss function for the rotation prediction task ($\mathcal{L}_{rotation}$). All other model parameters and the game-loss definition match those described in Section 2.2.

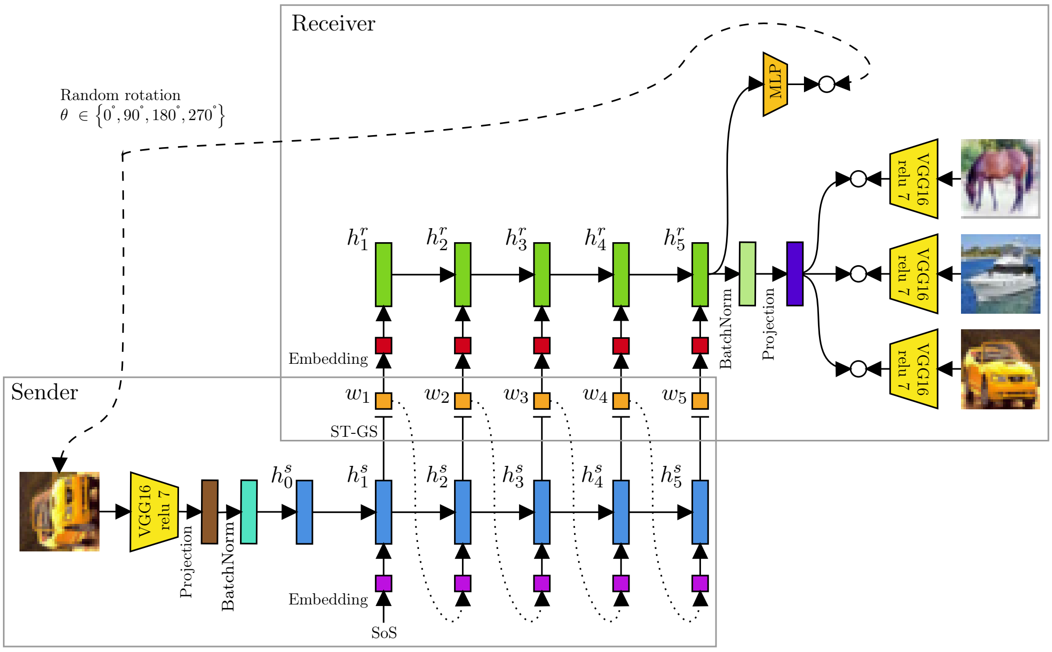


Figure 2.11: **Extended game with the receiver also required to guess the orientation of the sender's image.**

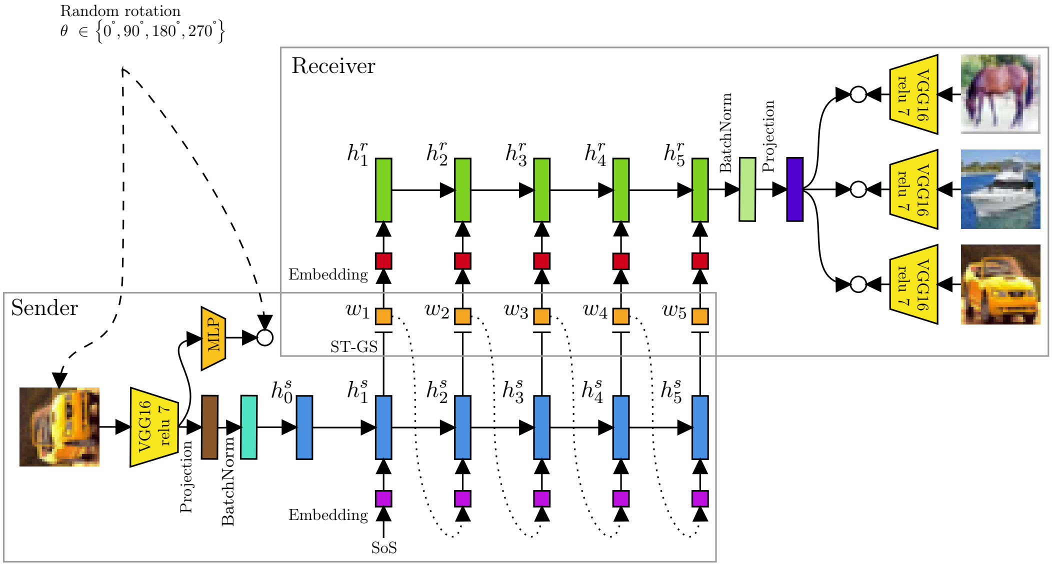


Figure 2.12: **Extended game with the sender augmented with an additional loss based on predicting the orientation of the input image.**

The results of these experiments are shown in Table 2.3. It should be noted that such models are difficult to train, hence multiple runs would be extremely time-consuming. We ran a series of experiments to find optimal weightings for the two losses such that the models succeed at the communication task while also acquiring notions of visual semantics. Both experiments presented, with the Sender-Predicts model (Figure 2.12) and the Receiver-Predicts model (Figure 2.11), used a weighted addition $0.5 \cdot \mathcal{L}_{rotation} + \mathcal{L}_{game}$,

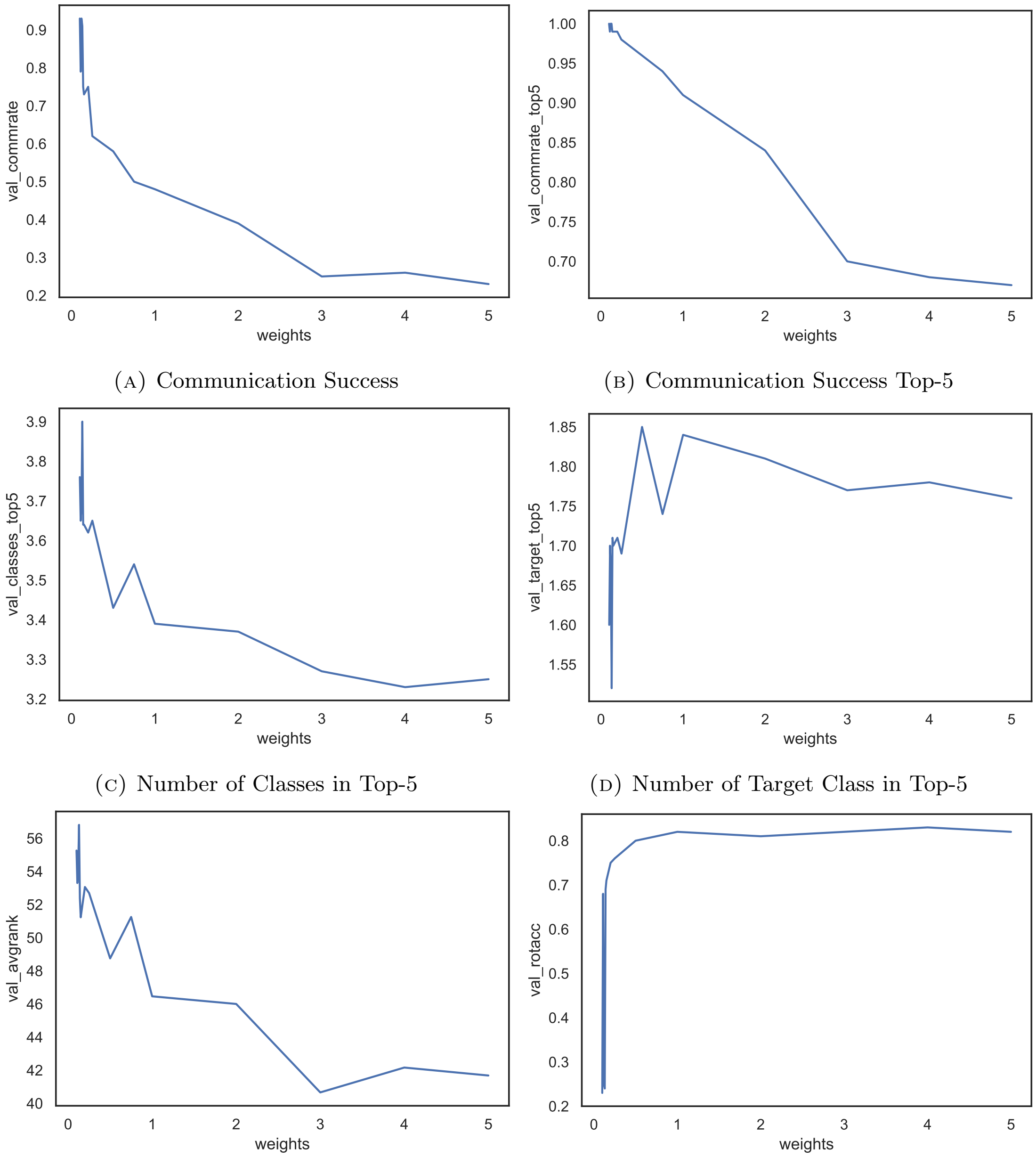
Table 2.3: **End-to-end learned models with an additional rotation prediction task.** Measures as per Table 2.1, except for the inclusion of the accuracy of rotation prediction.

Model	Comm. rate	Top-5 comm. rate	#Target class in top-5	Target class avg. rank	Rot. acc.
Receiver-Predicts (Fig. 2.11)	0.58	0.96	1.85	48.75	0.80
Sender-Predicts (Fig. 2.12)	0.72	0.98	2.05	42.89	0.83

where $\mathcal{L}_{game}$ refers to the original hinge-loss objective for the game proposed by Havrylov and Titov [2017]. For the latter model we also tried using additive loss with learned weights (following Kendall et al. [2018]) however this created a model with good gameplay performance, but an inability to predict rotation (and poor semantic representation ability).

The rotation loss weight was chosen from a variety of weightings, ranging from 0.1 to 5. The choice was determined by a series of experiments ran to find an optimal weighting such that the models succeed at the communication task while also acquiring notions of visual semantics. Figures 2.13 and 2.14 show the effect that various weightings of $\mathcal{L}_{rotation}$ have on the game metrics for the Receiver-Predicts, and respectively, the Sender-Predicts models. Naturally, there is a trade-off between the top-1 communication rate (Figures 2.13a, respectively 2.14a) and the metrics which quantitatively measure visual semantics (number of classes in top-5, occurrences of target class in top-5 and target class average rank in the batch). However, a similar trade-off does not occur between the top-5 communication rate (Figure 2.13b, respectively Figure 2.14b) and semantics. If the semantics improve, it implicitly means that more of the object category is captured in the learned language. As previously mentioned in Section 2.2.3, if the model only transmitted information about the object, the top-5 communication rate would be on average 0.36. Since this metric is significantly higher, it implies that the message must contain additional information, beyond the type of object. This could be visual semantics such as attributes of the object, but it could also just be a more robust hashing scheme based on pixel or low-level feature values.

Training these models is harder than the original sender-receiver model because the gradients pull the visual feature extractor in different directions; the game achieves good performance when the features behave like hash codes, whereas the rotation prediction task requires much more structured features. This conflict means that it is difficult to train models that can solve both tasks concurrently. Further work in developing optimisation strategies for these multi-game models is of critical importance in the future.



(A) Communication Success

(B) Communication Success Top-5

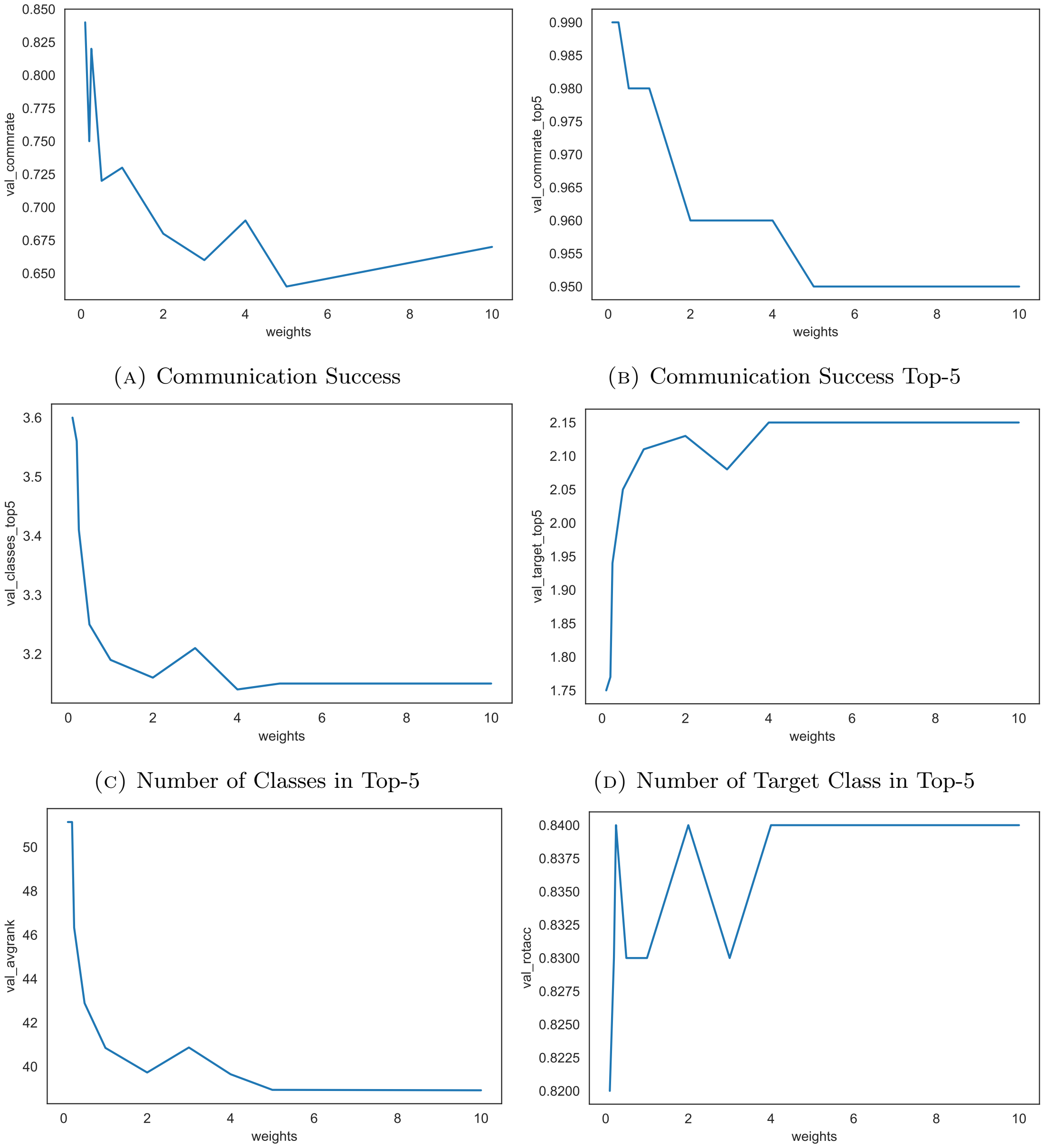
(C) Number of Classes in Top-5

(D) Number of Target Class in Top-5

(E) Average rank of Target Class

(F) Rotation Prediction Accuracy

FIGURE 2.13: **The trade-off between various weights of the $\mathcal{L}_{rotation}$ and the different metrics for the Receiver-Predicts model.**



(A) Communication Success

(B) Communication Success Top-5

(C) Number of Classes in Top-5

(D) Number of Target Class in Top-5

(E) Average rank of Target Class

(F) Rotation Prediction Accuracy

FIGURE 2.14: **The trade-off between various weights of the $\mathcal{L}_{rotation}$ and the different metrics for the Sender-Predicts model.**

Whilst there is still a way to go to achieve the best levels of gameplay performance shown in Tables 2.1 and 2.2, it is clear that these fully self-supervised end-to-end trained models can both learn a communication system to play the game(s) that diverges from a hashing solution towards something that better captures semantics. The lower gameplay performance might however just be a trade-off one has to live with when encouraging semantics with a fixed maximum message length; this is discussed further at the end of the following subsection.

2.7 Playing Games with Self-Supervised Pretraining

Having observed that a completely learned model, with the right augmentations or instructed to solve multiple tasks which enforce notions of 'objectness', can already acquire some visual semantics, we end by exploring the effect of combining these two approaches: the multi-task game described in Section 2.6 with the previously mentioned self-supervised SimCLR framework [Chen et al., 2020]. The goal of this is to test whether a pretrained feature extractor, also trained on a task which does not require human intervention, can further improve the meaning of the communication protocol, pushing it towards a more human-like version. In previous experiments, we showed that models with a self-supervised pretrained feature extractor can achieve impressive gameplay performance, while learning a semantically grounded communication protocol (see Table 2.1). However, in the more complex game setup presented in Section 2.5, these models also had difficulty in solving the communication task (see Table 2.2). In this set of experiments, the Sender-Predicts model described in Section 2.6 is used. We employ independent augmentations for the sender and receiver agents that match those detailed in the second half of Section 2.5. To some extent, this resembles Lowe et al. [2020]'s supervised self-play approach in which self-play in a multi-agent communication game and expert knowledge are interleaved. In our case, however, the VGG16 feature extractor network was pretrained with Chen et al. [2020]'s framework in a completely self-supervised way.

The results of the multi-objective game played with the Sender-Predicts model, in the initial setup and with the modified SimCLR transforms, are presented in Table 2.4. We compare the different types of weights in the feature extractor again: learned end-to-end, pretrained in a self-supervised way and fixed, or allowed to change during the gameplay. In addition to the models pretrained with the SimCLR framework, we also show examples with feature extractors pretrained on a rotation prediction task (see 'Pretrained Rot' models in Table 2.4). The motivation for running this type of local task first is to somehow 'bootstrap' the agents with semantic knowledge about the world. For the games which only start with a self-supervised pretrained VGG16, we chose to fix the weights of the feature extractor for the first 5 epochs before allowing any updates. This was based on empirical results which showed that it helped to stabilise the LSTM and Gumbel-softmax

Table 2.4: **The effect of interleaving self-supervision and multi-agent gameplay.** The game setup has two tasks, sender predicting rotation as per Table 2.3 while using various augmentations (original and SimCLR same or individual). Models with pretrained feature extractor networks have been trained with either the SimCLR framework ('Pretrained SS') or with a rotation prediction objective ('Pretrained Rot').

Feature Extractor	Comm. rate	Top-5 comm. rate	#Target class in top-5	Target class avg. rank	Rot. acc.
Sender & receiver images augmented with the original transforms:					
Learned end-end	0.72	0.98	2.05	42.89	0.83
Pretrained SS end-end	0.84	0.99	2.19	40.19	0.79
Pretrained SS & fixed	0.80	0.99	2.23	39.72	0.7
Pretrained Rot end-end	0.77	0.98	2.11	40.32	0.86
Pretrained Rot & fixed	0.73	0.96	2.12	40.13	0.87
Sender & receiver images augmented with SimCLR transforms:					
Learned end-end	0.53	0.92	2.22	37.16	0.80
Pretrained SS end-end	0.49	0.89	2.18	38.74	0.79
Pretrained SS & fixed	0.42	0.85	2.14	39.57	0.78

part of the models before allowing the gradients to flow through the pretrained feature extractor part. We hypothesise that this is due to the risk of bad initialisation in the LSTMs which can cause the models to fail to converge at the communication task. This observation can be generalised over all the experiments in this chapter, as all the models with a fixed feature extractor appear to be slightly more unstable than those with learned ones, in contrast to fully learned models which always converged (see Figure 2.10). As the results show, the model which best captures visual semantics is the one learned end-to-end using the SimCLR transforms. It is again obvious that between the two setups, the second makes the game significantly harder as the agents are now also required to extract and encode information about the object orientation, on top of seeing independently augmented input images. This is reflected in the drop in the top-1 communication success, although this does not hold for the top-5 rate.

Another interesting observation is that using a self-supervised pretrained feature extractor (with either the contrastive objective or the rotation prediction task) in the original setup helps improve the communication success and the semantics measures at the same time. This finding confirms that self-supervised pretraining in this type of game can be as beneficial, or even better, as the supervised pretraining on ImageNet used in a less complex variant of the game (see Table 2.2). Finally, we would like to point out that in the case of models with the feature extractor network pretrained on a rotation prediction task, preserve that knowledge during the main communication game; as a result, 'Pretrained Rot' models achieve the highest rotation accuracy.

2.8 Individual Visual Systems

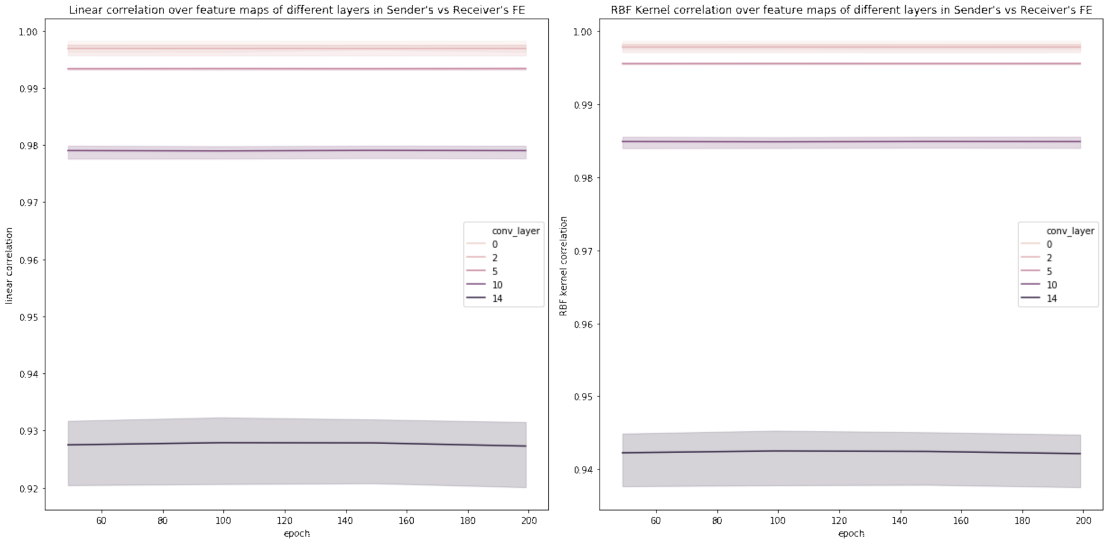
We conclude the experiments in this chapter by investigating whether a communication protocol can be established between agents which have distinct and independently pre-trained feature-extraction (FE) networks. As pointed out in Section 2.2, the sender and receiver agents share the feature extractor network and in related approaches, this used to be an ImageNet- pretrained CNN such as VGG16. Our intuition for choosing to model the agents with the same visual system is based on the idea that in the real world, humans have developed *shared* conventions when interpreting and communicating about visual scenes. As we have shown through a series of experiments, it is possible to capture visual semantics by learning the feature extractor while playing an image referential game. However, this approach can still be questioned for the condition of the two agents sharing the same visual system which, translated to the real world, would imply that any two humans see and perceive the visual world in the exact same way.

For this experiment, the basic game setup described in Section 2.2 was used. Using the SimCLR method [Chen et al., 2020], two distinct feature extractors were trained on two different subsets of the CIFAR-10 dataset. When evaluated with a linear classifier, the features learned by each of these models gave a performance of around 80%. The sender and receiver now have distinct visual perception experience. To allow the weights of the LSTM parts of the agents to stabilise at the beginning of the game, the weights of the pre-trained FEs are frozen for the first 5 epochs. After that, these weights are also allowed to modify during training. It is important to mention that, because in this setup the two agents have different 'visual systems', when we unlock the FEs weights, the optimisation requires a smaller learning rate for communication to be established. We found that decreasing the learning rate from 1e−3 to 1e−4 was enough for the agents to converge to a stable communication protocol.

As the results in Table 2.5 show, the communication success rate of two agents with distinct feature extraction networks is smaller than in the case of sharing this part of the architecture. On the other hand, the semantics measures have improved. One could conjuncture that having anchored each agent with some notions of visual semantics by

Table 2.5: **Comparison between models sharing the Feature Extraction (FE) network from Table 2.1 and models with distinct FEs.** In both cases, the FEs have previously been trained with SimCLR and allowed to modify weights during the communication game play. Measures as per Table 2.1.

Feature extractor	Comm. rate	Message length	Top-5 comm. rate	#Target class in top-5	Target class avg. rank
Same FE	0.83 (±0.02)	4.97 (±0.21)	0.99	2.23	39.71
Distinct FEs	0.73 (±0.05)	4.76 (±0.61)	0.98	2.45	34.47



(A) Similarity with Linear CKA

(B) Similarity with RBF Kernel CKA

FIGURE 2.15: **Measuring similarity over feature maps for the two distinct FE networks when showing the same visual input.**

training their feature extraction part with the SimCLR self-supervised method increases the probability of them transmitting messages which are also more semantically grounded than when only one such network was pre-trained and shared.

Finally, we measure the correlation over feature maps from the same layer in the two agents' FE networks. Kornblith et al. [2019]'s method was used to measure the similarity between layers in different trained models. Results of CKA with a linear kernel and an RBF kernel are shown in Figure 2.15. The similarity was averaged over the 10000 test images of CIFAR-10 and plotted against training time. However, playing the communication game does not seem to influence the feature extraction networks' representations in any way. As can be seen, the correlation between the first two convolution layers started at almost 100%, while for later layers this becomes lower. These results suggest that even though the FEs have been independently pre-trained on distinct subsets of CIFAR-10 and later allowed to update during the communication game, they learn similar representations of the input.

2.9 Discussion

In this chapter, we have explored different factors that influence the human interpretability of a communication protocol, that emerges from a pair of agents learning to play a referential signalling game with natural images. We first quantify the effect that using a pretrained visual feature extractor has on the ability of the language to capture visual semantics. We empirically showed that using pretrained feature extractor weights from a supervised task inductively biases the emergent communication channel to become more semantically aligned. We also showed that both random-fixed and learned feature extractors have less semantic alignment, but better gameplay ability due to their ability to learn hashing schemes that robustly identify particular images using very low-level information.

We then performed an analysis of the effect that different forms of data augmentation and transformation have on the agents' ability to communicate object semantics. Adding zero-mean Gaussian noise into the sender's image does not serve to improve the semantic alignment of messages but does perhaps have a mild effect on improving the robustness of the hashing scheme learned by the models. The addition of rotation to the sender's image results in a mild improvement in the semantic alignment, although in the case of the models with fixed feature extractors this is at the cost of gameplay success rate. More complex combinations of data transforms, applied independently to the sender's image and receiver's images, are demonstrated to give a sizeable boost to the visual semantic alignment for the model learned in an end-to-end fashion.

We then demonstrated that it is possible to formulate a multiple-game setting in which the emergent language is *more* semantically grounded also without the need for any outside supervision. We note these models represent difficult multi-task learning problems, and that the next steps in this direction would benefit from full consideration of multi-task learning approaches which deal with multiple objectives that conflict [e.g. Sener and Koltun, 2018; Kendall et al., 2018].

Finally, we have shown that pretraining the visual feature extractor on a self-supervised task, such as that of Chen et al. [2020] or Gidaris et al. [2018], can further improve the quality of the semantics notions captured by a fully learned model. One way of looking at self-supervised pretraining is to consider it as self-play of a different game, before engaging in the main communication task/game. From this point of view, further work in the area of emergent communication should explore other combinations of self-supervised tasks. Creating environments in which agents have to solve multiple tasks, concurrently or sequentially, while using the correct type of data augmentations seems to balance the trade-off between performing the task well and developing a communication protocol interpretable by humans. As Lowe et al. [2020] has also shown, interleaving supervision and self-play can benefit multi-agent tasks while reducing the amount of necessary human intervention.

Dessì et al. [2021]'s study, published after our work, also looked at the role of augmentations to improve semantics emergence in the communication protocol and, similarly to the findings presented in this chapter, confirmed the benefits of integrating self-supervised learning methods into emergent communication research. Dessì et al. [2021] explored communication via a one symbol bottleneck about a large number of concepts, by training on the ILSVRC-2012 dataset with 1000 categories. In a similar manner, it showed that communication can emerge even when the agents' visual systems are not anchored in an object recognition task, but instead, are being learnt from scratch. Once more, it confirmed that this is a promising avenue of research which would be worth revisiting in the future.

However, as concluded in a number of investigations on the emergence of a grounded human-interpretable language [Kottur et al., 2017; Bouchacourt and Baroni, 2018; Lowe et al., 2019; Dessì et al., 2021], the problem of interpretability is only partially assessed through preliminary quantitative and qualitative experiments. The development of embodied intelligent agents that can communicate among themselves and with humans ultimately requires experimental results with *humans involved in the process*.

To facilitate the communication between machines and human participants, in the next chapters we will explore a more directly interpretable means of communication which does not require mapping discrete messages, such as 1-symbol messages, to category names. The means of communication we propose goes back to prehistory and nowadays is considered universally understood: drawing and sketching.

Chapter 3

Differentiable Drawing and Sketching

"Drawing is the root of everything."

— Vincent van Gogh

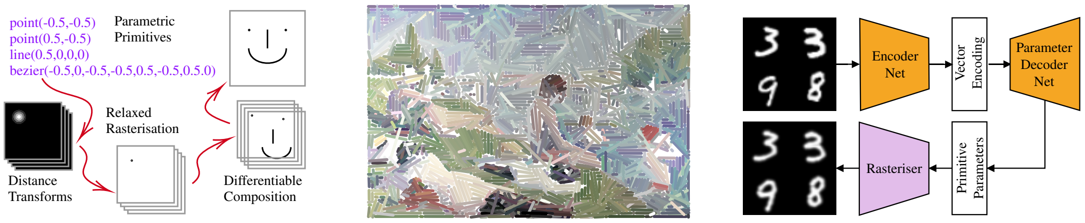
"All you need to paint is a few tools, a little instruction, and a vision in your mind."

— Bob Ross

Rather than an abstract language of tokens, as we have explored in the previous chapter, we now turn our attention to other types of communication that could be more easily interpretable to a human observer. If one were to look back at how humans developed written communication, they will come across ancestral drawings and sketches made on cave walls, stones and animal bones. These were created using different mediums, such as charcoal for black and hematite for red, and with various instruments (fingers, wooden sticks or flint stones).

Machine learning techniques such as generative models [Goodfellow et al., 2014] and neural style transfer models [Gatys et al., 2016; Jing et al., 2019] have extensively been used to produce sketch-like images and mimic the artistic behaviour of humans. However, such approaches are very unrealistic. When humans use drawing, sketching and writing to communicate they rarely do so by filling in pixels on a grid. Most methods (with some notable exceptions) of producing physically realised forms of drawing and writing by hand involve manipulating an instrument (a pen, paintbrush, pastel, etc) to mark a surface (paper, for example). In the digital world, this process is often approximated with vector graphics, in which paths are 'stroked' and then most often rasterised onto a pixel grid to produce digital images that can be displayed on a monitor or reproduced in hard copy.

To date, modelling the act of drawing with techniques such as deep neural networks has been relatively limited because the process of rasterisation using traditional approaches is not differentiable. The vast majority of recent work on image generation has operated



(A) Overview of the proposed approach to drawing. The final image is differentiable with respect to the primitive parameters.

(B) Seurat's 'Une baignade à Asnières' reduced to straight lines instead of points by gradient descent through the rasteriser.

(C) Encoder-Decoder-Rasteriser model for learning to map images to primitives. Orange blocks have learnable parameters.

FIGURE 3.1: **With a differentiable rasteriser (a), it is possible to optimise primitives (b), and build end-to-end learnable models (c).**

on the principle of trying to optimise outputs broadly at the pixel level utilising tools such as transpose convolutions which operate on raster representations. There are of course exceptions to this statement, where researchers have attempted to more closely consider the underlying process that humans use to draw and write [*e.g.* Lake et al., 2015], or to circumvent the non-differentiability of rasterisation [*e.g.* Zheng et al., 2019] using learning. These techniques, as well as a contemporaneous approach to relaxing modern vector graphics [Li et al., 2020] (taking a complementary, but different approach to ours) which was published during the production of this chapter, are described and discussed in Section 3.1.

Therefore, in order to build artificial agents that can communicate through drawing in a human-like fashion, which is not by turning pixels on and off, modelling the physical act of drawing is an extremely important part. This chapter proposes a differentiable relaxation of the rasterisation process, which we ultimately demonstrate allows us to build end-to-end learnable machines that can perform both image generation and inference tasks. More concretely, we demonstrate that we can build machines that turn digital raster images into parametric representations of continuous paths, and back into rasterised images again. Our approach is not constrained by any particular modelling assumptions about how images should be composed beyond the functions used being differentiable. This allows us to closely model the physical act of drawing with a pen on paper. We believe that the proposed approach will ultimately have many applications in future approaches to computer vision tasks related to topics including sketch retrieval, recognition, and generation, as well as topics related to understanding and analysing handwriting, and even to more general topics around understanding visual communication.

The main contributions of this chapter are as follows:

1. We present a bottom-up differentiable approach (see Figure 3.1a) to generating pixel rasters from parameterised vector primitives by reformulating and relaxing the rasterisation problem. This is coupled with a set of formulations that allow different approaches to composition. Our approach is detailed in Section 3.2.

2. We demonstrate that primitives can be optimised by minimising a loss against an existing raster image (*e.g.* Figure 3.1b), and show how different losses inform the result. Details are in Section 3.3.

3. Moreover, in Section 3.3.2.3 we show how the optimised vector primitives can be used to create physical sketches by learning programs to control a drawing robot.

4. A range of parameterisations of drawing primitives are defined and applied in end-to-end learnable autoencoder architectures (see Figure 3.1c). These are tested on a variety of handwritten/drawn image datasets and the performance is objectively compared in Section 3.4.

5. A PyTorch implementation of our approach, which allows others to experiment further, is available at `https://github.com/jonhare/DifferentiableSketching`.

3.1 The Current State of Sketch Research

Drawing, and in particular sketching, has been a means of conveying concepts, objects and stories since ancient times. There is a long history of sketch research in computer vision and human-computer interaction dating back to the 1960s [Sutherland, 1964]. Sketch applications have become increased in recent years due to the rapid development of deep-learning techniques that can successfully tackle tasks such as sketch recognition [Yu et al., 2017], generation [Zheng et al., 2019; Ha and Eck, 2018; Sangkloy et al., 2017], sketch-based retrieval [Choi et al., 2019; Sangkloy et al., 2016; Creswell and Bharath, 2016], semantic segmentation [Wu et al., 2018; Yang et al., 2020], grouping [Li et al., 2018], parsing [Sarvadevabhatla et al., 2017], sketch transformers [Ribeiro et al., 2020; Xu et al., 2021] and abstraction [Muhammad et al., 2018]. Xu et al. [2022] offer a recent and detailed survey of free-hand sketch research and applications, focusing on contemporary deep-learning techniques.

Sketches for digital analysis can be represented and saved in very diverse formats: as raster images made up of sparse matrices (black backgrounds with white lines) or dense matrices (white background with black lines), as graphs or vectors of sequences of strokes or euclidean coordinates that encode topological and temporal patterns [Xu et al., 2022]. As a result of the variety of possible representations, there also exist various tools to process sketches including deep learning tools such as CNNs, RNNs, GNNs and TCNNs. However, of all the modalities of representing sketches, the one that most closely mimics how humans sketch is that of vector representation and generation of sketches as a sequence of points, line or curve segments.

Our long-term goal for the research presented in this chapter is to be able to train models to learn how to produce the parameters of drawing primitives based on visual inputs with only limited supervision. Internally within our models, we want to bridge the gap between input and output rasters, and internal vector representations.

3.1.1 Vector graphics creation

In the field of computer graphics, vector graphics is a representation format which uses compact and resolution-independent mathematical shapes, such as points, lines and curves [Salomon, 2007], to create visual images. Some of the advantages of representing images as vector outlines include the possibility to scale to any size, without loss of image quality, and the independence from the output device's resolution [Selinger, 2003]. Vector graphics are commonly used in fonts, designing logos, user interfaces, web design and engineering design (*e.g.* architectural plans) as they are more easily edited, stylised and animated through the manipulation of the underlying geometry [Adobe, 2022; Autodesk, 2022; Homestyler, 2022].

Most inputs and outputs of hardware devices, however, are processed in bitmap format, or raster graphics, which describe visual images as grids of pixels rather than via geometry. *Rendering*, also known as rasterisation, is the process of transforming vector graphics to bitmap format, while *tracing* is the reverse process.

3.1.1.1 Traditional image vectorisation methods

The automatic creation of vector graphics has fallen behind compared to the powerful tools of machine learning, such as CNNs, that can learn to create and manipulate raster images. With respect to models that turn raster images into vectors, there is considerable classical literature looking at the problem of 'stroke-based rendering' where the objective is to turn raster images into a sequence of strokes [*e.g.* Hertzmann, 1998; Winkenbach and Salesin, 1994] for artistic or visual communication purposes. A good overview of these can be found in the tutorial by Hertzmann [2003], which breaks these approaches into Voronoi (broadly based on Lloyd's algorithm [Lloyd, 1982]), or 'trial and error' approaches which try to minimise a loss based on heuristic tests.

Traditional methods of vector graphics generation from raster images usually require segmenting the raster image into regions and then applying specialised tracing algorithms to capture edges and fit the contours with various vector primitives. Such vector generation algorithms range from fitting splines or Bézier curves [Selinger, 2003; Lecot and Levy, 2006; Xia et al., 2009], diffusion curves [Orzan et al., 2008; Xie et al., 2014], other geometric shapes such as polygons or triangles [Swaminarayan and Prasad, 2006; Demaret et al., 2006], to using gradient meshes [Sun et al., 2007; Lai et al., 2009] to approximate vector outlines.

To mimic the physical continuous strokes made by a human with a drawing instrument, vector format is probably the most sensible approach for digitally modelling sketches. This modality also prevents the generation of blurry sketches like in the case of raster image generation [Radford et al., 2015].

3.1.1.2 Generating vector graphics using deep learning

In recent years, automatic vector generation has advanced rapidly and deep learning-based approaches have been proposed [Mo et al., 2021; Smirnov et al., 2020; Li et al., 2020; Huang et al., 2019; Bessmeltsev and Solomon, 2019; Guo et al., 2019; Kim et al., 2018]. There is a body of recent literature describing models that operate purely on vector stroke data (that is, the process of actually drawing the vectors into an image is not part of the learning machinery). This includes recurrent generative models utilising VAEs for sketch data [*e.g.* Ha and Eck, 2018; Lopes et al., 2019], generative models utilising GANs for sketch generation in vector format [*e.g.* Balasubramanian et al., 2019; Azadi et al., 2018], few-shot learning [*e.g.* Azadi et al., 2018; Gao et al., 2019] and reinforcement learning [*e.g.* Zhou et al., 2018; Xie et al., 2013; Ganin et al., 2018]. Concurrent to our method presented in this chapter is the approach of Das et al. [2021] which proposed a generative model for training and inferring from point-cloud data to generate parametric sketches. Another line of work within sketch generation uses Bayesian Program Learning, rather than deep networks, to represent the act of drawing as a probabilistic generative model [Lake et al., 2015].

3.1.2 Vector graphics rasterisation

An important line of research within sketch generation is the process of rasterisation. Previous work in this area focused on the efficiency of algorithms and other traits like anti-aliasing or parallelisation of the rasterisation process [Batra et al., 2015; Manson and Schaefer, 2013; Duff, 1989; Fabris and Forrest, 1997; Kilgard and Bolz, 2012]. Until very recently, the process of rasterisation and rendering was thought to be non-differentiable, so two approaches were used to circumvent this problem. Firstly, there were models that use reinforcement learning to learn drawing actions through a traditional (non-differentiable) renderer [*e.g.* Ganin et al., 2018; Mellor et al., 2019]. and, secondly, there were approaches that 'learn' renderers (typically formulated as networks of transposed convolutions, or convolutions and upsampling operations) that take vector inputs and produce raster outputs [Zheng et al., 2019; Zou et al., 2020; Nakano, 2019; Huang et al., 2019]. Of the latter, the work by Zheng et al. [2019] is most similar to ours in its intent to explore sketches and to utilise encoder models to produce accurate stroke parameters from raster images. Our models in Section 3.4 are however fully end-to-end learnable, unlike Zheng et al.'s model in which the renderer network is trained separately. Models with learned rasterisers are also inherently inflexible in the sense that they have to be trained for every type of stroke parameterisation they can work with.

3.1.3 Differentiable rendering

Unlike neural approximations of rasterisation, a differentiable rasteriser allows for a potentially more principled or flexible approach to modelling the drawing process or the loss that is optimised. A number of models have been proposed that incorporate drawing into learning machinery.

Recent approaches to differentiable 3D rendering have garnered attention in the computer vision community [*e.g.* Liu et al., 2019; Kato et al., 2018], and indeed it is the work of Liu et al. [2019] that originally helped inform the approach we detail in Section 3.2. During the development of our approach, Li et al. [2020] presented a differentiable relaxation that takes advantage of how anti-aliasing is performed in modern computer graphics systems using multi-sampling, by providing differentiable relaxations. We consider this to be a top-down approach to the problem because it does not change the underlying rendering model. Conversely, we consider our approach to be bottom-up because we explicitly allow the rendering model to be flexibly defined in a way that is appropriate to the task. Building upon the differentiable rasteriser of Li et al. [2020], Reddy et al. [2021] proposed a neural network model that synthesis vector graphics using only raster-based supervision, much like the application we show in Section 3.4. Their model, however, is computationally more complex in an attempt to model vector graphics as closed-form Bézier curves shaped by deformations of the unit circle. Our approach, on the other hand, allows modelling complex compositions of shapes with various primitives and its effectiveness is demonstrated on several datasets without requiring vector supervision.

Other recent works (largely developed and published after our work) proposed methods for sketch generation using differentiable rendering. Das et al. [2021], which extends Das et al. [2020], explores a model for vector graphics sketch generation using variable degree Bézier curve fitting. This approach offers flexibility in choosing the degree of complexity for each curve, unlike all parametric curves having the same degree as in the previous version of Das et al. [2020]. However, while the model of Das et al. [2020] operates only on cloud point data, the improved version first requires a transformer-based encoder to parse raster datasets as ink-point clouds [Das et al., 2021] and the parametrisation is limited to only Bézier curves.

Another similar piece of work, but considerably different to ours, is that of Smirnov et al. who predicts parametric shape primitives using distance fields. Concretely, Smirnov et al. [2020] defines a general loss function based on a version of Chamfer distance and, along with two other specialised losses, maps 2D raster inputs to quadratic Bézier curves and 3D distance fields to cuboids. Similarly to our approach, the method proposed in

Smirnov et al. [2020] analytically computes distance fields to primitives such as Bézier curves. Although this approach does not specifically target differentiable rasterisation, the generalised loss is differentiable with respect to its parameters which allows for backpropagation-based learning. However, this approach can only output predefined topologies enforced by a class-dependent template loss. Moreover, our framework allows flexibility in choosing the primitive parametrisation and does not enforce any output topology.

Mo et al. [2021] also combine a differentiable rendering approach with an RNN to train on raster data only. Their framework targets a wide range of images and does so by modelling a virtual pen that zooms in and out on the raster image to generate sequential strokes. The whole process involves 4 main stages: cropping, stroke generation, stroke rendering and pasting. The differentiable renderer is based on Huang et al. [2019]'s method and is a neural network that approximates the stroke image given the learned parameters of a quadratic Bèzier curve. Similar to our approach, this enables the use of a raster-based only loss. Again, our approach goes beyond only Bèzier parametrisation as discussed in the next section.

3.2 Differentiable Relaxations of Rasterisation

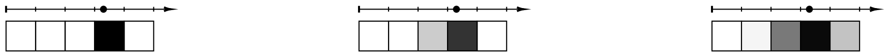
In this section, we discuss the problem of drawing, or rasterising points, lines and curves defined in a continuous world space $\mathcal{W}$ into an image space $\mathcal{I}$. Our objective is to present a formalisation that allows us to ultimately define rasterisation functions that are differentiable with respect to their world space parameters (*e.g.* the (co)ordinate of a point, or (co)ordinates of the beginning and end of a line segment).

3.2.1 1D Rasterisation

We first consider the problem of rasterising a one-dimensional point $p \in \mathcal{W}$ where $\mathcal{W} = \mathbb{R}$. Concretely, the process of rasterisation of the point p can be defined by a function, $f(n;p)$, that computes a value (typically $[0,1]$) for every pixel in the image space $\mathcal{I}$, whose position is given by $n \in \mathcal{I}$. Such a function represents a scalar field over the space of possible values of n. Commonly we consider values of n to be non-negative integers from the lattice or grid, $\mathbb{Z}^1_{0+}$, defining a pixel in the image.

3.2.1.1 Simple closest-pixel rasterisation functions

If we assume that the 0th pixel covers the domain $[0,1)$ in the world space of a point p, and that the 1st pixel covers $[1,2)$, etc. Nearest-neighbour rasterisation then maps the



(A) Nearest-neighbour rasterisation: the closest pixel is shaded by the by flooring the point ordinate. There is no useful gradient information.

(B) Anti-aliased rasterisation: the closest two pixels are shaded proportionally by to the distance from the point to those two pixels.

(C) Rasterisation using Equation 3.3 with $\sigma^2 = 1$. Every pixel in the image will have a (small) gradient with respect to the point ordinate.

FIGURE 3.2: **Different point rasterisation functions illustrated in one-dimension.**

real-valued point, p, to an image by rounding down:

$$f(n;p) = \begin{cases} 1 & \text{if } \lfloor p \rfloor = n \\ 0 & \text{otherwise} . \end{cases} \tag{3.1}$$

This process is illustrated in Figure 3.2a. An alternative rasterisation scheme, illustrated in Figure 3.2b is to interpolate over the two closest pixels. Assuming that a pixel has maximal value when the point being rasterised lies at its midpoint, then:

$$f(n;p) = \begin{cases} 1.5 - p + \lfloor p - 0.5 \rfloor & \text{if } \lfloor p - 0.5 \rfloor = n \\ 0.5 + p - \lceil p - 0.5 \rceil & \text{if } \lceil p - 0.5 \rceil = n \\ 0 & \text{otherwise} . \end{cases} \tag{3.2}$$

These functions (extended to 2D) are actually implicitly used in many computer graphics systems, but rarely in the form we have written them. Most graphics subroutines approach the rasterisation problem from the perspective of directly determining which pixels in n should have a colour associated with them given p as this is more efficient if the objective is just to draw the primitive p.

3.2.1.2 Differentiable relaxations

Ideally, we would like to be able to define a rasterisation function that is differentiable with respect to p. This would allow p to be optimised with respect to some objective. The rasterisation function given by Equation 3.1 is piecewise differentiable with respect to p, but the gradient is zero almost everywhere which is not useful. Although Equation 3.2 has some gradient in the two pixels nearest to p, overall it has the same key problem: the gradient is zero almost everywhere.

We would like to define a rasterisation function that has gradient for all (or at least a large proportion of) possible values of n. This function should be continuous and differentiable almost everywhere. The anti-aliased rasterisation approach gives some hint as to how this could be achieved: the function could compute a value for every n based on the distance between n and p. Distance metrics have an infinite upper bound, whereas we want our pixel values to be finitely bounded in $[0, 1]$, so inversion and application of a non-linearity are necessary. The properties of the chosen function should give values close to 1 when n and p are *close*, and values near 0 when they are *far apart*.

An obvious choice of non-linearity would be to exponentiate the negative squared distances, and use a scaling factor σ^2 to control the fuzziness of the rasterisation and the size of the point or width of the line stroke (see Figure 3.2c):

$$f(n;p) = \exp\left(\frac{-d^2(n, p-0.5)}{\sigma^2}\right) . \tag{3.3}$$

It can be shown that there is a direct linear relationship (see proof in Section 3.2.1.4) between the size of a point or thickness of a line, t, and the value of σ: $\sigma \approx 0.54925t \,\forall\, t > 0$.

3.2.1.3 Relaxed rasterisation in N-dimensions

All of the 1D rasterisation functions previously defined can be trivially extended to rasterise a *point* in two or more dimensions. For example, if the point $\boldsymbol{p}$ was considered to be a vector in the world space $\mathcal{W} = \mathbb{R}^2$ and correspondingly $\boldsymbol{n}$ was a vector in the image space[1], $\mathcal{I} = \mathbb{Z}^2_{0+}$, and the floor and ceiling operators are applied element-wise then all three 1D rasterisation functions hold in two (or more) dimensions.

3.2.1.4 Relating σ to point size and line thickness

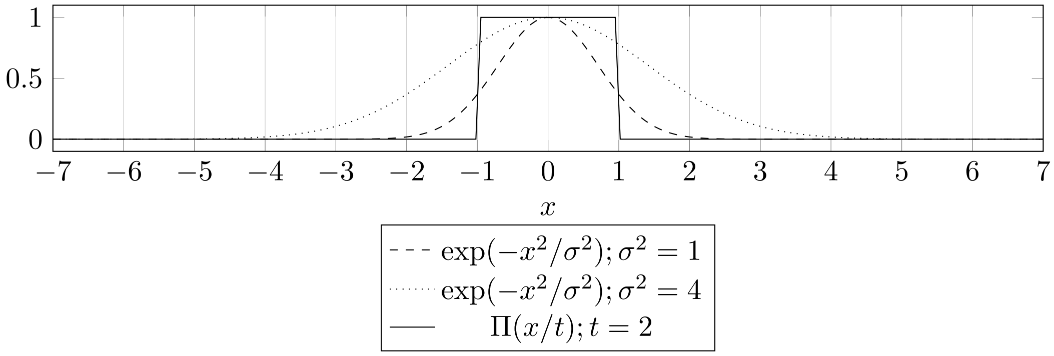


FIGURE 3.3: **Illustration of a target point of thickness $t = 2$ pixels and approximations with the *exp* rasterisation function with different σ^2 values.**

Consider the 1D rasterisation of a point of size t given by the scaled unit box function $\Pi(x/t)$ and the relaxed rasterisation given by $\exp(-d^2(x)/\sigma^2)$ as illustrated in Figure 3.3. We want to find a relationship between the value of t and σ^2 when trying to minimise

[1] Note that it is most common to use positive integers to index pixels in the image space, but this isn't a requirement; the image space could be unbounded or real for example.

the squared difference of the functions across the entire domain x,

$$\begin{aligned} \min_{\sigma^2} \quad & \int_{-\infty}^{\infty} (e^{-x^2/\sigma^2} - \Pi(x/t))^2 \, dx \\ \text{s.t.} \quad & t > 0 \\ & \sigma^2 > 0 \, . \end{aligned} \tag{3.4}$$

The integral term can be expanded and evaluated as follows (assuming the constraints $t > 0$ and $\sigma^2 > 0$):

$$\begin{aligned} & \int_{-\infty}^{\infty} (e^{-x^2/\sigma^2} - \Pi(x/t))^2 \, dx \\ &= \int_{-\infty}^{\infty} e^{-2x^2/\sigma^2} - 2\Pi(x/t)e^{-x^2/\sigma^2} + \Pi(x/t)^2 \, dx \\ &= \sigma\sqrt{\frac{\pi}{2}} - 2\sigma\sqrt{\pi}\,\mathrm{erf}\left(\frac{t}{2\sigma}\right) + t \, . \end{aligned} \tag{3.5}$$

Now, differentiating and setting to zero gives

$$\begin{aligned} 0 &= \frac{d\left(\sigma\sqrt{\frac{\pi}{2}} - 2\sigma\sqrt{\pi}\,\mathrm{erf}\left(\frac{t}{2\sigma}\right) + t\right)}{d\sigma} \\ &= \sqrt{\frac{\pi}{2}} - 2\sqrt{\pi}\frac{d\left(\sigma\,\mathrm{erf}\left(\frac{t}{2\sigma}\right)\right)}{d\sigma} \\ &= \sqrt{\frac{\pi}{2}} - 2\sqrt{\pi}\left(\mathrm{erf}\left(\frac{t}{2\sigma}\right) + \sigma\frac{d\left(\mathrm{erf}\left(\frac{t}{2\sigma}\right)\right)}{d\sigma}\right) \\ &= \sqrt{\frac{\pi}{2}} - 2\sqrt{\pi}\,\mathrm{erf}\left(\frac{t}{2\sigma}\right) - 2\sqrt{\pi}\sigma\left(-\frac{te^{-t^2/(4\sigma^2)}}{\sigma^2\sqrt{\pi}}\right) \\ &= \sqrt{\frac{\pi}{2}} - 2\sqrt{\pi}\,\mathrm{erf}\left(\frac{t}{2\sigma}\right) + \frac{2te^{-t^2/(4\sigma^2)}}{\sigma} \, . \end{aligned} \tag{3.6}$$

Noting the common factors of t/σ in Equation 3.6 we can write the right hand side as an expression in terms of $c = t/\sigma$:

$$\sqrt{\frac{\pi}{2}} - 2\sqrt{\pi}\,\mathrm{erf}\left(\frac{c}{2}\right) + 2ce^{-c^2/4} \, . \tag{3.7}$$

As shown in Figure 3.4 this expression is monotonically decreasing and has a single root, which can be estimated numerically as $c \approx 1.820657$.

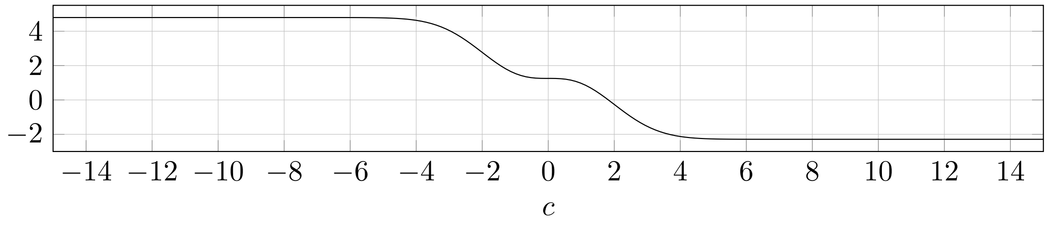


FIGURE 3.4: **Plot of Equation 3.7.**

This implies the relationship between t and σ is linear: $\sigma \approx 0.54925t \, \forall \, t > 0$. This can be easily verified by substituting $\sigma = 0.54925t$ in Equation 3.6.

3.2.2 Line segments

A *line segment* can be defined by its start coordinate $\boldsymbol{s} = [s_x, s_y]$ and end coordinate $\boldsymbol{e} = [e_x, e_y]$. The normal approaches to rasterising lines in computer graphics [*e.g.* Bresenham, 1965; Wu, 1991] are highly optimised and work by considering just the pixels that intersect the line or are within a few pixels of it. These algorithms typically iterate over the line, setting the underlying pixels values accordingly. To develop a general set of (potentially differentiable) rasterisation functions we need to consider a formalisation of rasterisation as we did in the 1D case where we consider a function that defines a scalar field over the set of all pixel positions, n, in the image given a particular line segment: $f(\boldsymbol{n}; \boldsymbol{s}, \boldsymbol{e})$.

To rasterise a line segment one needs to consider how close a pixel is to the segment. We can efficiently compute the squared Euclidean distance of an arbitrary pixel $\boldsymbol{n}$ to the closest point on the line segment as follows:

$$
\begin{aligned}
\boldsymbol{m} &= \boldsymbol{e} - \boldsymbol{s} \;, \\
t &= \frac{((\boldsymbol{n} - \boldsymbol{s}) \cdot \boldsymbol{m})}{(\boldsymbol{m} \cdot \boldsymbol{m})} \;, \\
\mathrm{d}^2_{\mathrm{seg}}(\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{e}) &= \begin{cases} \|\boldsymbol{n} - \boldsymbol{s}\|_2^2 & \text{if } t \leq 0 \\ \|\boldsymbol{n} - (\boldsymbol{s} + t\boldsymbol{m})\|_2^2 & \text{if } 0 < t < 1 \\ \|\boldsymbol{n} - \boldsymbol{e}\|_2^2 & \text{if } t \geq 1 \;. \end{cases}
\end{aligned}
\tag{3.8}
$$

Concretely, $\mathrm{d}^2_{\mathrm{seg}}(\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{e})$ is the squared Euclidean Distance Transform of the line segment (illustrated in Figure 3.5). It defines a scalar field in which the value is equal to the squared distance to the closest point on the line segment. This function is piecewise smooth and differentiable with respect to the line segment parameters everywhere for a given $\boldsymbol{n}$.

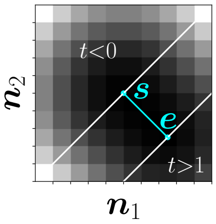


FIGURE 3.5: **Squared Euclidean distance of an arbitrary pixel n to the closest point on the line segment.**

In the case of nearest-neighbour rasterisation (shown in Figure 3.6) one would ask if the line passes through the pixel in question and only fill it if that were the case:

$$f(\boldsymbol{n};\boldsymbol{s},\boldsymbol{e}) = \begin{cases} 1 & \text{if } \mathrm{d}^2_{\mathrm{seg}}(\boldsymbol{n},\boldsymbol{s},\boldsymbol{e}) \leq \delta^2 \\ 0 & \text{otherwise} . \end{cases} \tag{3.9}$$

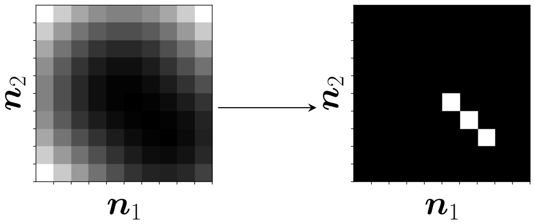


FIGURE 3.6: **Nearest-neighbour rasterisation of a line segment.**

Assuming a 1-1 mapping between the domains of the coordinate system of the image space and world space, then $\delta^2 = 0.5$ would give a rasterisation that mimics the 1-pixel wide line that would be drawn by Bresenham's algorithm [Bresenham, 1965]. If we replace the calculation of distance to a point in Equation 3.3 with the minimum distance to the line segment we get a line segment rasteriser (see Figure 3.7) that is differentiable with respect to the parameters of the line segment $\boldsymbol{s}$ and $\boldsymbol{e}$:

$$f(\boldsymbol{n};\boldsymbol{s},\boldsymbol{e}) = \exp\left(\frac{-\,\mathrm{d}^2_{\mathrm{seg}}(\boldsymbol{n},\boldsymbol{s},\boldsymbol{e})}{\sigma^2}\right) . \tag{3.10}$$

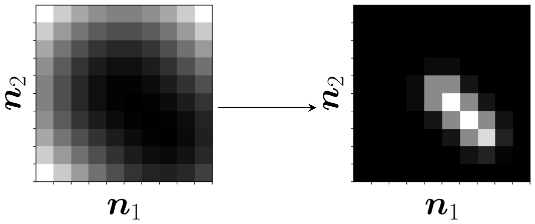


FIGURE 3.7: **Differentiable rasterisation of a line segment with respect to its start and end parameters.**

3.2.3 Curves

It is common in computer graphics to utilise parametric curves $C(t, \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ defines the parameters and $0 \leq t \leq 1$. Typically $C(t, \boldsymbol{\theta})$ is polynomial (usually quadratic or cubic in t). The parameters $\boldsymbol{\theta}$ are commonly specified in Bézier (*e.g.* Bézier Curves) or Hermite form (*e.g.* Catmull-Rom splines) as described in Section 3.2.3.1. To rasterise a curve (irrespective of the parameterisation) in a way that is differentiable with respect to the parameters we can follow the same general approach that was taken for line segments — *compute the minimum Squared Euclidean distance between each coordinate* $\boldsymbol{n} \in \mathcal{I}$ *and the curve*,

$$\begin{aligned} \mathrm{d}^2_{\mathrm{cur}}(\boldsymbol{n}, \boldsymbol{\theta}) = \min_t \quad & \|C(t, \boldsymbol{\theta}) - \boldsymbol{n}\|_2^2 \\ \text{s.t.} \quad & 0 \leq t \leq 1 \, . \end{aligned} \tag{3.11}$$

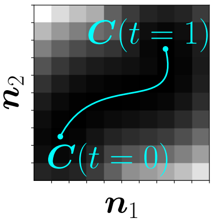


FIGURE 3.8: **Squared Euclidean distance of an arbitrary pixel n to the closest point on the curve.**

This is illustrated in Figure 3.8. As in the case of line segments, this distance transform can then be combined with a rasterisation function that works in terms of a distance (see also Figure 3.9):

$$f(\boldsymbol{n}; \boldsymbol{\theta}) = \exp\left(\frac{-\,\mathrm{d}^2_{\mathrm{cur}}(\boldsymbol{n}, \boldsymbol{\theta})}{\sigma^2}\right) \, . \tag{3.12}$$

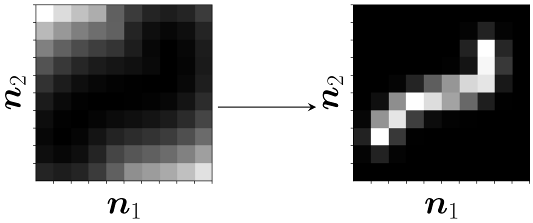


FIGURE 3.9: **Differentiable rasterisation function of a curve segment with respect to its parameters.**

The only additional challenge over the rasterisation of line segments is that the computation of the distance map requires solving a constrained minimisation problem and doesn't have a closed-form solution. A number of different approaches are possible (see Section 3.2.3.2), however, in practice, we have had success with both a fast polyline approximation[2] and a recursive approach which are both easily vectorised as tensor operations that can be performed efficiently on a GPU.

[2] Note that there is a potential for a small change in curve's parameters to cause a large difference in the polyline approximation, although we have not seen this become an issue in practice.

3.2.3.1 Curve parameterisations

Curves are often represented mathematically by parametric functions $C(t)$ that give the coordinates of the curve for values of t, commonly in the closed interval $[0, 1]$, with $t = 0$ representing the start point and $t = 1$ representing the end point of the curve. Curves are often parameterised as the coefficients of polynomial basis functions, either in Hermite (*e.g.* the curve is a linear combination of Hermite bases) or Bézier form (the curve is represented as a linear combination of Bernstein bases). The following parametric curve formulations are commonly used in computer graphics (and can all be used in our differentiable rasterisation approach):

Quadratic Bézier curves are parameterised by three points: the start of the curve $\boldsymbol{P}_0$, the end of the curve $\boldsymbol{P}_2$ and the control point $\boldsymbol{P}_1$ which is the point the tangents to the curve at $\boldsymbol{P}_0$ and $\boldsymbol{P}_2$ intersect. The curve would not normally pass through $\boldsymbol{P}_1$. The curve can be thought of as leaving $\boldsymbol{P}_0$ in the direction of $\boldsymbol{P}_1$ and gradually bending to arrive at $\boldsymbol{P}_2$ from the direction of $\boldsymbol{P}_1$. The quadratic Bézier is defined as:

$$\mathrm{C}_{\mathrm{bez}^2}(t, \boldsymbol{\theta}) = (1-t)^2\boldsymbol{P}_0 + 2(1-t)t\boldsymbol{P}_1 + t^2\boldsymbol{P}_2 \tag{3.13}$$

where

$$\boldsymbol{\theta} = [\boldsymbol{P}_0|\boldsymbol{P}_1|\boldsymbol{P}_2]\,.$$

Cubic Bézier curves are defined by four points: $\boldsymbol{P}_0$ is the start of the curve; $\boldsymbol{P}_1$ is the first control point and indicates the direction the curve leaves $\boldsymbol{P}_0$ from; $\boldsymbol{P}_2$ is the second control point and indicates the direction that the curve arrives at the final end point $\boldsymbol{P}_3$ from. The curve would not normally pass through either control point. The cubic Bézier is defined as:

$$\begin{aligned}\mathrm{C}_{\mathrm{bez}^3}(t, \boldsymbol{\theta}) =& (1-t)^3\boldsymbol{P}_0 + 3(1-t)^2t\boldsymbol{P}_1 \\ &+ 3(1-t)t^2\boldsymbol{P}_2 + t^3\boldsymbol{P}_3\end{aligned} \tag{3.14}$$

where

$$\boldsymbol{\theta} = [\boldsymbol{P}_0|\boldsymbol{P}_1|\boldsymbol{P}_2|\boldsymbol{P}_3]\,.$$

Catmull-Rom splines parameterise a curve by 4 points which the curve passes through smoothly. The curve is only drawn between the middle pair of points:

$$\mathrm{C}_{\mathrm{crs}}(t, \boldsymbol{\theta}) = \frac{t_2 - t}{t_2 - t_1}\boldsymbol{B}_1 + \frac{t - t_1}{t_2 - t_1}\boldsymbol{B}_2 \tag{3.15}$$

where

$$\begin{aligned}\boldsymbol{B}_1 &= \frac{t_2 - t}{t_2 - t_0}\boldsymbol{A}_1 + \frac{t - t_0}{t_2 - t_0}\boldsymbol{A}_2 \\ \boldsymbol{B}_2 &= \frac{t_3 - t}{t_3 - t_1}\boldsymbol{A}_2 + \frac{t - t_1}{t_3 - t_1}\boldsymbol{A}_3\end{aligned}$$

$$
\begin{aligned}
\boldsymbol{A}_1 &= \frac{t_1 - t}{t_1 - t_0}\boldsymbol{P}_0 + \frac{t - t_0}{t_1 - t_0}\boldsymbol{P}_1 \\
\boldsymbol{A}_2 &= \frac{t_2 - t}{t_2 - t_1}\boldsymbol{P}_1 + \frac{t - t_1}{t_2 - t_1}\boldsymbol{P}_2 \\
\boldsymbol{A}_3 &= \frac{t_3 - t}{t_3 - t_2}\boldsymbol{P}_2 + \frac{t - t_2}{t_3 - t_2}\boldsymbol{P}_3 \\
t_0 &= 0 \\
t_{i+1} &= \|\boldsymbol{P}_{i+1} - \boldsymbol{P}_i\|_2^\alpha + t_i
\end{aligned}
$$

and

$$\boldsymbol{\theta} = [\boldsymbol{P}_0|\boldsymbol{P}_1|\boldsymbol{P}_2|\boldsymbol{P}_3] \,.$$

The *centripetal* Catmull-Rom spline sets α to 0.5, which has the advantage that cusps or self-intersections cannot be formed in the curve.

3.2.3.2 Computing the squared Euclidean distance transform for a curve

In general, it is not possible to write a closed-form expression for the (squared) distance of an arbitrary point, $\boldsymbol{n}$ to the closest point on a curve, $C(t)$,

$$
\begin{aligned}
\mathrm{d}^2_{\mathrm{cur}}(\boldsymbol{n}) = \min_t \quad & \|C(t) - \boldsymbol{n}\|_2^2 \\
\text{s.t.} \quad & 0 \leq t \leq 1 \,.
\end{aligned}
\tag{3.16}
$$

Algorithm 1 Polyline approximation for the closest point on a curve. This approximation breaks the curve into *segments* uniform-Δt line segments, and might be sub-optimal in areas of high curvature (if such areas were to exist, then an adaptive variant of this algorithm could instead be used).

```
Function MinDistanceToCurvePolyline($\boldsymbol{\theta}$, $C$, $\boldsymbol{n}$, $segments$)
    Data:
        $\boldsymbol{\theta}$: curve parameters.
        $C$: function defining coordinates of curve at a distance $0 \leq t \leq 1$ along it.
        $\boldsymbol{n}$: coordinate to compute distance from.
        $segments$: number of line segments to use in the approximation.
    Result: the square of the minimum distance between $\boldsymbol{n}$ and the curve.

    $mindist \leftarrow \infty$
    for ( $i = 1$; $i \leq segments$ ; $i = i + 1$ ) {
        $t_0 \leftarrow (i - 1) \;/\; segments$
         $t_1 \leftarrow (i) \;/\; segments$

        $dist \leftarrow \mathrm{d}^2_{\mathrm{seg}}(\boldsymbol{n}, C(t_0, \boldsymbol{\theta}), C(t_1, \boldsymbol{\theta}))$ // See Equation 3.8
        if $dist < mindist$ then
            $mindist \leftarrow dist$

    return $mindist$
```

Algorithm 2 Recursive brute-force search for the closest point on a curve. This is approximate in the sense that if *slices* is too small the wrong minima might be located, and that *iters* controls the precision of the solution that is found.

```
Function MinDistanceToCurveBruteForce($\boldsymbol{\theta}$, $C$, $\boldsymbol{n}$, $t_{min}$, $t_{max}$, $iters$, $slices$, $mindist{=}\infty$)
    Data:
      $\boldsymbol{\theta}$: curve parameters.
      $C$: function defining coordinates of curve at a distance $0 \leq t \leq 1$ along it.
      $\boldsymbol{n}$: coordinate to compute distance from.
      $t_{min}$: starting value of $t$ for the search.
      $t_{max}$: ending value of $t$ for the search.
      $iters$: number of iterations to perform.
      $slices$: number of intervals between $t_{min}$ and $t_{max}$ to compute the distance at.
      $mindist$: current minimum distance estimate.
    Result: the square of the minimum distance between $\boldsymbol{n}$ and the curve.
    if $iters \leq 0$ then
        return $mindist$
    $\Delta_t \leftarrow (t_{max} - t_{min})$ / $slices$
    $t \leftarrow t_{min}$
    $t_{best} \leftarrow t_{min}$

    repeat
        $dist \leftarrow \|C(t, \boldsymbol{\theta}) - \boldsymbol{n}\|_2^2$
        if $dist < mindist$ then
            $mindist \leftarrow dist$
            $t_{best} \leftarrow t$
        $t \leftarrow t + \Delta_t$
    until $t \geq t_{max}$;
    return MinDistanceToCurveBruteForce($\boldsymbol{\theta}$, $C$, $\boldsymbol{n}$, $t_{best} - \Delta_t$, $t_{best} + \Delta_t$, $iters{-}1$, $slices$, $mindist$)
```

Potential approaches to computing this would for example be through a polyline approximation (see Algorithm 1), a recursive brute force search (see Algorithm 2) or a method based on finding the roots of the polynomial given by the derivative

$$\frac{d}{dt}\|C(t) - \boldsymbol{n}\|_2^2 \, . \tag{3.17}$$

In the latter case, the root-finding itself could be achieved in several ways; for example, by computing the real eigenvalues of the companion matrix formed from Equation 3.17 that lie between 0 and 1 and selecting the one that gives minimum distance, or by locating two values of t that give opposing signs of Equation 3.17 and applying the bisection method. Another potential alternative is the method proposed by Li et al. [2020] which uses bisection with the Newton-Raphson method, with the initial guess computed using isolator polynomials [Sederberg and Chang, 1994].

The challenge of all the latter approaches is efficient vectorised batch implementation, whereby computation of distance transforms (the computation of the minimum distance to a curve for all points in the image space) is performed for a *batch* of curves in parallel making efficient use of many-core hardware. An approach based on root finding using the real eigenvalues of the companion matrix, for example, should ultimately prove to be more accurate than a polyline approximation, and potentially better and faster than the brute-force search, however at the time of writing there are not any hardware optimised batch generalised eigenvalue decomposition (GEVD) implementations available; a batch GEVD implementation (for small matrices) is necessary as the decomposition would have to be computed for every pixel $\boldsymbol{n} \in \mathcal{I}$.

Currently, we have proof-of-concept implementations using the former polyline approximation and brute force approaches, and these are both vectorised to run on many-core (particularly GPU) hardware. The polyline approximation is in general faster (obviously both the polyline and brute force approaches allow the degree of precision to be adjusted, and that changes the computational complexity), but it does have a potential disadvantage that the approximation can introduce degeneracies whereby a small change in a curve's parameters cause a topological change in the polyline approximation. In practice, however, we have not found this to be a problem in all our experiments with handwritten characters, which all use a 10-segment polyline approximation for each curve segment that is drawn.

3.2.4 Composing multiple primitives

To rasterise multiple lines[3] we can consider combining the rasterisations of different line segments into a single image. We denote images produced by rasterising different line segments $\{\boldsymbol{s_1}, \boldsymbol{e_1}\}, \{\boldsymbol{s_2}, \boldsymbol{e_2}\}, \dots, \{\boldsymbol{s_i}, \boldsymbol{e_i}\}$ into matrices $\boldsymbol{I}^{(1)}, \boldsymbol{I}^{(2)}, \dots, \boldsymbol{I}^{(n)}$ defined over the same image space $\mathcal{I}$. In the simplest case, where we have binary rasterisations, we might consider that the logical-or of corresponding pixels would produce the desired effect of selecting any pixels that were shaded in the individual rasterisations as being shaded in the final output:

$$c(\boldsymbol{I}^{(1)}, \boldsymbol{I}^{(2)}, \dots, \boldsymbol{I}^{(n)}) = \boldsymbol{I}^{(1)} \vee \boldsymbol{I}^{(2)} \vee \dots \vee \boldsymbol{I}^{(n)} \, . \tag{3.18}$$

[3] We're considering composing multiple line segments, but everything here also applies to multiple points and curves, as well as combinations of line segments, points and curves, or indeed any other raster.

3.2.4.1 The *soft-or* composition operator

We can relax this composition to be differentiable and also allow the pixel values to be non binary (but restricted to $[0, 1]$) as follows:

$$\mathrm{c}_{\mathrm{softor}}(\boldsymbol{I}^{(1)}, \boldsymbol{I}^{(2)}, \dots, \boldsymbol{I}^{(n)}) = \mathbf{1} - \prod_{i=1}^{n} (\mathbf{1} - \boldsymbol{I}^{(i)}) \; . \tag{3.19}$$

Effectively if a pixel is 'on' in any of the individual images then this will select it as being 'on' in the output. This approach treats all the input images as a set; the output will be the same irrespective of the order they appear in. The majority of experiments in this chapter use the *soft-or* function. We might however consider alternative drawing functions that enable different effects and models of drawing and blending, to be achieved. Next, we discuss a few potential options, including the *over* operator used for our colour drawing examples. Note the focus here is on drawing opaque *colours*; compositions for colour with transparency are discussed in Section 3.2.5.3.

3.2.4.2 The *over* composition operator

The first potential alternative approach to the soft-or would be to define a composition that respects the ordering of the images and 'paints' each stroke *over* the top of the other (whilst not allowing background 0 pixels to cover already filled pixels) from the background to the foreground. Taking inspiration from Porter and Duff [1984]'s methods for alpha composition of computer graphics we could define a composition of image $\boldsymbol{A}$ painted over image $\boldsymbol{B}$ as:

$$\mathrm{c}_{\mathrm{over}}(\boldsymbol{A}, \boldsymbol{B}) = \boldsymbol{A} + \boldsymbol{B}(\mathbf{1} - \boldsymbol{A}) \; . \tag{3.20}$$

This function could then be applied recursively over a sequence of depth-ordered rasterisations to compose in the desired way:

$$\mathrm{c}_{\mathrm{over}}(\boldsymbol{I}_4, \mathrm{c}_{\mathrm{over}}(\boldsymbol{I}_3, \mathrm{c}_{\mathrm{over}}(\boldsymbol{I}^{(2)}, \boldsymbol{I}^{(1)}))) \; . \tag{3.21}$$

This type of approach does however have a significant problem in terms of implementation: because it is recursive and sequential, it is not easily vectorised and introduces a significant processing bottleneck which makes it intractable to use with large numbers of images. This problem can be circumvented by rewriting[4] Equation 3.21 as follows,

$$\mathrm{c}_{\mathrm{over}}(\boldsymbol{I}^{(1)}, \dots, \boldsymbol{I}^{(n)}) = \sum_{i=1}^{n} \boldsymbol{I}^{(i)} \odot \prod_{j=1}^{i-1} (\mathbf{1} - \boldsymbol{I}^{(j)}) \; . \tag{3.22}$$

[4] This was first noted by Sintorn and Assarsson [2009] for Porter and Duff's *over* operator with an alpha channel (see also Section 3.2.5.3).

In this form, we can see that in essence the computation required consists of the calculation of the cumulative product of a difference, a multiplication, and a summation; all of which can be efficiently vectorised. For numerical stability, the cumulative product can be computed as the exponentiated sum of the log differences,

$$\mathrm{c}_{\mathrm{over}}(\dots) = \sum_{i=1}^{n} \boldsymbol{I}^{(i)} \odot \exp\left(\sum_{j=1}^{i-1} \log(\mathbf{1} - \boldsymbol{I}^{(j)}) \right) . \tag{3.23}$$

The inner summation can easily be implemented using the `cumsum` operator built into most tensor processing libraries; note, however, that standard implementations will likely include cumulative sum up to and including the i-th image, so this must then be subtracted to give the required value. Additional care must also be taken to avoid taking the logarithm of zero; in practice adding a small epsilon value suffices.

3.2.4.3 The *max* composition operator

Another possible alternative composition would be to take the per-pixel maximum over the set of images:

$$\begin{aligned} &\mathrm{c}_{\mathrm{max}_{\mathrm{i,j}}}(\boldsymbol{I}^{(1)}, \boldsymbol{I}^{(2)}, \dots, \boldsymbol{I}^{(n)}) \\ &\quad = \max(\boldsymbol{I}^{(1)}_{i,j}, \boldsymbol{I}^{(2)}_{i,j}, \dots, \boldsymbol{I}^{(n)}_{i,j}) . \end{aligned} \tag{3.24}$$

Clearly this does not have usable gradients because of the max, however, a suitable differentiable relaxation exists with the smoothmax function,

$$\mathrm{smoothmax}(\boldsymbol{x}) = \mathrm{softmax}(\boldsymbol{x}/\tau)^{\top} \boldsymbol{x} , \tag{3.25}$$

where $\boldsymbol{x}$ is a vector of values to find the maximum of, and τ is a temperature parameter. As $\tau \to 0$, $\mathrm{smoothmax}(\boldsymbol{x}) \to \max(\boldsymbol{x})$. Equation 3.25 can be applied pixel-wise over a vector formed from the stacking of $[\boldsymbol{I}^{(1)}_{i,j}, \boldsymbol{I}^{(2)}_{i,j}, \dots, \boldsymbol{I}^{(n)}_{i,j}]$ to form a composition function:

$$\begin{aligned} &\mathrm{c}_{\mathrm{smoothmax}_{\mathrm{i,j}}}(\boldsymbol{I}^{(1)}, \boldsymbol{I}^{(2)}, \dots, \boldsymbol{I}^{(n)}) \\ &\quad = \mathrm{smoothmax}([\boldsymbol{I}^{(1)}_{i,j}, \boldsymbol{I}^{(2)}_{i,j}, \dots, \boldsymbol{I}^{(n)}_{i,j}]^{\top}) . \end{aligned} \tag{3.26}$$

3.2.5 Extended drawing

Clearly, at this point, we have all the components required to construct a basic drawing system. There are, however, a number of aspects that have not been considered, including, for example how to draw in colour. As we focus the remainder of the chapter on utilising the approach we have already described, this section briefly discusses additional extensions related to drawing and rasterisation.

3.2.5.1 Stroke width

As demonstrated in Section 3.2.1.4, there is a direct relationship between stroke thickness and the σ parameter used by the rasterisation function. In all the experimental results shown in Section 3.3 and Section 3.4, we used the same fixed σ for all strokes, although it should be immediately evident that this isn't a requirement, and that different strokes could have different σ values, and thus different thicknesses.

Going further, the σ value doesn't have to be a hyperparameter of the model; without changing anything within the rasterisation approach it is evident that one can compute gradients with respect to σ for every stroke that is drawn. As such, it is entirely possible to learn the line thickness of each stroke (either independently or together) by appropriately parameterising the model.

Real drawings sometimes exhibit a variation in stroke width along the length of a stroke; often this is a result of variations in pressure on the drawing instrument. It is possible to incorporate such variation into our drawing model by noting that our functions for both line segments and curves have a parameter $0 \leq t \leq 1$ along their length that can be used as an input to a function that produces different values of σ along the length of the line (or equivalently we can modify the distance map). Such a function could be parameterised by *e.g.* a simple neural network, and thus learned during the training or optimisation of a model.

3.2.5.2 Colour

Different shades of grey for individual strokes can be achieved by scalar multiplication of each stroke's raster with a grey value before composition (note that soft-or would no longer necessarily be appropriate, so a different composition would likely be used). The grey value could be learned or be a hyperparameter.

To rasterise full-colour strokes, the simplest approach is to replicate the image for a rasterised stroke three times in the channel dimension, and then multiply by a tuple of values corresponding to the desired red, green, and blue values. Again, the parameters can be learned, as is illustrated in Figure 3.1b, which uses the *over* composition operation (Equation 3.23).

If we want to rasterise lines along which the colour changes, we can follow the same methodology for changing stroke width and learn functions that emit colour as a function of the relative position, t, along the stroke.

3.2.5.3 Incorporating transparency

The differentiable rasterisation approach for colour described above can also be extended to deal with transparency. If we assume a pre-multiplied alpha colour model, where the the red, green and blue values of a pixel represent emission, and the alpha value represents occlusion, then we can directly use Porter and Duff [1984]'s compositing arithmetic. For example, the over operator with alpha,

$$\begin{aligned} c_o &= c_a + c_b(1 - \alpha_a) \\ \alpha_o &= \alpha_a + \alpha_b(1 - \alpha_a) \ , \end{aligned} \tag{3.27}$$

allows for models that can learn appropriate colour and transparency for each stroke drawn.

3.2.6 Advantages and limitations

The rasterisation process described in this section in principle allows gradients to flow from every pixel in the image to the parameters of a rendered primitive (note however that in practice this is not the case because of finite numeric precision). This is in contrast to the work of Li et al. [2020] where the gradients are limited by the size of the filter. The advantage of our method is that optimisation should be easier with more gradient. Computationally, our approach can be entirely implemented as batched tensor operations (this includes computation of distance transforms for all primitives), so all computation can be performed on the GPU making use of all available processing resources, and unlike Li et al. [2020]'s approach, does not involve the CPU for rendering. The disadvantage is that memory usage could be very high, particularly for batches of large images with many primitives (in our original envisaged use case of exploring simple sketching and writing this is not a problem, however). One interesting idea to explore in the future would be to utilise sparse tensors to reduce storage requirements by not storing pixels contributing to no value or gradient. Another potential criticism of our approach is that the generated images will be very slightly blurry as a result of the relaxation; again, for our envisaged use case this is not a problem, and it is always possible to use the relaxation for learning/optimisation, and then switch to a regular render for generation at inference time. Finally, we draw attention to the fact that our approach is not restricted to 2D, and can be *e.g.* directly applied to 3D data for voxel rasterisation.

3.3 Direct Optimisation of Primitive Parameters

With the machinery defined in Section 3.2 it is now possible to define a complete system that takes the parameters describing primitives and rasterises those primitives into an image. If a loss function is introduced in the image space, between the complete rasterised image and a fixed target image, it becomes possible to compute gradients with respect to the parameters of the primitives that created the rasterised image. Minimising this loss will adapt the underlying primitives to "shapes" that best fit the target image.

A commonly used 'reconstruction' loss function for images is the mean squared error between the target and the generated image. We can thus formalise the optimisation problem as,

$$\min_{\boldsymbol{\theta}} \|R(\boldsymbol{\theta}) - \boldsymbol{T}\|_2^2 \,, \tag{3.28}$$

for a target image, $\boldsymbol{T}$ and rasterisation function R defined over the same image space $\mathcal{I}$. The rasterisation function itself is defined as a composition $c(\dots)$ (see Section 3.2.4) over k primitives, themselves rasterised by primitive rasterisation functions, $f^{(i)}$ (*e.g.* Equations (3.3), (3.10) and (3.12)):

$$\begin{aligned} R(\boldsymbol{\theta}) = c(f^{(1)}(\boldsymbol{\theta}^{(1)}), f^{(2)}(\boldsymbol{\theta}^{(2)}), \dots, f^{(k)}(\boldsymbol{\theta}^{(k)})) \\ \text{where } \boldsymbol{\theta} = [\boldsymbol{\theta}^{(1)}|\boldsymbol{\theta}^{(2)}|\dots|\boldsymbol{\theta}^{(k)}] \,. \end{aligned} \tag{3.29}$$

If the rasterisation function R is differentiable with respect to $\boldsymbol{\theta}$, then the minimisation problem in Equation 3.28 can be solved using gradient descent. Note that the problem is in general non-convex, with potentially many local optima[5]; see Section 3.3.3 for more discussion. Additionally, the magnitude of gradients can become vanishingly small, which is particularly problematic with fixed-precision arithmetic; this problem can however be overcome as we demonstrate in the following sections.

3.3.1 Loss functions

MSE loss is not the only possible choice; in fact, MSE has one significant disadvantage in that if we are drawing in black and white, but optimising a grey-level image, then the loss landscape would be very flat. This is illustrated in Table 3.1 where it can be seen that both configurations of the generated image in the first input to the loss function produce exactly the same MSE value. Human vision does not suffer from the same problem; we can see that the two input images to the loss functions are clearly different. In addition, if we look from far enough away at the striped example on the second row, the image and the target would begin to look the same to us. To build this phenomenon into the loss function we can incorporate a notion of spatial scale. We utilise two such approaches: BlurMSE, a single-scale version of MSE in which the input, and optionally the target are blurred by a Gaussian filter of a predetermined standard deviation; and the SSMSE, a scale-space variant in which a scale-space (or optionally a scale pyramid) is built for both the input and target, and the loss is accumulated over all levels. Our implementation

[5] The optimisation landscape has considerable permutation symmetry. For example: the start and end of a line segment could be swapped with no change to the resultant image; if the composition is non-sequential the order of the rendered primitives could be permuted; etc.

of the scale-space follows Lowe [2004], and constructs a space with octaves defined by a doubling of the standard deviation, and a fixed number of intervals per octave. As can be seen from Table 3.1, the losses incorporating scale produce smaller values for the striped input image on the second row, compared to the half-half image on the first row, thus indicating usable gradient information.

Table 3.1: **Loss functions incorporating scale can overcome limitations of MSE and induce gradients.**

$\mathcal{L} =$	MSE	SSMSE	BlurMSE
$\mathcal{L}$ (◧ , ■)	0.25	0.42	0.13
$\mathcal{L}$ (▥ , ■)	0.25	0.25	0.02

A Gaussian scale-space alone is not necessarily a good measure of how a human perceives an image as it fundamentally only helps capture areas of light and dark, and ensures they are shaded accordingly in the generated image. Many perceptually motivated distance metrics have been proposed in the past, such as the well-known SSIM [Zhou Wang et al., 2004] and its variants. More recently, it has been shown that features from deep convolutional networks can correlate well with human perceptual judgements of image similarity, and this has motivated the development of CNN-based perceptual losses like LPIPS [Zhang et al., 2018]. Because a loss based on deep features would inherently be differentiable, we can utilise it as an objective when optimising primitive parameters that define an image.

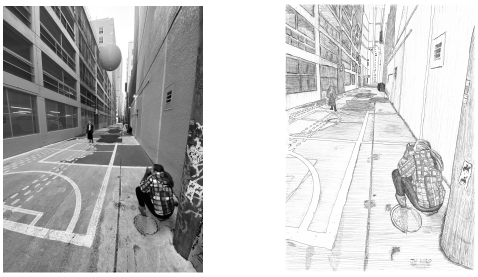
3.3.2 Examples

Next, we present examples of direct optimisation of primitive parameters.

3.3.2.1 Image-based optimisation

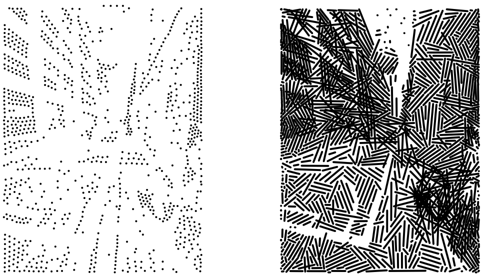
To demonstrate the effectiveness of our approach for optimising primitives against a real image we provide a number of examples. All of the generated images in Figures 3.11 and 3.12 utilise the 200 × 266 pixels input image in Figure 3.10a as the target image to optimise against. Points and pixels are optimised to have a 1-pixel diameter/thickness in the image space. The domain of the world-space is constrained to [-1,1] on the y-axis and scaled proportionally on the x-axis. All generated examples were optimised using Adam [Kingma and Ba, 2015] with a learning rate of 0.01 for 500 iterations. Figure 3.11 shows the results from optimising 1000 points and 1000 lines using blurred MSE loss and demonstrates the overall effect that can be achieved. Figure 3.12 shows the effect of optimising 500 line segments from the same starting point using a range of different losses.

It is instructive to compare how the automatically generated sketches compare to an image drawn by a human. Figure 3.10b is a hand-drawn pen and ink sketch of the same scene as used in the generation of Figures 3.11 and 3.12. It is clear that all of the sketches broadly capture the overall structure of the scene and areas of light and dark. However, there are significant differences in the way this is captured. The losses based on MSE (including scale-space and blurred) all display the same trait of capturing the local intensity, although this is much more pronounced in the scale-space and blur variants, which also capture more detail. Changing the number of intervals per octave in the scale-space losses has very little overall effect (subtle changes around the 'balloon'). The perceptual loss using AlexNet captures a highly local structure, but overall the resultant image is perhaps the least perceptually similar (or interpretable) of all the images. The



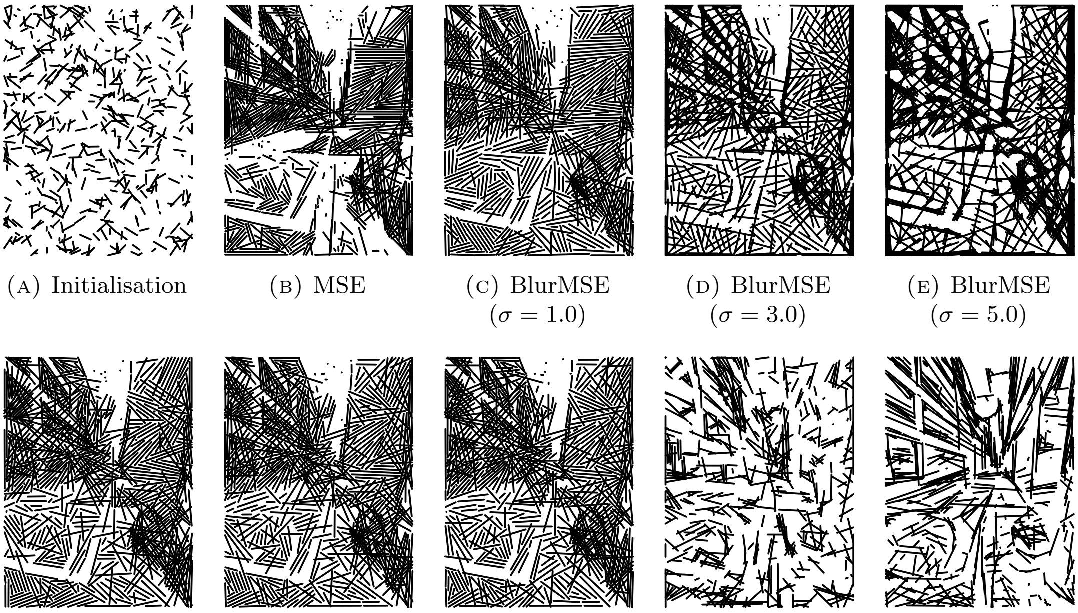
(A) Photo (B) Pen & Ink Sketch

Figure 3.10: **Pictures of DM by JH.**



(A) 1000 Points (B) 1000 lines

Figure 3.11: **Optimising against Figure 3.10a using BlurMSE ($\sigma = 1.0$).**



(A) Initialisation (B) MSE (C) BlurMSE ($\sigma = 1.0$) (D) BlurMSE ($\sigma = 3.0$) (E) BlurMSE ($\sigma = 5.0$)

(F) SSMSE, 1i/o (G) SSMSE, 2i/o (H) SSMSE, 4i/o (I) LPIPS(AlexNet) (J) LPIPS(VGG)

Figure 3.12: **Images created by optimising parameters using gradient descent with different losses.** Parameters optimised to fit the photo shown in Figure 3.10a starting from the random lines in Figure 3.12a. All images using SSMSE use 5 octaves, and 'i/o' abbreviates the number of intervals per octave. Note that regular MSE is just BlurMSE with $\sigma = 0$.

perceptual loss using VGG captures a lot of the structure of the image; it is interesting how much of the broad shape information is captured, and how areas of light and dark are also represented. In addition, we can observe that the overall brightness on the right-hand side is lighter than on the left, mimicking the human-drawn sketch, even though the raw grey-level values in the input image are similar on both sides. The differences between the two perceptual losses reflect the observation that the VGG variant is closer to traditional notions of perceptual difference when used for optimisation [Zhang et al., 2018]. Related to the observation that the VGG model seems to capture shape information rather well, we wonder if direct optimisation in the way we have performed it might lead to a new way to probe the (lack of) shape bias in different neural architectures [Geirhos et al., 2019]. This could ultimately help us move closer to networks that robustly recognise objects from both sketches and photographs.

The extended drawing operators detailed in Section 3.2.5 can be utilised directly in the optimisation approach previously described. In Figures 3.13 to 3.18 we present additional results on a wide variety of images showing the result of performing image optimisation with both the MSE loss and LPIPS(VGG) loss, together with different drawing configurations. These examples highlight again the fact that the LPIPS (VGG) loss is strikingly good at giving results that capture strong perceptual features.

3.3.2.2 Optimising cartoons

Cartoons are not well suited for the ‘direct optimisation’ setup described in this section. Whilst the rasteriser can be used for this task, there are considerable inductive biases that would be useful to incorporate to make optimisation easier. For example, incorporating strong priors for initialisation and using a more sensible loss (probably one based on Chamfer distance) would considerably reduce the speed of convergence. It might also be beneficial to work iteratively adding one curve at a time (with appropriate modifications to the loss to allow it to remain local).

However, as demonstrated in Figure 3.19, optimising randomly initialised curves (single segment Catmull-Rom splines) and their widths with the MSE loss works well for a cartoon raster to vector conversion task. The errors in these conversions are limited to missing small features — for example in the ears. The only optimisation ‘tricks’ required were to run for enough iterations (the examples in Figure 3.19 were allowed 10000 iterations, although they had converged before the end) and either using more lines than required, or using a smaller number and randomly re-initialising any that had widths learned to be zero (which become invisible) during the first half of the iteration limit. All of these details are only to overcome the fact that poorly initialised curves (ones far from any black pixel in the raster along their length) will naturally have strong gradients forcing them to be removed by reducing their stroke width to zero.

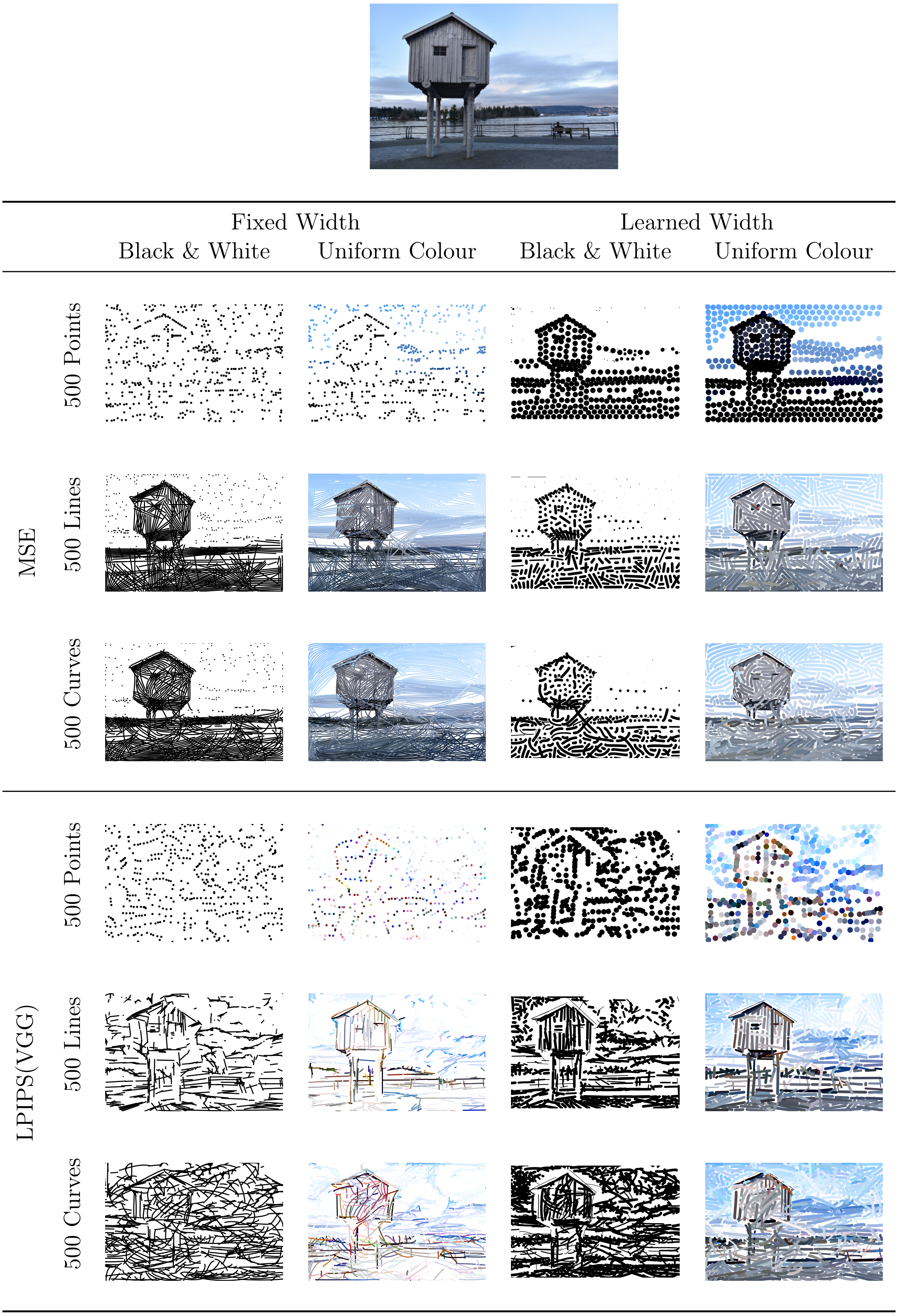


FIGURE 3.13: **Examples of image optimisation (i).**

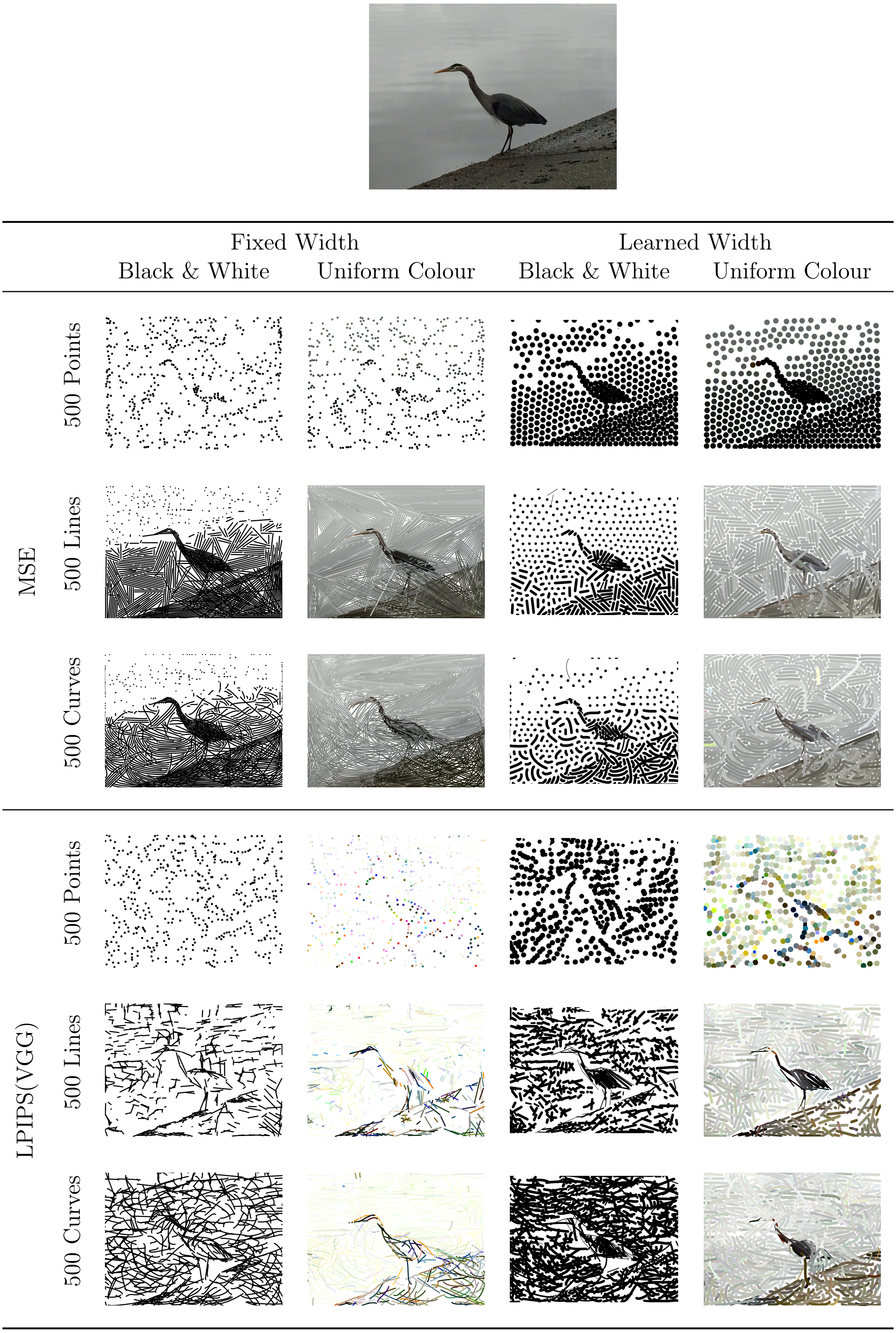


Figure 3.14: **Examples of image optimisation (ii).**

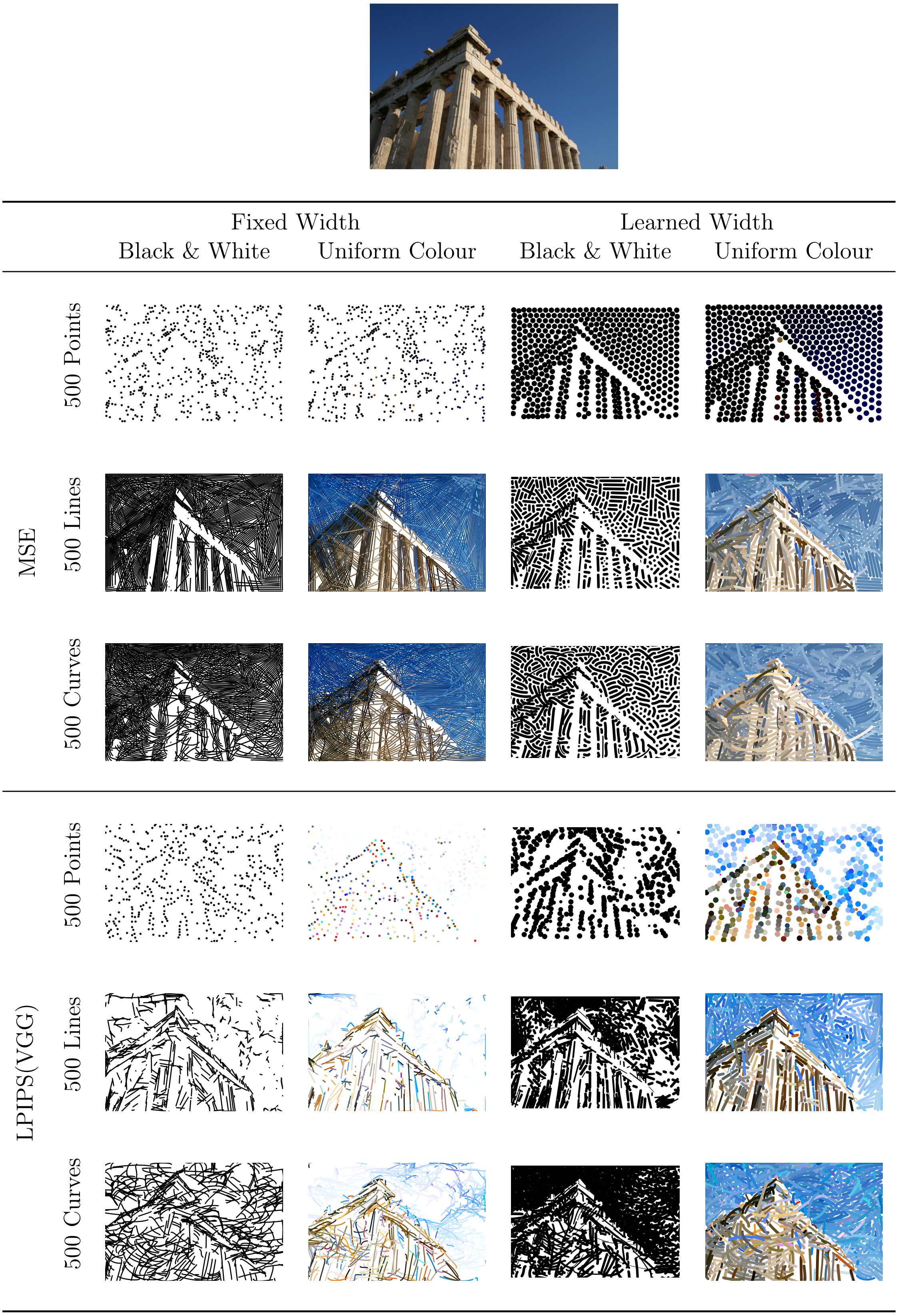


FIGURE 3.15: **Examples of image optimisation (iii).**

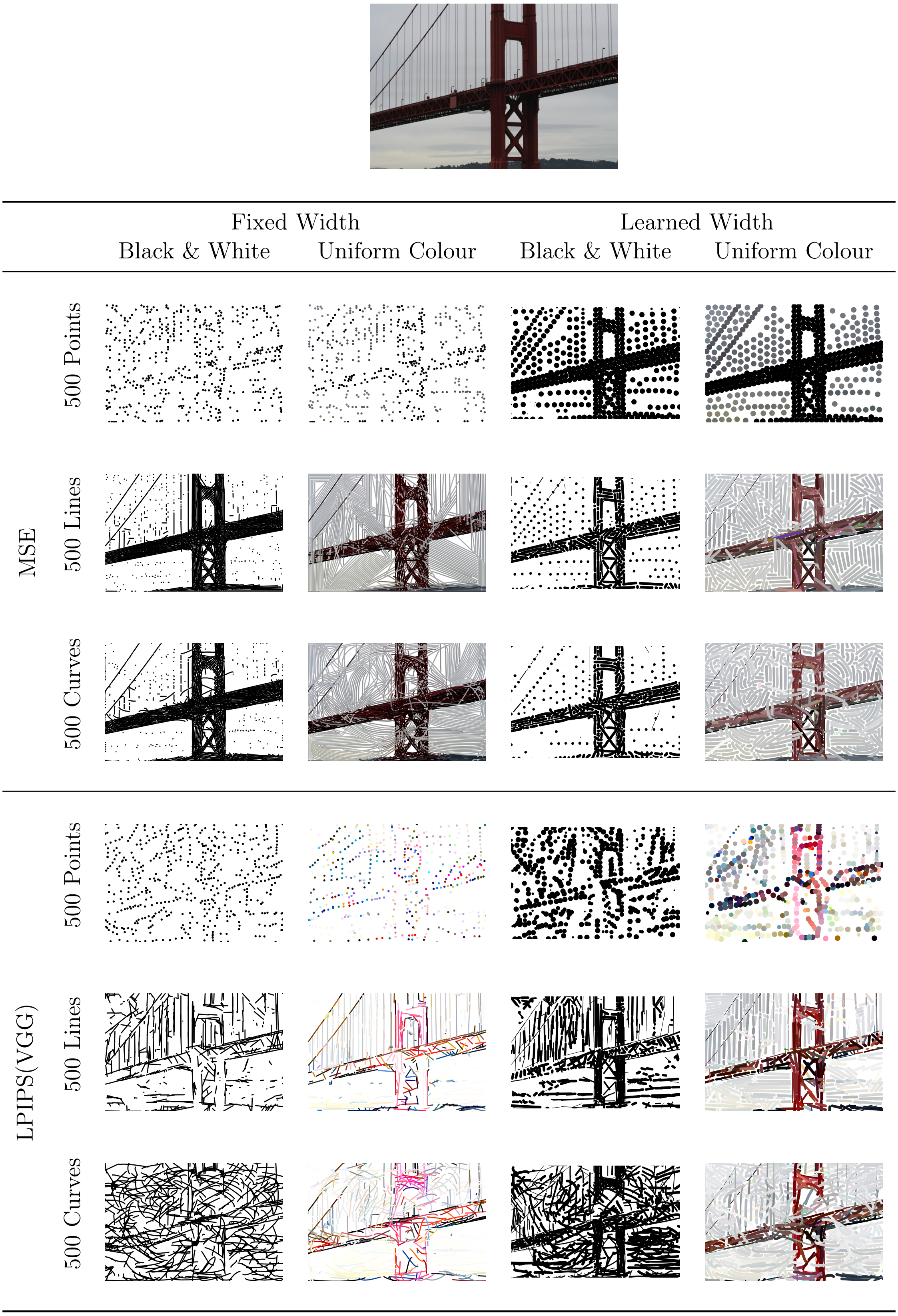


FIGURE 3.16: **Examples of image optimisation (iv).**

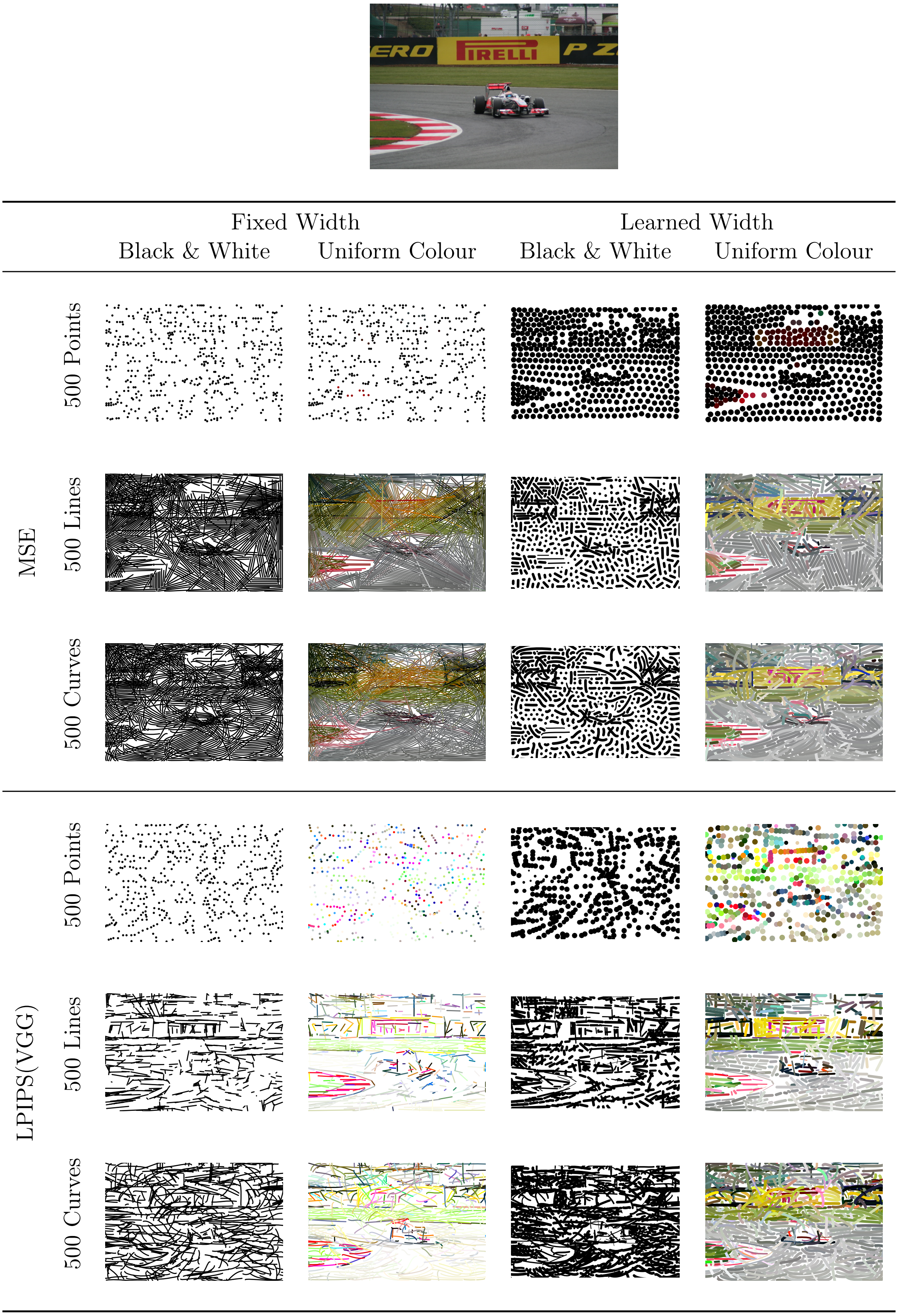


Figure 3.17: **Examples of image optimisation (v).**

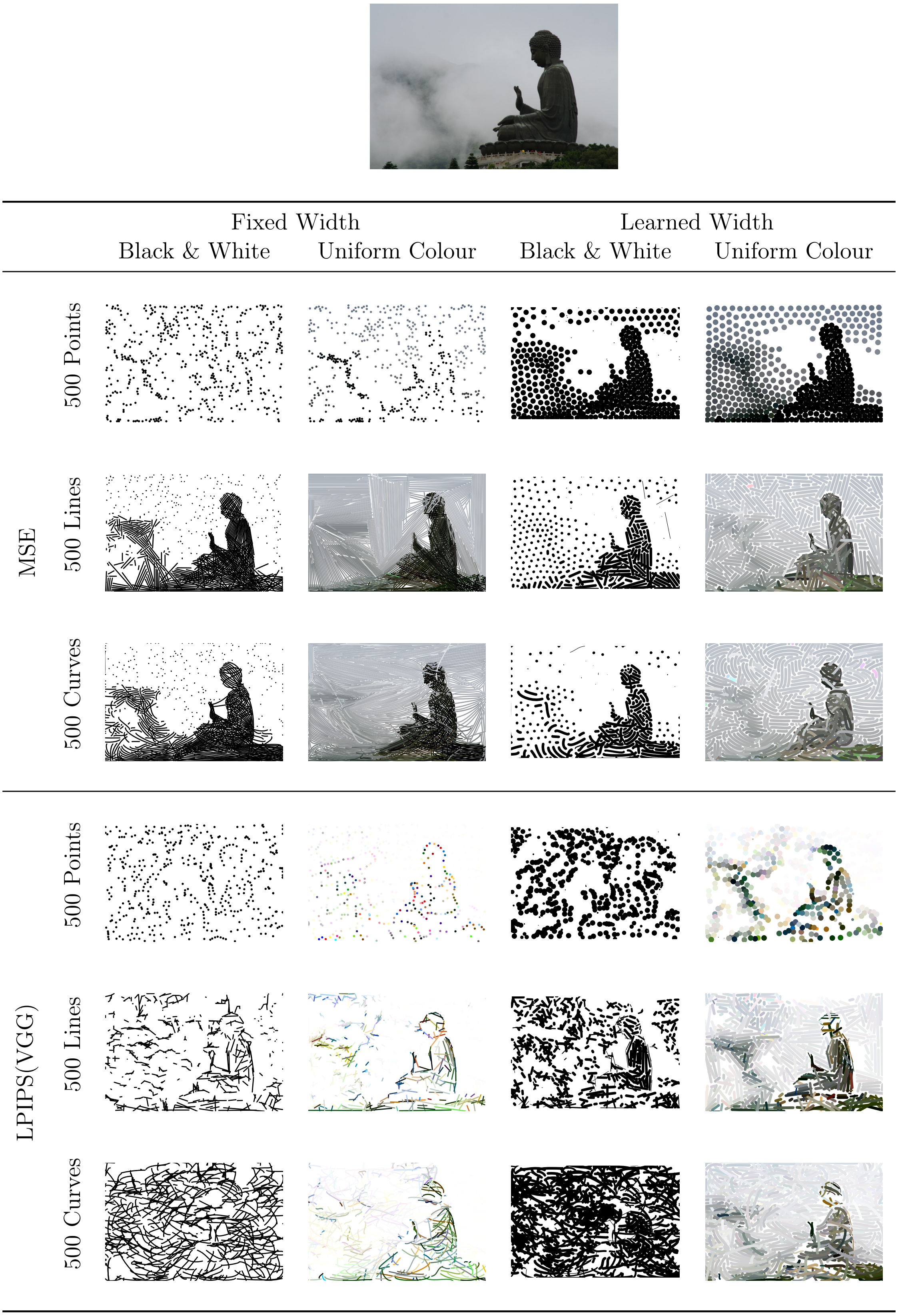
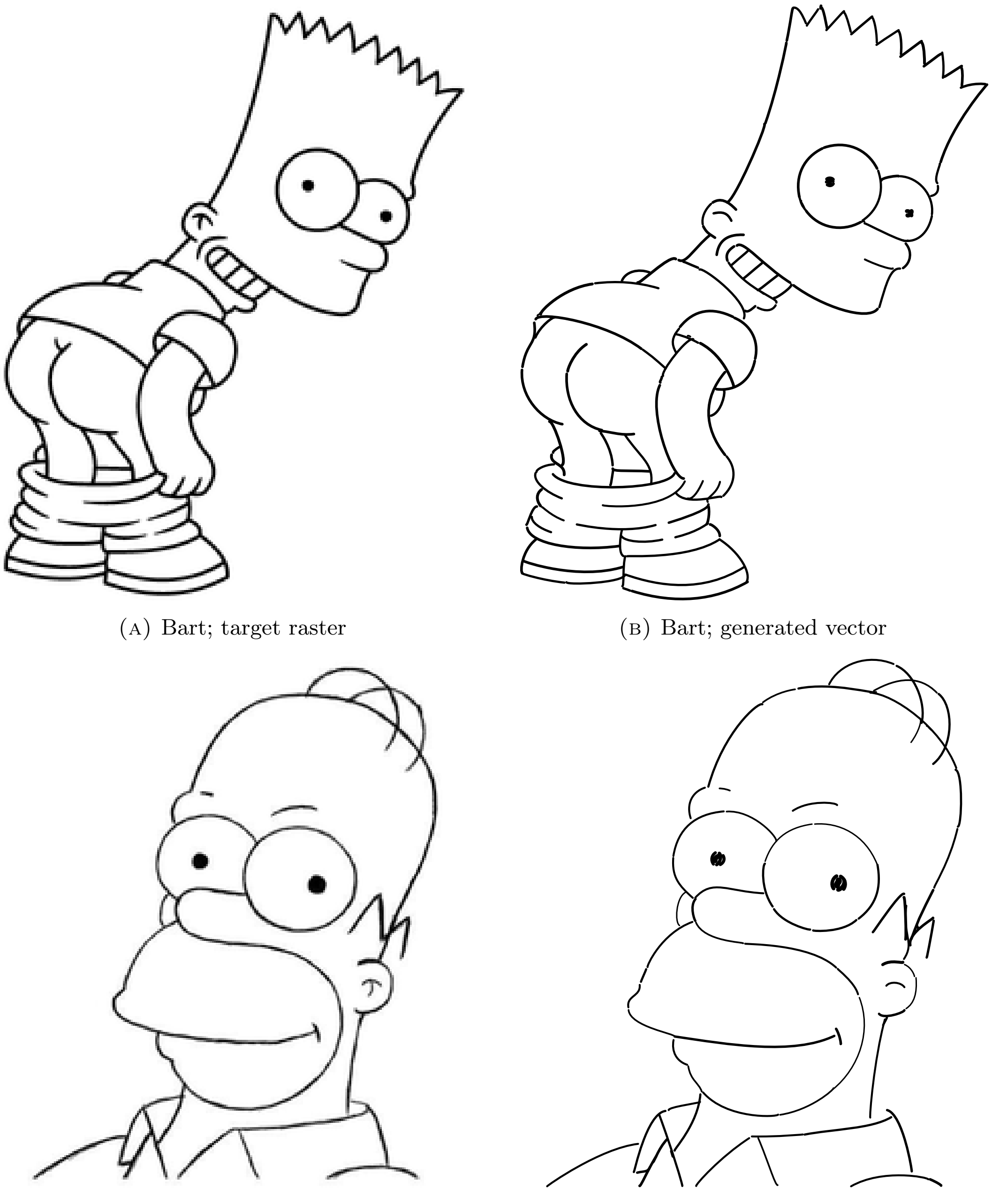


FIGURE 3.18: **Examples of image optimisation (vi).**



(A) Bart; target raster

(B) Bart; generated vector

(C) Homer; target raster

(D) Homer; generated vector

FIGURE 3.19: **Results of converting cartoon rasters to vectors using direct optimisation of Catmull-Rom Spline curve segments with randomly initialised curves and learned stroke width.** 10000 iterations were performed, and for the first 5000 iterations any curve with near-zero stroke width was randomly reinitialised. Zoom-in to better view the differences and the details of the generated vector versions.

3.3.2.3 Physically embodied optimisation

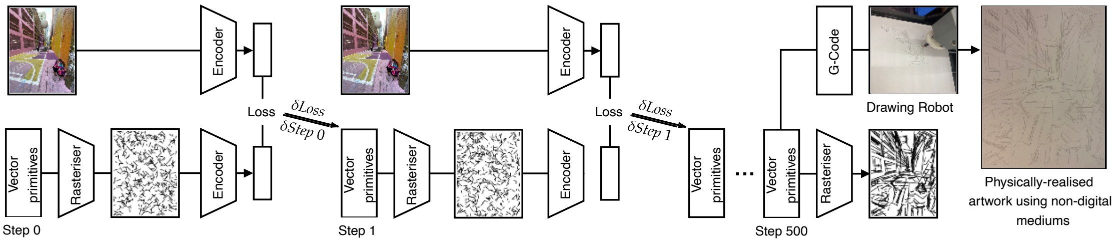

Encoder
Loss
$\frac{\delta Loss}{\delta Step\ 0}$
$\frac{\delta Loss}{\delta Step\ 1}$
Vector primitives
Rasteriser
Step 0
Step 1
Step 500
G-Code
Drawing Robot
Physically-realised artwork using non-digital mediums

FIGURE 3.20: **We create physical sketches by learning programs to control a drawing robot.** A differentiable rasteriser is used to optimise sets of drawing strokes to match an input image, using deep networks to provide an encoding for which we can compute a loss. The optimised drawing primitives can then be translated into G-code commands which command a robot to draw the image using drawing instruments such as pens and pencils on a physical support medium.

We demonstrate that it is possible using the differentiable rasteriser coupled with the direct image optimisation technique to control a drawing robot that can produce physical sketches. Figure 3.20 shows a diagram of the complete system. The optimised stroke parameters can be transformed into instructions that control a drawing robot that can manipulate a drawing instrument like a pen or pencil over a support medium. This allows us to produce physical artefacts directly from the model. Using this technique, we explored how different internal representations of pretrained encoder networks manifest themselves in terms of illustrations of photographs as shown in Figure 3.21[6].

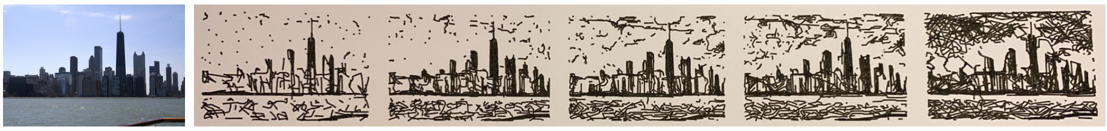


FIGURE 3.21: **lustrations created from VGG-16 network with SIN weights at a range of different depths (early layers left, later layers right).** Each image was created using 1000 straight lines.

For producing the physical sketches we used a custom modified gantry-style robot to manipulate a pen or pencil over a paper support medium as illustrated in Figure 3.22. We used `vpype` (`https://github.com/abey79/vpype`) to optimise the stroke order produced by our network to minimise unnecessary movements between strokes, and `juicy-gcode` (`https://hackage.haskell.org/package/juicy-gcode`) to convert the strokes into G-code instructions that could be performed by the drawing robot.

3.3.2.4 A variation of neural-style transfer

What if Edvard Munch's 'The Scream' was painted in the Pointillism technique? Or Cubism? The direct optimisation technique described in this section can also be viewed

[6] For more examples see our artwork, *Perceptions*, in `https://neuripscreativityworkshop.github.io/2021/#/gallery`.

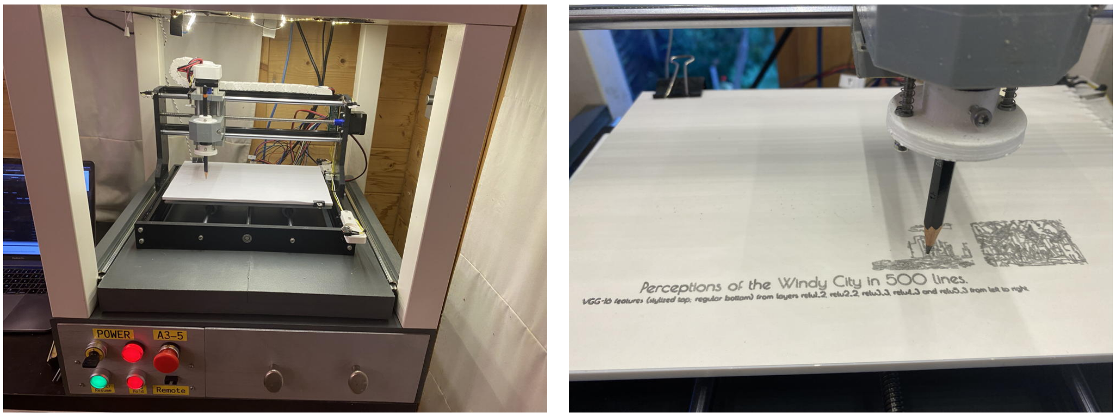
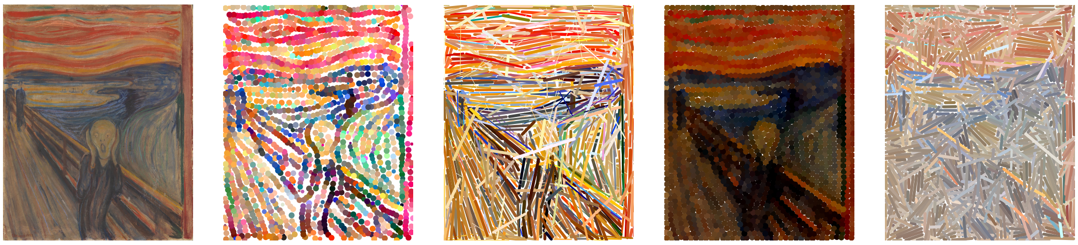

POWER
A3-5
Remote
Perceptions of the Windy City in 500 lines.
VGG-16 features (stylized top, regular bottom) from layers relu1.2, relu2.2, relu3.3, relu4.3 and relu5.3 from left to right.

FIGURE 3.22: **Photos of the drawing robot in action.**

as a variation of the idea of neural style transfer [Gatys et al., 2015]. The distinction, however, is that instead of generating raster images we directly create the underlying stroke information that allows an image to be drawn under different physical constraints.

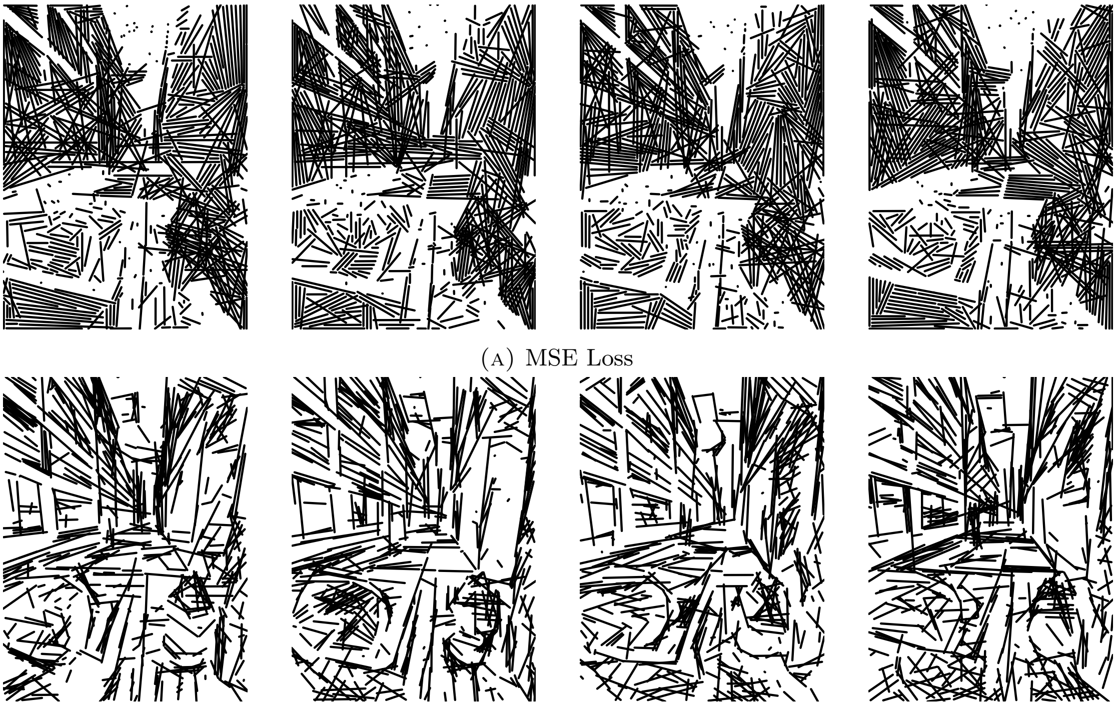


(A) 'The Scream' (B) 2000 Points (C) 1000 Lines (D) 2000 Points (E) 1000 lines

FIGURE 3.23: **Munch's 'The Scream' reduced to points and straight lines by gradient decent** using LPIPS loss with SIN-pretrained VGG16 (figs. 3.23b and 3.23c) and MSE loss (figs. 3.23d and 3.23e).

Figure 3.23 showcases the results of optimising different primitives to fit a photo of the original "The Scream" by Edvard Munch (Figure 3.23a[7]). We display results by optimising 2000 uniformly coloured points (Figures 3.23b and 3.23d) and 1000 coloured lines (Figures 3.23c and 3.23e) to fit the original image by minimising either the MSE or the LPIPS(VGG) loss. The contrast between the images with the same type of primitive parametrisation, but using a different loss, is striking. The perceptual loss captures the shape information rather well while moving away from the colour or texture scheme of the original or the variant realised with MSE loss. The copies thus produced could be tagged as belonging to the Fauvism art movement. Changing the parametrisation of the drawings gives us an idea of what the painting would have looked like if it were to have been drawn in a Pointillist style (Figures 3.23b and 3.23d), or a more abstract, Cubism-like style (Figures 3.23c and 3.23e), while at the same time, making it possible for the drawing to be physically produced on paper by a drawing agent such as the one discussed in Section 3.3.2.3.

[7] Image sourced from `https://en.wikipedia.org/wiki/The_Scream`.



(A) MSE Loss

(B) LPIPS(VGG) Loss

Figure 3.24: **Exploring the effect of different random initialisations.** Four different random seeds were used for the initial line. The resultant images were created by optimising stroke parameters to fit Figure 3.10a using MSE and LPIPS(VGG) loss. Images directly above/below each other correspond to the same random seed.

3.3.3 Potential challenges in optimisation

As mentioned at the beginning of Section 3.3, the loss landscape when using a differentiable rasteriser is challenging because of factors such as permutation symmetry from being able to draw strokes in either direction, as well as many local optima. In the case of direct optimisation, the resultant images can be very sensitive to initialisation. For our autoencoder experiments in Section 3.4, we found no problems with training, however, in the future we want to explore how different priors (*e.g.* preference for long strokes, preference to draw from left to right), could affect this.

Effect of initialisation. Simple losses like MSE are very sensitive to initialisation when used to optimise against a photo. As can be seen in Figure 3.24a, initialisations can have a dramatic effect on the orientation of 'shaded' sections of the resultant image. This is in itself not something to worry about as it just results in a *artistically* different result, but the broad perceptual appearance of tone is still preserved. Different losses can overcome this sensitivity to an extent, however. As can be seen in Figure 3.24b, using the LPIPS(VGG) loss for example always captures perceptually important directions in the resultant image, although the initialisation can still affect the rendition in localised areas.

3.4 Application: Autotracing Autoencoder

We next look at models that learn to perform autotracing of handwritten characters and freehand sketches with only self-supervision. The structure of our autotracing model, shown in Figure 3.1c, is similar to that of a standard autoencoder, with two main components: an image encoder that creates a latent encoding, and a parameter decoder that decodes a latent vector to 'stroke data'. This stroke data is then rasterised into the output image. Both the encoder and parameter decoder have learnable parameters, but the rasterisation is entirely fixed.

We next demonstrate a series of decoders which allow for different approaches to drawing. For example, we consider stroke parametrisation functions such as independent straight lines/curves, connected lines/curves through a series of consecutive points, and sets of points with learned connections between them. These models lay the groundwork for future exploration of learned, differentiable models of sketching that are more similar to how humans write/draw that *e.g.* address the challenges set out by Lake et al. [2015].

Encoders. For experiments on MNIST [LeCun et al., 1998], we present results using a simple multi-layer perceptron encoder network. For more complex characters of Omniglot [Lake et al., 2015], a convolutional network is preferred. When comparing against StrokeNet [Zheng et al., 2019] (Table 3.2b), we replicate their VGG-like Encoder. The exact model architectures are detailed in Mihai and Hare [2021a].

Decoders. Our decoder networks allow for different parametrisations of 'stroke data' that are then used by the rasteriser described in Section 3.2. The decoder transforms a vector encoding of the input image into lists of stroke primitives which aim to reproduce the input image when rasterised. In the simplest case, the latent vector can be decoded to a fixed number of line segments (*LineDecoder*), each defined by its start and end points. Next, we provide *PolyLineDecoder*, for which a stroke is represented as a sequence of consecutive points. Instead of line segments, we can choose to use curves parameterised as Catmull-Rom splines (*CRSDecoder*) or Bézier curves (*BézierDecoder*). In both cases, we can control the number of joined curve segments by specifying how many points (CRS) or segments (Bézier) are used. To allow more flexibility in modelling, we have also explored decoders which incorporate sub-networks to learn to produce a set of 2d points, and the upper-triangular portion of a soft connection matrix between points (optionally including the diagonal). The network producing the connection matrix uses a sigmoid to ensure values are between 0 and 1. To utilise the connection matrix, all possible combinations of lines are rasterised and multiplied by the appropriate connection weight before composition (*PolyConnect*). In the case of Bézier curves (*BézierConnect*) each point in the connection matrix corresponds to both an end point and its corresponding control point, and when drawing curves, the end point is drawn using the mirror of its control point allowing for smooth multiple-segment curves to be created. Zheng et al. [2019] proposed a recurrent model using a visual working memory; the network is presented at each timestep with the features of the target image, together with the current canvas, which is then encoded, concatenated with the input, and transformed to the parameters of a new stroke which is rendered and overlaid on the canvas. We experimented with this approach but found it hard to train and computationally expensive, so we also investigated a simple GRU [Cho et al., 2014] based RNN which is fed a target image's encoding as its initial hidden state, along with a projection of a zeroed input. The GRU output is projected to a set of Bézier curve parameters for rendering, and also re-projected for input at the next time step. Full implementation details are given in Mihai and Hare [2021a]. In the next sections, we present the results of the autotracing model trained on various datasets.

Table 3.2: **Reconstruction performance of parameterisations, measured by MSE and classification accuracy with a classifier trained on unencoded training sets of the respective datasets.** #* indicates the number of (L)ines, (S)egments, and (P)oints. All Scaled MNIST models use the same 'StrokeNet Agent' architecture [Zheng et al., 2019] to map images to primitive parameters.

(a) MNIST Test Dataset (baseline unencoded acc. 98.60%).

Decoder	#P	#S	#L	MSE	Acc.
Line	10	1	5	0.0195	94.06%
PolyLine	16	15	1	0.0225	93.27%
PolyConnect	16	-	-	0.0118	96.47%
CRS	16	14	1	0.0208	94.63%
Bézier	20	1	5	0.0136	96.34%
BézierConnect	16	-	-	0.0116	96.43%

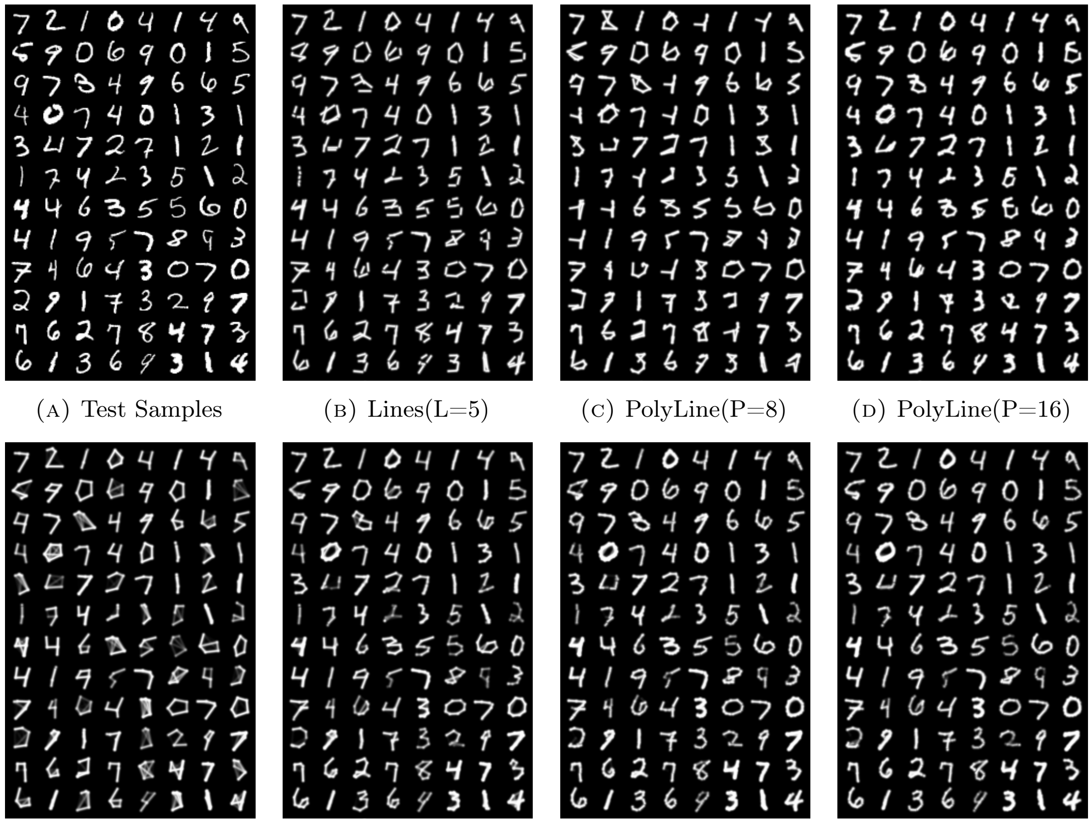
(b) Scaled MNIST Dataset (baseline unencoded acc. 98.58%).

Model	Steps	#P	#S	#L	Acc.
StrokeNet [Zheng et al., 2019]	3 (SN)	16	14	1	95.25%
StrokeNet [Zheng et al., 2019]	1	16	14	1	97.75%
Ours, CRS	1	16	14	1	97.12%
Ours, Bézier	3 (GRU)	4	1	1	96.97%
Ours, Bézier	1	7	2	2	98.28%
Ours, Bézier	1	43	14	1	97.94%

3.4.1 MNIST (28×28 pixels)

Table 3.2a shows the effect of different stroke parametrisations on MNIST. As an objective measure, we compute the classification accuracy of rasterised sketches from the test set using a classifier (baseline accuracy of 98.6%); reconstructions that capture the character should have higher accuracy. *Connect* models, which generate strokes based on a learned connection matrix for the given number of points, perform best due to the flexibility of deciding which points should be joined in a line/curve segment.

Figures 3.25 and 3.26 illustrate the difference between the various stroke parametrisations. Varying the number of (L)ines, (S)egments, and (P)oints and introducing a learned connection matrix between them leads to distinct approaches to drawing. As depicted in figs. 3.25g and 3.26g, *Connect* models produce the closest reconstructions. Likewise, parametrisation using simple Bézier curves (Figure 3.26c) leads to convincing results.



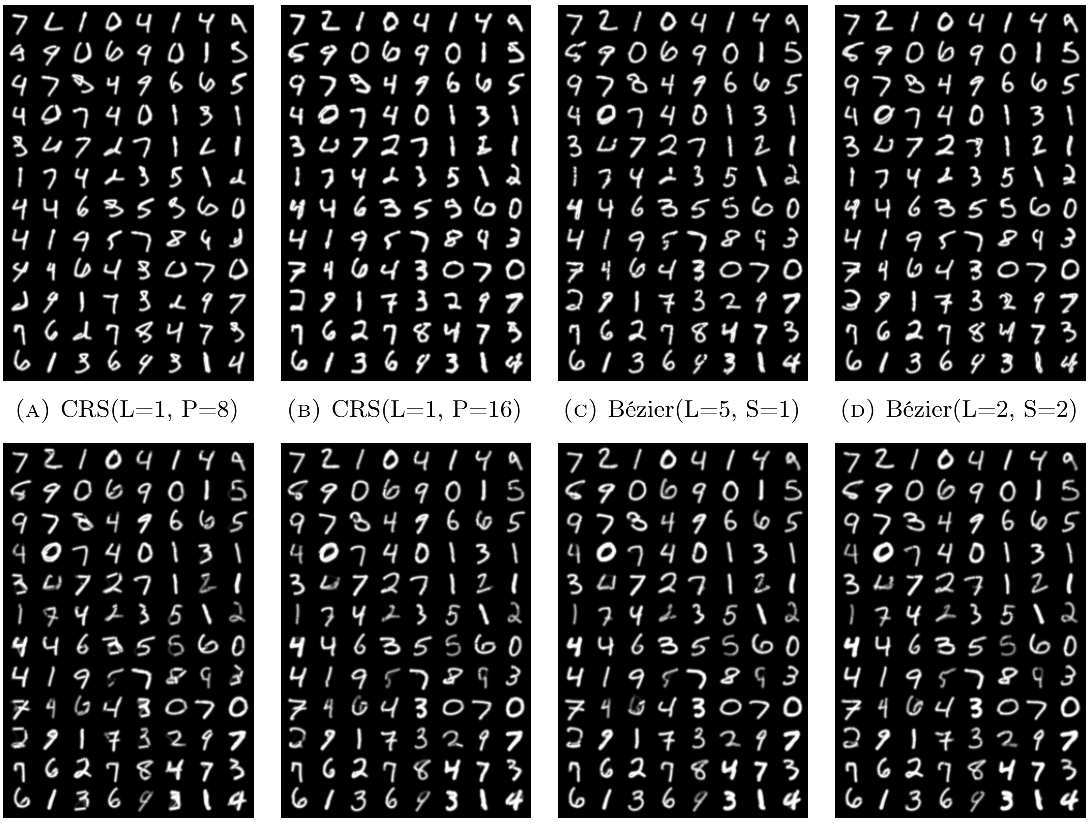
(A) Test Samples (B) Lines(L=5) (C) PolyLine(P=8) (D) PolyLine(P=16)

(E) PolyConnect(P=5) (F) PolyConnect(P=8) (G) PolyConnect(P=16) (H) PolyConnect(P=32)

Figure 3.25: **MNIST test set samples and reconstructions using different parameterisations of 'stroke data'**: Lines, PolyLine (*i.e.* a series of consecutive (P)oints) and PolyConnect (a set of 2d (P)oints joined by a learned connection matrix).

3.4.2 Scaled MNIST (256 × 256 pixels) - StrokeNet comparison

Following Zheng et al. [2019] we perform a similar experiment on their scaled MNIST dataset and also compare to the pretrained StrokeNet models that are publicly available. The StrokeNet paper [Zheng et al., 2019] describes an evaluation of the model on scaled-up MNIST characters by comparing performance against a CNN-based classifier trained on the scaled images, and then evaluated on the reconstructions. The paper implies that the MNIST characters were just re-sampled to 256 × 256, however from analysis of the source code it can be determined that the scaling procedure was to: resize the 28 × 28 characters



(A) CRS(L=1, P=8)

(B) CRS(L=1, P=16)

(C) Bézier(L=5, S=1)

(D) Bézier(L=2, S=2)

(E) BézierConnect (P=5)

(F) BézierConnect (P=8)

(G) BézierConnect (P=16)

(H) BézierConnect (P=32)

FIGURE 3.26: **MNIST test set reconstructions (of samples in Figure 3.25a) continued:** with curves parametrised as Catmull-Rom splines (CRS) and Bézier curves (Bézier and BézierConnect). In both CRS and Bézier Decoders, we can vary the number of (L)ines, (P)oints and, respectively, (S)egments. BézierConnect allows control over the (P)oints joined by the learned connection matrix.

to 120 × 120 using bilinear interpolation, pad the 120 × 120 images to 256 × 256, and change the contrast by multiplying pixels by 0.6. Although the original rationale for these choices is unclear, we follow exactly the same procedure for this experiment.

The structure of the classifier model of Zheng et al. [2019] is not described beyond it being convolutional with 5-layers, and no code for this aspect of the experiments was provided. We thus chose to implement our own classifier as two convolutional layers and three linear layers, each, with the exception of the last, followed by `ReLU` nonlinear function (for implementation details see Mihai and Hare [2021a]).

We did not use any form of regularisation or dropout during training. The classifier network was trained for 10 epochs using the Adam optimiser with a learning rate of 0.001 and PyTorch's `CrossEntropyLoss` which incorporates the Softmax activation. This network performs considerably better than the results presented in the original paper on the raw scaled MNIST test dataset (originally reported accuracy is 90.82%, whereas the above network achieves 98.58%). To compute the performance of the StrokeNet paper with our classification network we take the pretrained model weights provided by the

StrokeNet authors and use them to generate reconstructions of the scaled MNIST test set, which are then fed to the classifier network to make predictions from. Again we found considerably higher performance than was originally reported.

Our autotracing experiment results are shown in Table 3.2b. The accuracies of all models are high indicating good reconstructions, but we note that MNIST doesn't require complex decoders.

3.4.3 Omniglot (28×28 pixels)

Table 3.3 shows the effect of different parameterisations on the Omniglot dataset [Lake et al., 2015]. All the models demonstrate reasonable generalisation to the test dataset (as measured by MSE) even though the test alphabets are completely disjoint from the training/validation ones.

Reconstructions of models with different parametrisations are shown in Figure 3.28. Some small details of the characters are missing, and it is clear that the models do not always choose to draw stokes in the way a human would, but the performance is generally good. Bézier curves work particularly well, although we note that they do appear to struggle with forming dots as is the case in the Braille alphabet which can be found in the training/validation sets (see Figure 3.27).

TABLE 3.3: **Omniglot validation and test MSE for models constructed with different parameterisations and architecture** (*i.e.* recurrent vs single-(St)ep). Bézier* corresponds to the model whose reconstructions were shown in Figure 3.27 and has `hidden1` = 512 and `hidden2` = 1024.

Decoder	St	#P	#S	#L	Val	Test
Line	1	20	1	10	0.0189	0.0223
PolyConnect	1	16	-	-	0.0127	0.0151
Bézier	1	20	1	5	0.0158	0.0194
BézierConnect	1	16	-	-	0.0117	0.0144
RNNBézier	10	16	1	1	0.0152	0.0181
Bézier*	1	50	3	5	0.0091	0.0118

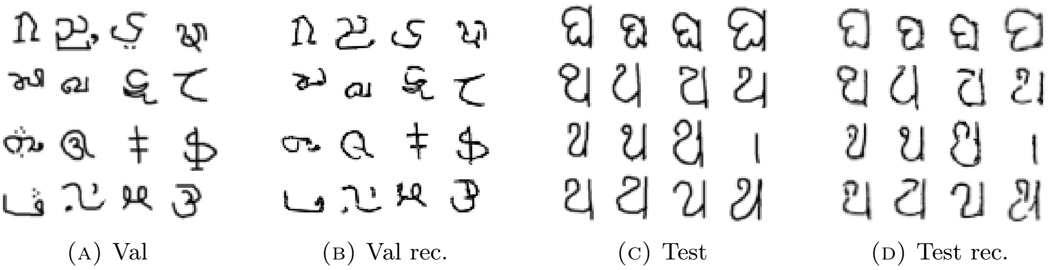

(A) Val (B) Val rec. (C) Test (D) Test rec.

FIGURE 3.27: **28-pixel Omniglot validation and test data samples and *Bézier* model (3 segment, 5 line) reconstructions.**

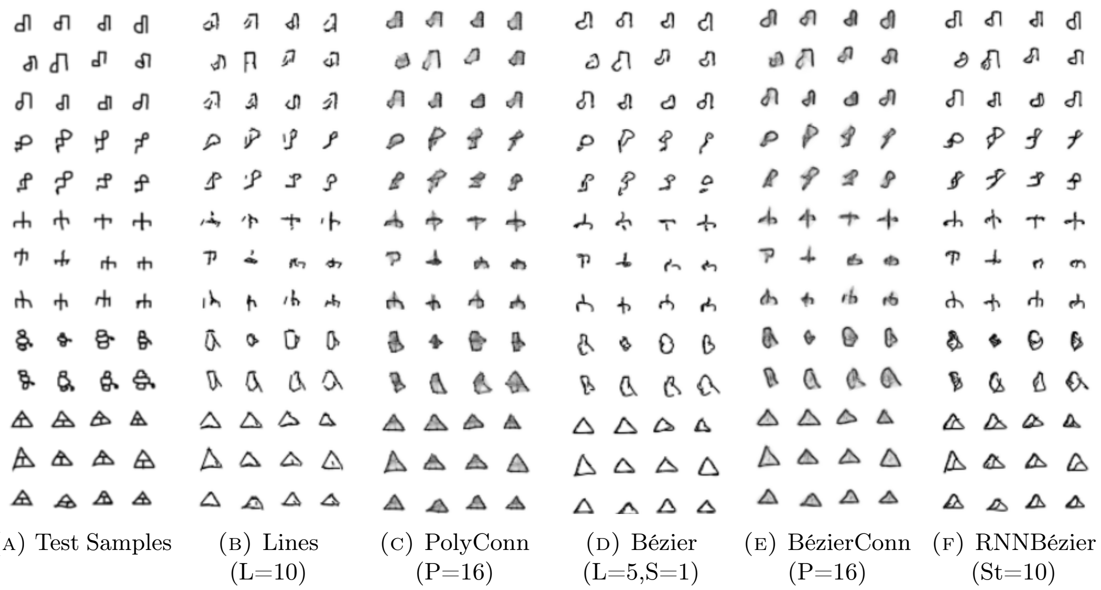

(A) Test Samples (B) Lines (L=10) (C) PolyConn (P=16) (D) Bézier (L=5,S=1) (E) BézierConn (P=16) (F) RNNBézier (St=10)

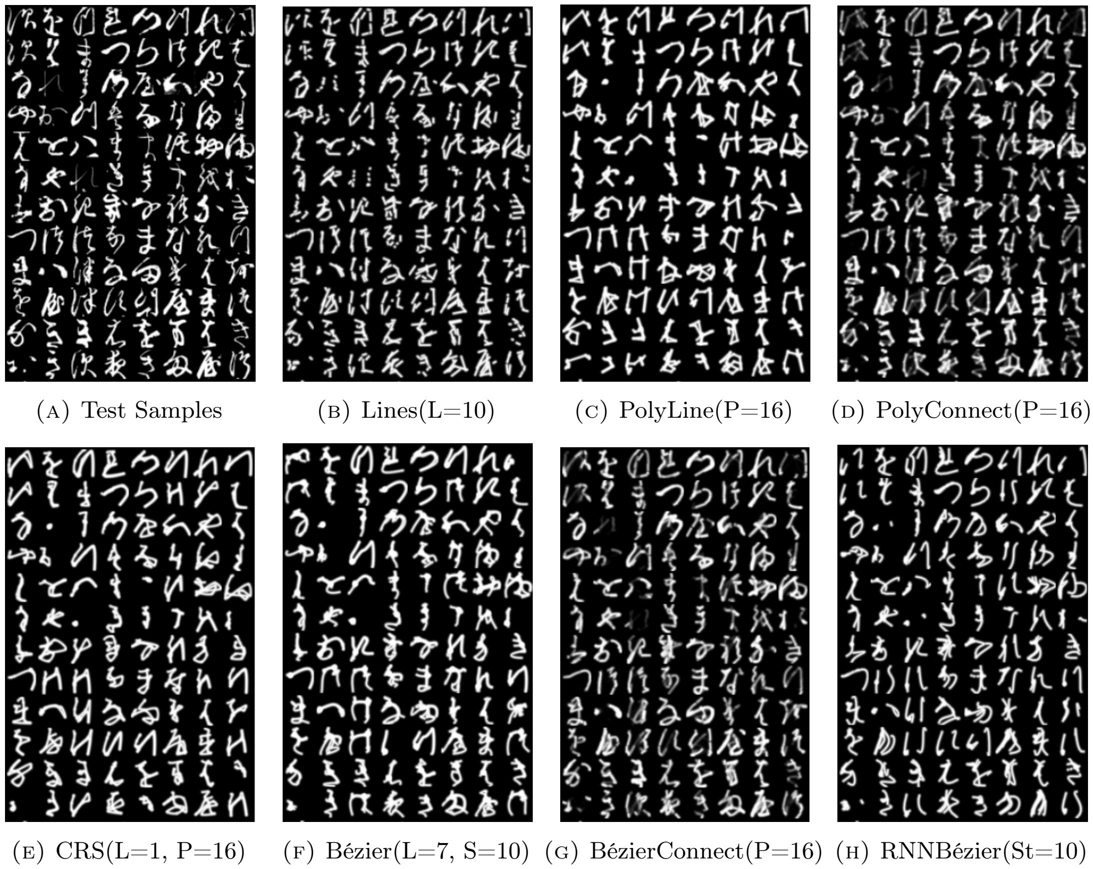
FIGURE 3.28: **Omniglot test set samples and reconstructions using different parameterisations of 'stroke data'.**

3.4.4 KMNIST (28 × 28 pixels)

Table 3.4 shows a comparison between different parametrisations performed on KMNIST [Clanuwat et al., 2018], the Japanese Hiragana dataset. We provide test MSE and the classification accuracy of the drawn sketches. Samples of test reconstructions using different decoders are shown in Figure 3.29. The *BézierConnect* model reaches the highest accuracy and creates the closest reconstructions as shown in Figure 3.29g.

TABLE 3.4: **KMNIST test MSE and classification accuracy (with a classifier trained on the un-encoded training set) for models constructed with different parameterisations.**

Decoder	St	#P	#S	#L	Test	Acc. %
Line	1	20	1	10	0.0431	87.2
PolyLine	1	16	15	1	0.0654	75.06
PolyConnect	1	16	-	-	0.0282	89.09
CRS	1	16	14	1	0.0635	76.2
Bézier	1	217	10	7	0.061	82.07
BézierConnect	1	16	-	-	0.0249	90.15
RNNBézier	10	16	1	1	0.0496	80.19



(A) Test Samples (B) Lines(L=10) (C) PolyLine(P=16) (D) PolyConnect(P=16)

(E) CRS(L=1, P=16) (F) Bézier(L=7, S=10) (G) BézierConnect(P=16) (H) RNNBézier(St=10)

Figure 3.29: **KMNIST test set samples and reconstructions using different parameterisations of 'stroke data'.**

3.4.5 QuickDraw (128 × 128 pixels)

Lastly, we present the results of the autotracing experiment run on the Yoga class of QuickDraw[8], a 50 million human drawing dataset across 345 image categories. For this experiment, a total of 70000 doodles of yoga poses have been used and split so that the

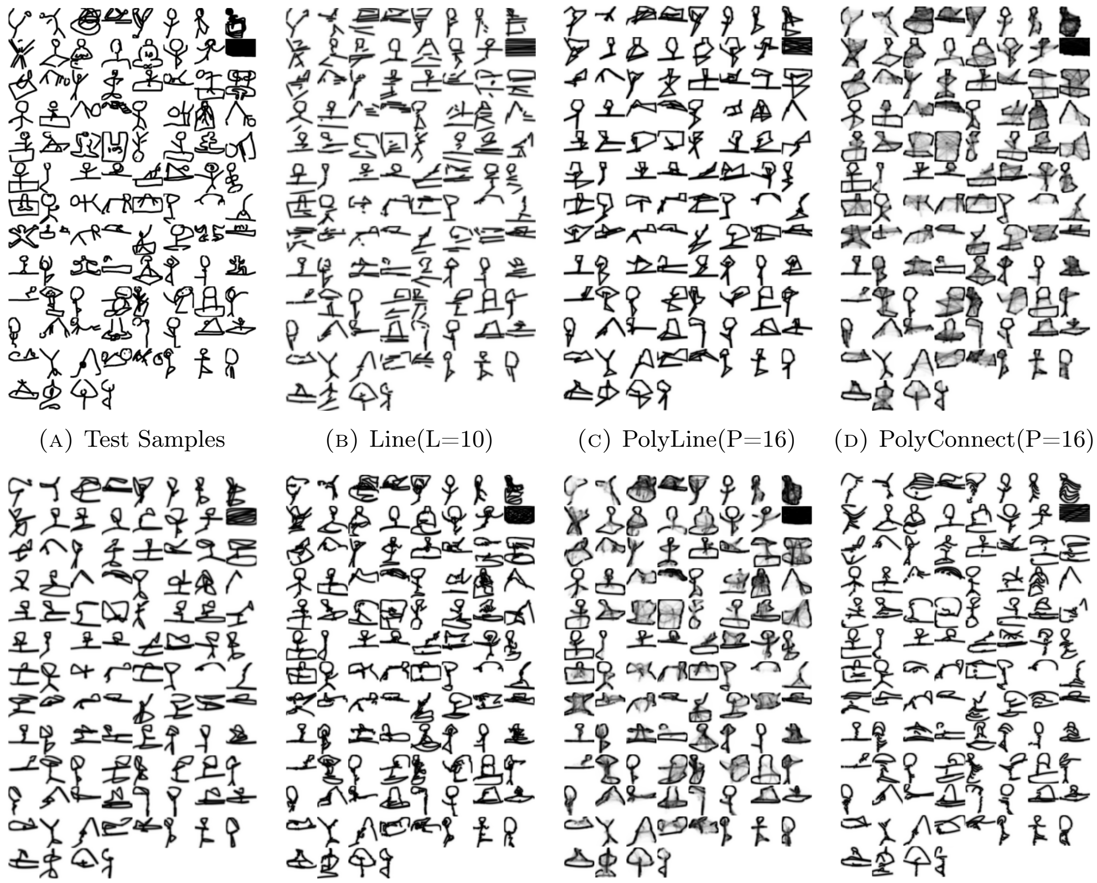
Table 3.5: **QuickDraw validation and test MSE for models constructed with different parameterisations.**

Decoder	St	#P	#S	#L	Val	Test
Line	1	20	1	10	0.086	0.080
PolyLine	1	16	15	1	0.101	0.092
PolyConnect	1	16	-	-	0.063	0.062
CRS	1	16	14	1	0.100	0.091
Bézier	1	50	3	5	0.0766	0.070
BézierConnect	1	16	-	-	0.049	0.048
RNNBézier	10	16	1	1	0.0844	0.076

[8] https://github.com/googlecreativelab/quickdraw-dataset

test, validation and train subsets were disjoint. Table 3.5 shows validation and test MSE for different parametrisations.

Figure 3.30 illustrates reconstructions of test samples for the different models. As seen before, learning the connections between points leads to the best results and produces the most visually similar reconstructions (figs. 3.30d and 3.30g).


(A) Test Samples (B) Line(L=10) (C) PolyLine(P=16) (D) PolyConnect(P=16)

(E) CRS(L=1, P=16) (F) Bézier(L=5, S=3) (G) BézierConnect(P=16) (H) RNNBézier(St=10)

Figure 3.30: **QuickDraw test set samples and reconstructions using different parameterisations of 'stroke data'.** Learning the connections between points leads to the most similar reconstructions (figs. 3.30d and 3.30g).

3.5 Summary

This chapter introduced a derivation of a bottom-up differentiable approach to rasterising vector primitives into images, that allows gradients to flow through every pixel in the image to the underlying primitive's parameters. Our approach allows us to construct end-to-end models of vision that learn primitive parameters directly from raster images. The proposed method of differentiable rasterisation allows us to leverage large raster image datasets and deep-learning frameworks to train end-to-end models using gradient descent. Further, we have demonstrated how effective sketch generation can be achieved with different losses, and how parameterisations can change what a model learns.

Our approach is only a building block towards future applications and research. Our own motivation for designing this approach is to use it to explore visual communication, as opposed to discrete language, although there are undoubtedly many potential use-cases. For us, questions to be answered based on the work presented in this chapter involve looking at how one might build models that can learn to produce the appropriate number of strokes (and choose between different types of primitive). As part of this, it is clear that reconstruction performance alone should not be the key driver of gradient; the ability to communicate information is more important. Both attention and weak supervision to better mimic humans are also key to this endeavour.

Chapter 4

Communication through Sketching

"It doesn't matter if you can't speak the same language. If you have pictures, or better still, if you can draw things, then you can communicate anything to anyone."

— Antony Gormley

"I prefer drawing to talking. Drawing is faster, and leaves less room for lies."

— Le Corbusier

Imagine you and a friend are playing a game where you have to get your friend to guess an object in the room by you sketching the object. No other communication is allowed beyond the sketched image. This is an example of a *referential communication game*. To play this game you need to have learned how to draw in a way that your friend can understand. This chapter explores how artificial agents parameterised by neural networks can learn to play similar drawing games and, thus, explore a different communication modality than usually explored in emergent communication literature.

Evidence that visual communication preceded written language and provided a basis for it goes back to prehistory, in forms such as cave and rock paintings depicting traces of our distant ancestors. Emergent communication research has sought to explore how agents can learn to communicate in order to collaboratively solve tasks. As discussed in Chapter 2, existing research in this area has focused on language, with a learned communication channel transmitting sequences of discrete tokens between the agents. In this chapter, we shift our attention to a visual communication channel between agents that are allowed to draw with simple strokes. Our agents are parameterised by deep neural networks and the differentiable drawing procedure presented in Chapter 3 allows for end-to-end training. In the framework of a referential communication game, we demonstrate that agents can not only successfully learn to communicate by drawing, but with appropriate inductive biases, can do so in a fashion that humans can interpret. The aim of this chapter is to encourage future research to consider visual communication as a more flexible and directly interpretable alternative of training collaborative agents. The work presented here led to our NeurIPS 2021 conference paper [Mihai and Hare, 2021b].

The chapter starts by discussing the motivation for exploring a visual communication channel instead of a token-based protocol and highlights the contributions. It then gives an overview of relevant literature which explored visual communication (see Section 4.2). The third section describes the experimental setup, covering the game objectives, the drawing agents' architecture as well as training details and additional constraints. The first series of experiments (Section 4.4) explore the visual communication protocol emerging between artificial agents. The second experimental part (Section 4.5) attempts to answer the question of whether the emergent protocol can be interpreted by humans and to identify the factors which encourage semantic interpretability.

4.1 Motivation and Contributions

As discussed in Section 1.3.2.1, innovations in artificial neural networks, deep and reinforcement learning techniques have led to research on multi-agent emergent communication that pursues interactions in the form of gameplay between agents to induce human-like communication [Chaabouni et al., 2020; Guo, 2019; Ren et al., 2020; Lazaridou et al., 2018; Havrylov and Titov, 2017]. Artificial communicating agents can collaborate to solve various tasks: image referential games with realistic visual input [Havrylov and Titov, 2017; Lazaridou et al., 2017, 2018], negotiation [Cao et al., 2018], navigation of virtual environments [Das et al., 2019; Jaques et al., 2019], reconstruction of missing input [Chaabouni et al., 2020; Guo, 2019] and, more recently, drawing games [Fernando et al., 2020]. The key to achieving the shared goal in many of these games is collaboration, and implicitly, communication. To date, most studies on communication emergence in multi-agent games have focused on exploring a language-based communication channel as the one presented in Chapter 2, with messages represented by discrete tokens or token sequences [Havrylov and Titov, 2017; Mordatch and Abbeel, 2017; Das et al., 2017; Lazaridou et al., 2018, 2017; Kharitonov et al., 2020; Guo, 2019]. However, these communication protocols, although efficient for solving the task, can be difficult for a human to interpret, especially without further processing or human supervision [Kottur et al., 2017; Lowe et al., 2019; Chaabouni et al., 2019]. In this chapter, we propose a direct and potentially self-explainable means of transmitting knowledge: *sketching*.

Concretely, we propose a visual communication channel in the context of image-based referential games. We leverage the method for differentiable sketching that enables us to construct an agent that *can learn to communicate intent through drawing*. Through a range of experiments, we show that:

- Agents can successfully communicate about real-world images through a sketching game. However, training with a loss that tries to maximise gameplay alone does not lead to human decipherable sketches, irrespective of any visual system preconditioning;

- Introducing a perceptual loss improves human interpretability of the communication protocol, at little to no cost in the gameplay success;
- Changes to the game objective, such as playing an object-oriented game, can steer the emergent communication protocol towards a more pictographic or symbolic form of expression;
- Inducing a shape bias into the agents' visual system leads to more explainable drawings;
- A drawing agent trained with a perceptual loss can successfully communicate and play the game with a human.

4.2 Sketching as Visual Communication

We take inspiration from the process and evolution of writing discussed in Section 1.3.1. Written language has undergone many transitions from early times to reach the forms we now know: from pictures and drawings to word-syllabic, syllabic and, finally, alphabetic systems. Evidence suggests pre- and early-humans were able to communicate by drawing long before developing the various stages of written language [Henshilwood and Dubreuil, 2009; Robinson, 2002]. Drawings such as petrograms and petroglyphs exist from the oldest palaeolithic times and may have been used to record past experiences, events, beliefs or simply the relation with other beings [Hoffmann et al., 2018; Fox, 1937]. These pictorial characters which are merely impressions of real objects or beings stand at the basis of all writing [Gelb, 1963]. Studies on the communication systems developed in primitive societies compare ancient drawings to the very early sketches drawn by children and talk about their tendency of concretely identifying certain things or events in their surrounding world [Gelb, 1963; Kellogg, 1969]. Psychological and behavioural studies have shown that children try to communicate to the world through the images they create even when they cannot associate them with words [Farokhi and Hashemi, 2011]. This leads us to question if drawing is a more natural way of *starting* to study emergent communication and if it could lead to better-written communication later on.

Drawing can also be regarded as an effective tool for organising information and displaying patterns, otherwise hidden in numerical or linguistic representations. As a visualisation technique, drawing has been studied in relation to cognitive functions such as observation, communication, explanation and problem-solving [Ainsworth et al., 2011; Fan, 2015]. These are essential traits in developing scientific thinking. In Section 1.3.3, we covered the relevance of games which involve drawing, such as Pictionary, for educational purposes. Fan [2015] reviews evidence from the cognitive and educational research literature on the role of drawing in improving such cognitive functions. This work highlights several benefits of drawing including improved observational capacities and ability to communicate scientific thinking in social contexts. Moreover, Fan [2015]'s analysis encourages the expansion of graphical literacy as it has the potential to help people discover more than the real and tangible world around them, but also envisage and create an improved state for it.

Particularly relevant to the work presented in this chapter is the study of Schwartz [1995], which shows that pairs tasked with problem-solving tend to develop more abstract visual representations, such as directed-graphs and matrices, compared to similar individuals solving the problem on their own. This finding is thought to be connected to the need of negotiating a *common* representation that links the two perspectives upon the problem's solution. On the other hand, an individual working alone, *i.e.* without feedback, on a problem is likely to produce more complex and pictorial problem-solving representations.

There exists a broad line of research on the emergence of graphical conventions established throughout human communication [Fay et al., 2010; Brennan and Clark, 1996; Galantucci, 2005; Garrod et al., 2007]. More recently in the cognitive science literature, neural models of sketching have been developed to study the factors which enable contextual flexibility in visual communication [Fan et al., 2020]. Hawkins et al. [2021] also studied how the visual resemblance between objects and drawings, and social communicative context shape the emergence of graphical conventions amongst humans. This study included a series of experiments, in which similar to our work, participants, humans in their case, played a Pictionary-like communication game. Their findings show that *visual resemblance* plays a key role initially. However, as the gameplay progresses, participants rely more on the previously *shared* experience, also observed by Fay et al. [2010]. Moreover, a shared experience with the same partner leads to much simpler drawings over time that preserve the most distinctive visual information. In this sense, Garrod et al. [2007] also provides evidence that, with progressive interaction, drawings become more iconic or symbolic. Another important factor in the simplification of drawings is *feedback*, as it has been shown that in the absence of it, drawings become more complex [Garrod et al., 2007; Hupet and Chantraine, 1992; Schwartz, 1995].

A different direction which explores drawing includes works such as Fernando et al. [2020] who attempt to automate the artistic process of drawing by training agents in a reinforcement learning framework, to play a variety of drawing games. However, the focus of our research is to open the doorway to exploring different types of communication between artificial agents and humans. The novelty of our work is also evident in the model framework which can be easily extended well beyond aspects previously studied.

Concurrent to our study of emergent communication using the modality of sketches, Qiu et al. [2021] also enabled agents to communicate through sketching but focused on the emergence and evolution of graphical symbols. There are several significant differences between our approaches. From an architectural point of view, their sender agent is modelled as a two-staged module, which first encodes images to sketches by capturing the edge information before starting to play the game. The sketch image is then concatenated with a blank canvas and processed by a pretrained sketching module which outputs vector parametrisations of five strokes at each step. The previously blank canvas is updated with the new strokes and passed to the receiver to query the context images; these steps repeat until the receiver makes a choice. The sequencing of actions seems rather unnatural and the pretrained sketching module requires discriminative subject classes to perform well.

4.3 A Model for Learning to Communicate by Drawing

We present a model consisting of two agents, the sender and the receiver, in which the sender learns to communicate through drawing while playing a game with the receiver. The overall architecture of the agents in the context of the game they are learning to play is shown in Figure 4.1. Full code for the model and all experiments can be found at `https://github.com/Ddaniela13/LearningToDraw`.

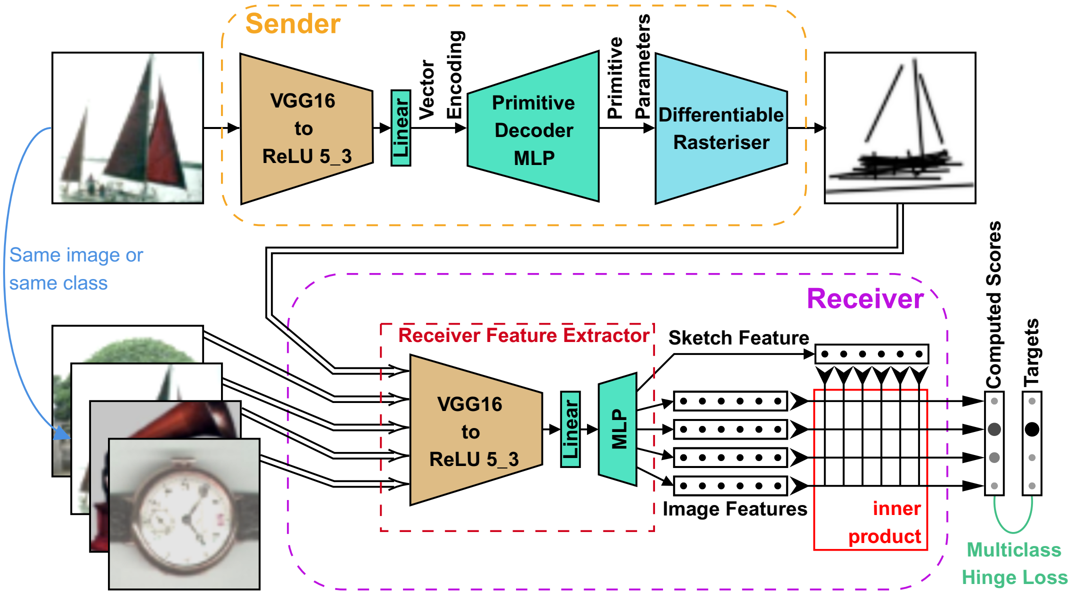


FIGURE 4.1: **Overview of the agent architecture and game setup.** The 'sender' agent is presented with an image and sketches its content through a learnable drawing procedure. The 'receiver' agent is presented with the sketch and a collection of photographs, and has to learn to correctly associate the sketch with the corresponding photograph by predicting scores which are compared to a one-hot target. Both agents are parameterised by neural networks trained end-to-end using gradient methods.

4.3.1 The game environment

Our experimental setup builds upon the image referential game previously explored in studies of emergent communication [Havrylov and Titov, 2017; Lazaridou et al., 2017, 2018] that derives from Lewis's signalling game [Lewis, 1969]. We implemented several variants of Havrylov and Titov [2017]'s image guessing game. The overall setting of these games is formulated as follows:

1. Two target photographs, $\mathbf{P}_s$ and $\mathbf{P}_r$, and set of K distractor photographs $\{\mathbf{P}_d^{(k)}\}_{k=1}^K$ are selected.
2. There are two agents: a sender and a receiver.
3. After being presented the $\mathbf{P}_s$ target image, the sender has to formulate a message conveying information about that image.
4. Given the message and the set of photographs, $\{\mathbf{P}_d^{(k)}\}_{k=1}^K \cup \{\mathbf{P}_r\}$, consisting of all the distractors and the target $\mathbf{P}_r$, the receiver has to identify the target correctly.

The specifics of how the photographs are selected (step 1 above) depend on the game variant as described below. Success in these games is measured by the binary ability of the receiver to correctly guess the correct image or not; as such, the measure of *communication rate* is used to assess averaged performance over many games using independent images to those used during training. Unlike Havrylov and Titov [2017]'s game in which the sender helps the receiver identify the correct image by sending a message constructed as a sequence of tokens drawn from a predefined vocabulary, in this chapter we propose using a directly interpretable means of communication: *sketching the target photograph.*

Original game variant. In Havrylov and Titov [2017]'s variant of the game there is a pool of photos from which the distractors and target $\mathbf{P}_s$ are drawn randomly without replacement. The target $\mathbf{P}_r$ is set to be equal to $\mathbf{P}_s$. In our *original* variant experiments the number of distractors, K, is set to 99.

Object-oriented game variants. In addition to the original setup, we explored two slightly different and potentially harder game configurations which were intended to induce the agents to draw sketches that would be more representative of the object class they belong to rather than to the specific instance of the class. These setups use labelled datasets where each image belongs to a class based on its contents. In the first of these variants (we refer to this as *OO-game same*), the target $\mathbf{P}_r$ is set to be equal to $\mathbf{P}_s$, and the distractors and target are sampled such that their class labels are disjoint (that is every photo provided to the receiver has a different class). The second setup (*OO-game different*) is similar to the first, but the target $\mathbf{P}_r$ is chosen to be a different photograph with the same class label as target $\mathbf{P}_s$. The intention behind these games is to explore a universally interpretable depiction of the different object classes, which does not focus on individual details but rather conveys the concept. To some extent, this task is an example of multiple instance classification within a weakly supervised setting [Amores, 2013], which has been previously explored in the emergent communication literature [Lazaridou et al., 2017].

4.3.2 Agents' architectures

Both agents act on visual inputs; the role of the sender is to create a sketch based on a single input photograph, whilst the receiver takes the sketch and a set of photographic inputs and produces a score for each photograph as output. The agents are parameterised by deep neural networks and are trained using standard gradient techniques (Section 4.3.3).

The agent's early visual system. We choose to model the early visual systems of both agents with the head part of the VGG16 CNN architecture [Simonyan and Zisserman, 2015] through to the ReLU activation at the end of the last convolutional layer (commonly referred to as the `ReLU5_3` layer) before the final max-pooling and fully connected layers. In all experiments presented in this chapter, we utilise pretrained weights and freeze this part of the model during training. We justify this choice on the basis that it provides the agents with an initial grounding in understanding the statistics of the visual world, and ensures that the visual system cannot collapse and remains universal. The weights are the standard `torchvision` ImageNet weights, except in the cases where we explore the effect of shape bias (see Section 4.4.10). As these pretrained weights were learned with images that were normalised according to the ImageNet statistics, all inputs to the VGG16 backbone (including sketches) are normalised accordingly. The output feature maps of this convolutional backbone are flattened and are linearly projected to a fixed dimensional vector encoding (64-dimensions unless otherwise specified). Because the datasets used in gameplay have different resolutions, the number of weights in the learned projection varies. Lastly, it is worth noting that our intuition for using a VGG16 over other networks comes from it still being widely used as a proxy for the early human visual system or just as a feature extractor in many recent papers [Singer et al., 2020; Nonaka et al., 2021; Storrs et al., 2020]. In Section 5.1 we look at a more recent feature extraction backbone.

Sender agent. The goal of the sender is to produce a sketch from the input photograph. For experiments in Section 4.4, we restrict the production of sketches to be a drawing composed of 20 black, constant width, straight lines on a white canvas of the same size as the input images. An investigation into the effect of the sketch complexity, *i.e.* varying the number of lines, can be found in Section 4.4.4. It is of course possible to have a much more flexible definition of a sketch and incorporate many different modelling assumptions. We choose to leave such exploration for future work and focus on the key question of whether we can actually achieve successful (and potentially interpretable) communication with our simplified but not unrealistic setup.

Given an input image, the agent's early visual system produces a vector encoding which is then processed by a three-layer multilayer perceptron (MLP) that learns to decode the primitive parameters used to draw the sketch. This MLP has ReLU activations on the first two layers and tanh activation on the final layer. Unless otherwise specified, the first two layers have 64 and 256 neurons respectively. The output layer produces four values for each line that will be drawn; the values are the start and end coordinates of each line stroke in an image canvas with the origin at the centre and edges at ±1.

To produce a sketch image from the line parameters output by the MLP, we utilise the differentiable rasterisation approach introduced in Chapter 3. At a high level, this approach works by computing the distance transform on a pixel grid for each primitive being rendered. A relaxed approximation of a rasterisation function is applied to the distance transform to compute a raster image of the specific primitive. Finally, a differentiable composition function is applied to compose the individual rasters into a single image. More specifically, the squared Euclidean Distance Transform is computed, $\mathrm{d}^2_{\mathrm{seg}}(\boldsymbol{s}, \boldsymbol{e})$ over all pixels in the image, for each line segment starting at coordinate $\boldsymbol{s}$ and ending at $\boldsymbol{e}$. These squared distance transforms are simply images in which the value of each pixel is replaced with the closest squared distance to the line (computed when the pixels are mapped to the same coordinate system as the line — so the top left of the image is $(-1, -1)$ and bottom-right is $(1, 1)$). Using the subscript i to refer to the i-th line in the sketch, each $\mathrm{d}^2_{\mathrm{seg}}(\boldsymbol{s_i}, \boldsymbol{e_i})$ is rasterised as

$$\mathbf{R}_i = \exp\left(-\frac{\mathrm{d}^2_{\mathrm{seg}}(\boldsymbol{s_i}, \boldsymbol{e_i}))}{\sigma^2}\right) , \tag{4.1}$$

where σ^2 is a hyperparameter that controls how far gradients flow in the image, as well as the visible thickness of the line ($\sigma^2 = 5 \times 10^{-4}$ for all experiments in this chapter). We adopt the soft-or composition function (see Equation 3.19) to compose the individual line rasters into a single image, but incorporate an inversion so that a sketch image, $\mathbf{S}$, with a white canvas and black lines is produced,

$$\mathbf{S} = \prod_{i=1}^{n} (\mathbf{1} - \mathbf{R}_i) , \tag{4.2}$$

where n is the number of lines. Finally, because the backbone CNNs work with three-band colour images, we replicate the greyscale sketch image three times across the channel dimension.

Receiver Agent. The receiver agent is given a set of photographs and a sketch image and is responsible for predicting which photograph matches the sketch under the rules of the specific game being played. The receiver's visual system is coupled with a two-layer MLP with a ReLU nonlinearity on the first layer (the latter layer has no activation function). Unless otherwise specified, all experiments use 64 neurons in the first layer and 64 in the final layer. The sketch image and each photograph are passed through the visual system and MLP independently to produce a feature vector representation of the respective input. A score vector $\boldsymbol{x}$ is produced for the photographs by computing the scalar product of the sketch feature with the feature of each respective photograph. This score vector is un-normalised but could be viewed as a probability distribution by passing it through a softmax. The photograph with the highest score is the one predicted.

4.3.3 Training details

By incorporating a loss between the predicted scores of the receiver agent and the known correct target photograph, it is possible to propagate gradients back through both the receiver and sender agents. As such, we can train the agents to play the different game settings. For the loss function, we follow Havrylov and Titov [2017] and choose to use Weston and Watkins [1999]'s multi-class generalisation of hinge loss (aka multi margin loss),

$$\mathrm{l}_{\mathrm{game}}(\boldsymbol{x}, y) = \sum_{j \neq y} \max(0, 1 - \boldsymbol{x}_y + \boldsymbol{x}_j) \ , \tag{4.3}$$

where $\boldsymbol{x}$ is the score vector produced by the receiver, and y is the true index of the target, and the subscripts indicate indexing into the vector. The rationale for this choice is that the (soft) margin constraint should help force the distractor photographs' features to be more dissimilar to the sketch feature. Tests using cross-entropy also indicated that it could work well as an alternative, however.

Optimisation of the parameters of both agents is performed using the Adam optimiser with an initial learning rate of 1×10^{-4} for all experiments. For efficiency, we train the model with batches of games where the sender is given multiple images which are converted to sketches and passed to the receiver which reuses the same set of photographs for each sketch in the batch (with each sketch targeting a different receiver photograph). The order of the targets with respect to the input image's sketches is shuffled every batch. Batch size is $K + 1$, where K is the number of distractors, for all experiments. Unless otherwise stated, training was performed for 250 epochs. A mixture of Nvidia GTX1080s, RTX2080s, Quadro RTX8000s, and an RTX-Titan was used for training the models. Higher resolution images required more memory. Training time varied from around 488 games/second (10 secs/epoch) for games using STL-10 [Coates et al., 2011] to around 175 games/second (around 5 mins/epoch) for Caltech-101 [Fei-Fei et al., 2004] experiments with 128×128 pixel images.

4.3.4 Making the sender agent's sketches more perceptually relevant

Perception of drawings has a long history of study in neuroscience [see e.g. Sayim and Cavanagh, 2011, for an overview]. In order to induce the sender to produce sketches that are more interpretable, we explore the idea of using an additional loss function between the differences in feature maps of the backbone CNN from the produced sketch and the input image. Such a loss has a direct grounding in biology, where it has been observed through human brain imaging studies that sketches and photographs of the same scene result in similar activations of neuron populations in area V4 of the visual cortex, as well as other areas related to higher-order visual cognition [Walther et al., 2011]. At the same time, it has also been demonstrated that differences in feature maps from pre-trained CNN architectures can be good proxies for approximating human notions of perceptual similarity between pairs of images [Zhang et al., 2018].

Inspired by Zhang et al. [2018] we formulate a loss based on the normalised differences between feature maps of the backbone network from the application of the network to both the input photograph and the corresponding sketch. Unlike Zhang et al., we choose not to learn weightings for each feature map channel individually, but rather we consider all feature maps produced by a layer of the backbone to be weighted equally. Learning individual channel weighting would be an interesting direction for future research, but is challenging because we would want to avoid the network learning zero weights for each channel, where the perceptual loss is basically ignored.

Figure 4.2 illustrates our perceptual loss formulation; note also that unlike Zhang et al. [2018] the final averaging operation does incorporate a (per-layer) weighting, $\boldsymbol{w}_l$, which we explore the effect of in Section 4.4.7. More formally, denoting the sketch as $\mathbf{S}$ and corresponding photo as $\mathbf{P}$, we extract $L = 5$ feature maps, $\hat{\mathbf{S}}^{(l)}, \hat{\mathbf{P}}^{(l)} \in \mathbb{R}^{H_l \times W_l \times C_l}$, for the l-th layer from the backbone VGG16 network and unit normalise each across the channel dimension. The loss is thus defined as,

$$\mathrm{l}_{\text{perceptual}}(\mathbf{S}, \mathbf{P}, \boldsymbol{w}) = \sum_l \frac{\boldsymbol{w}_l}{H_l W_l} \sum_{h,w} \|\hat{\mathbf{S}}^{(l)}_{hw} - \hat{\mathbf{P}}^{(l)}_{hw}\|_2^2 \,. \tag{4.4}$$

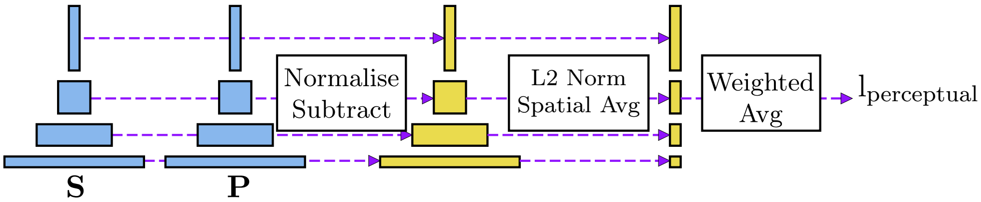


FIGURE 4.2: **Computing a 'perceptual' loss with the early visual system.** Features are extracted from the sketch **S** and corresponding photograph **P** from different layers of the backbone. The features are normalised over channels and subtracted. We take the sum of the squared differences over channels and average spatially. Finally, we compute a weighted average across layers.

To extract the feature maps we choose to use the outputs of the VGG16 layers immediately before the max-pooling layers (`relu1_2`, `relu2_2`, `relu3_3`, `relu4_3` and `relu5_3`). During training, this perceptual loss is added to the game loss ($\mathrm{l}_{\mathrm{game}}$). We note that the perceptual loss formulation is equivalent to the *content loss* in neural style transfer [Gatys et al., 2016]. Neural style transfer combines this content loss with a *style loss* which encourages the texture statistics of a generated raster image to match a target image (which could be a sketch). Our model is different because, as highlighted in Section 3.3.2.4, instead of a loss *encouraging* a sketch-like style we directly *impose* production of sketches by drawing strokes.

4.4 Experiments and Findings

We next present a series of experiments where we explore if it is possible for the two agents to learn to successfully communicate, and what factors affect human interpretation of the drawings. We report numerical results averaged across 10 seeds for models evaluated on test sets isolated from training. Sample sketches from one seed are shown, but an overlay of 10 seeds can be found in Section 4.4.6.

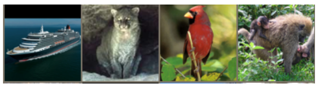
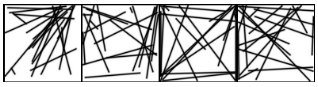
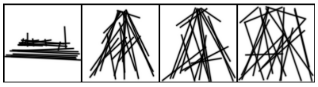
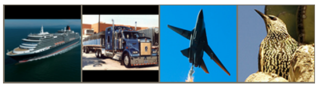
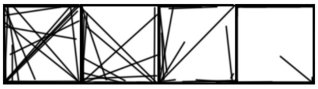
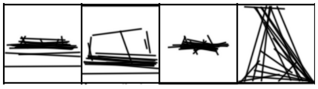
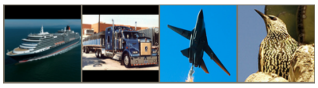
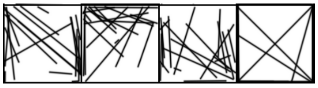
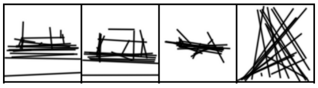
4.4.1 Can agents communicate by learning to draw?

We explore the game setups described in Section 4.3.1 and train our agents to play the games using 96×96 photographs from the STL-10 dataset [Coates et al., 2011]. For the *original* game we use 99 distractors. For the object-oriented games, due to the dataset only having 10 classes, we are limited to 9 distractors.

In Table 4.1, we show quantitative and qualitative results of the visual communication game played under the three different configurations. The results demonstrate that it is possible for agents to successfully play this type of image referential game by learning to draw. One can observe that although agents achieve a high communication success rate, using only the $\mathrm{l}_{\mathrm{game}}$ loss leads to the emergence of a communication protocol that is indecipherable to a human. However, the addition of the perceptual loss, motivated in Section 4.3.4, significantly improves the interpretability of the communication channel, as shown through the human evaluation discussed in Section 4.5, at almost no cost to the actual communication success rate.

One interesting observation is that although the sketches for some of the classes have greatly improved when incorporating the perceptual loss, for photographs of animals or birds, the sketches are not particularly representative of the class instance or distinguishable for the human eye. In the following sections, we explore the model to try to better understand what factors affect drawing production.

Table 4.1: **Communication success rate and example sketches produced by the agents in order to achieve the game objective in various setups and with different losses.** Sample input images seen by the sender (the left column) are described as the sketches in the second and third column. Although successful communication seems to be achieved in all setups, the addition of the perceptual loss significantly improves human interpretability of the drawings. Examples are from STL-10.

	l_{game}	$l_{\text{game}} + l_{\text{perceptual}}$
Original game	71.8% (±6.1)	69.57% (±2.6)
		
OO-game same	95.46% (±0.6)	96.04% (±0.5)
		
OO-game different	82.72% (±0.8)	81.09% (±0.6)
		

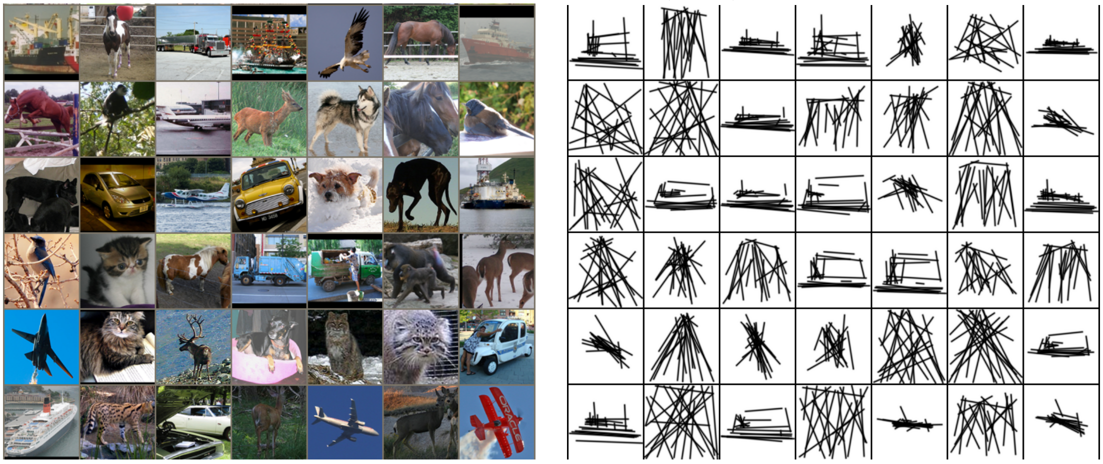
4.4.2 What does sketching under different game setups look like?

Table 4.2 illustrates more examples of sketches drawn under different game configurations. Clearly, some classes are better represented and more interpretable to a human than others.

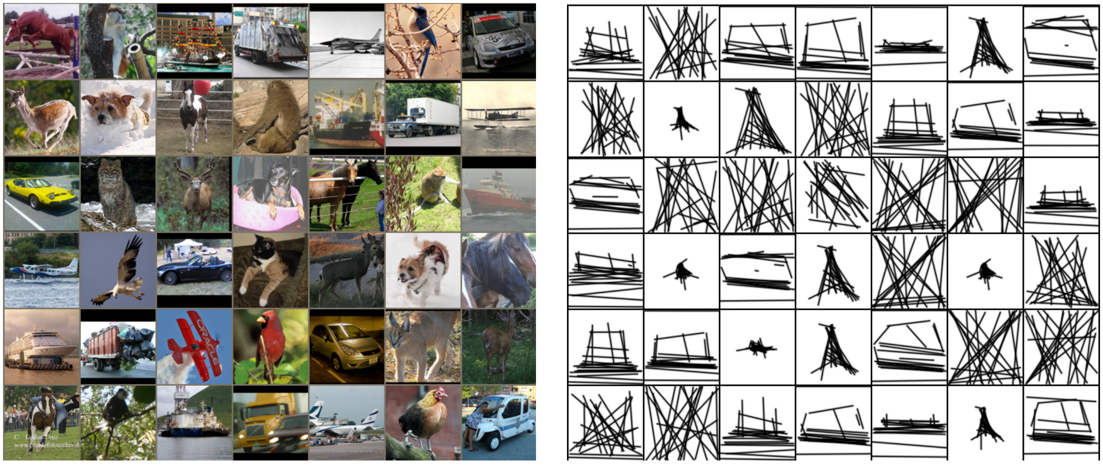
Further, in Figure 4.3, we provide an example of the full reference games to help the reader understand how difficult the original game setting, with 99 distractors, would be to play for a human receiver. This should shed some light on how "interpretable" the communication is in the full context given to the receiver agent, which may contain many perceptually similar distractors in the original setting. The game in either of the object-oriented settings shown in Figures 4.4 and 4.5, played on STL-10 classes, seems much more feasible to a human receiver.

TABLE 4.2: **More example sketches produced by the agents in the three different game setups using the $l_{\text{game}} + l_{\text{perceptual}}$ loss.** Examples are from STL-10.

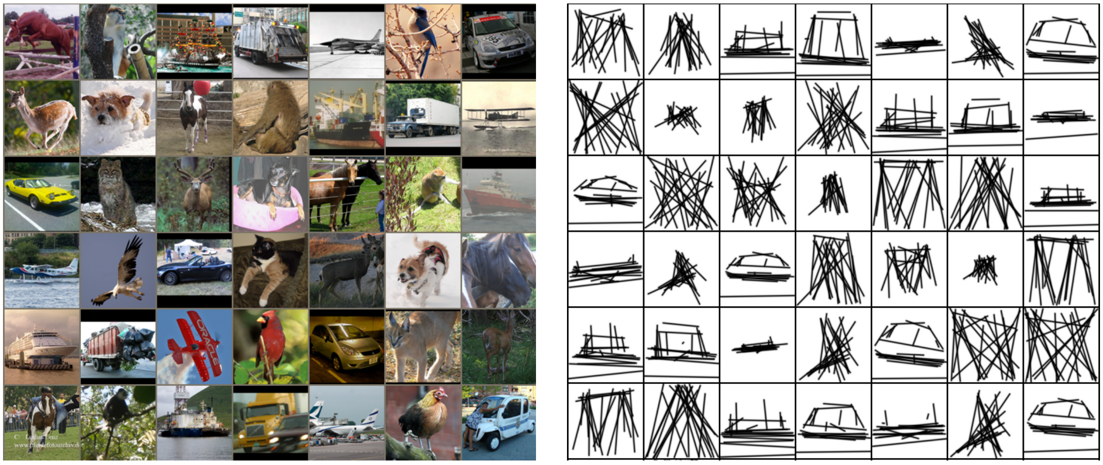
Original game: 69.57% (±2.6)



OO-game same: 96.04% (±0.5)



OO-game different: 81.09% (±0.6)



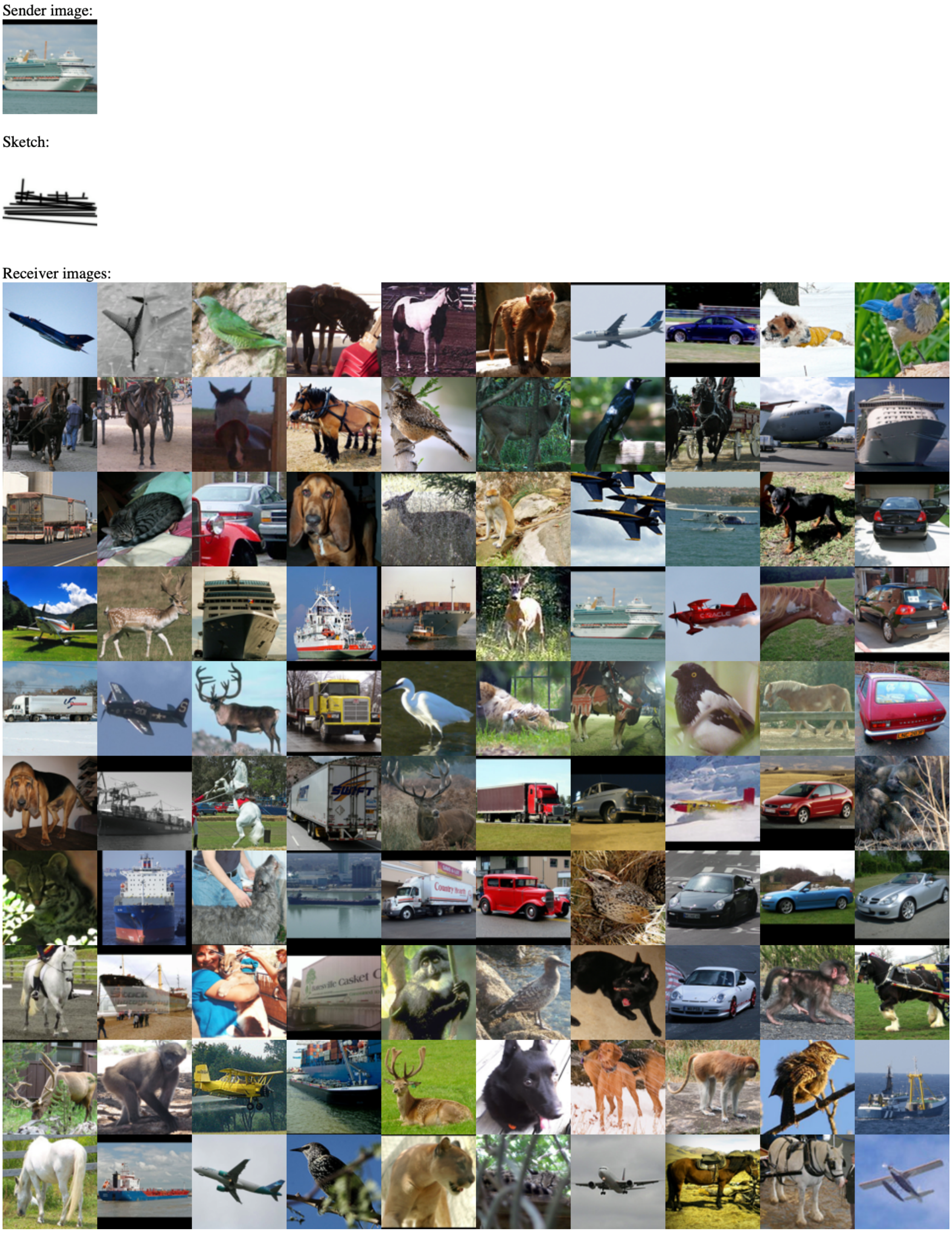


Figure 4.3: **Example of full reference game -** ***original*** **setting with 99 distractors.**

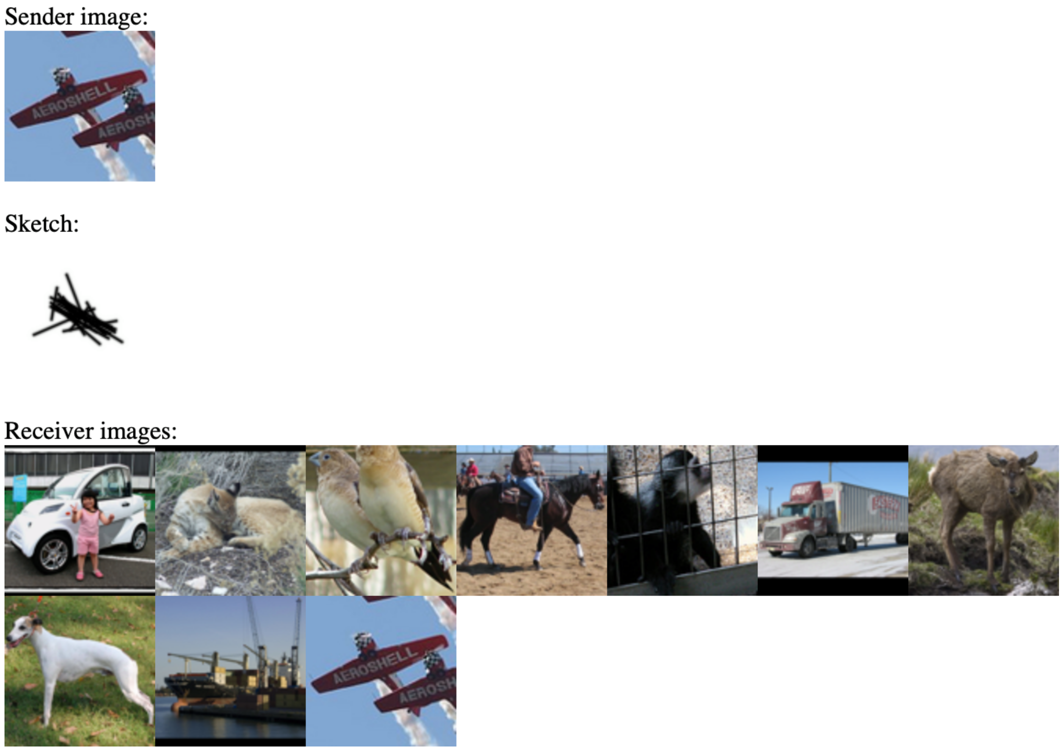


Figure 4.4: **Example of full reference game - *object-oriented same* setting** in which the sender's target is part of the set of images shown to the receiver.

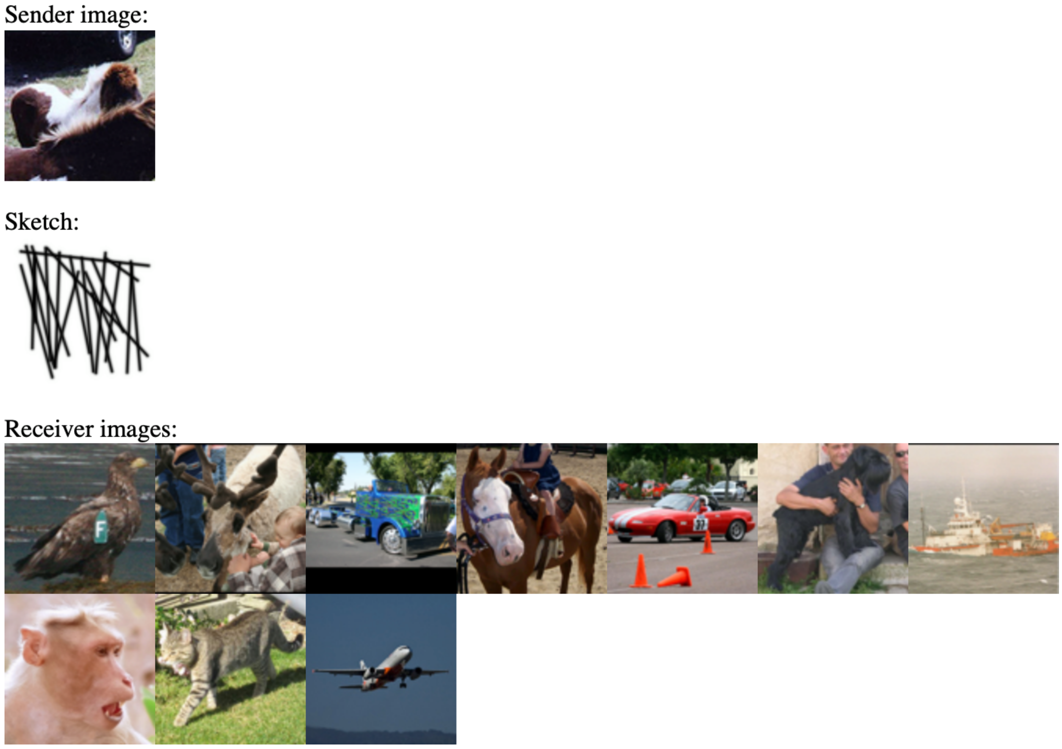


Figure 4.5: **Example of full reference game - *object-oriented different* setting**. The receiver's target is a different image that belongs to the same class as the sender's.

4.4.3 Does the *OO-game* influence the sketches to be more recognisable as the type of object?

Comparing the qualitative results of different game formats from Table 4.2, we notice that agents develop distinct strategies for representing the target photograph under different conditions. If there is more variability in the sketches that correspond to photographs from the same class in the original game setup, and a bit less in the *OO-game same*, the sketches become more like symbols representing all the photographs from one class when playing *OO-game different*. In other words, the object-oriented games influence the sketches to be more recognisable as the type of object, than the specific instance of the class.

Finally, it is worth noting how our results connect to how humans communicate through sketching when constrained under similar settings. The far/close contexts used by Fan et al. [2020] are somewhat equivalent to our original/object-oriented settings. As Fan et al. observe when humans play a similar drawing game, our agents achieve a higher recognition accuracy in settings that involve targets from different classes and develop different communication behaviours based on the context of the receiver.

4.4.4 How does the sketch complexity influence communication?

An interesting question one might ask about a model that learns to communicate by drawing is how complex the sketch image needs to be so that its meaning can be conveyed successfully and communication can be established. We attempt to answer this question by varying the number of lines that our model is allowed to draw to represent the input photograph. In Table 4.3, we show results for experiments run with 5, 10 and 20 lines allowed for sketching. As with previous experiments, we provide the communication success rate with a standard deviation over 10 seeds and qualitative results under two game setups. Under the original game format, contrary to what one might expect, the communication rate decreases as the number of lines is increased (see also Table 4.4). From a visual point of view, using more lines results in sketches that are more interpretable to a human observer as supported by the evidence discussed in Section 4.5, although that does not seem to correlate with the agent's communication strategy. Varying the complexity of drawings in the object-oriented game does not significantly influence the communication rate. The sketches, however, show once more that such a setup can induce a more interpretable communication channel. It is clear that even when drawing 5 lines, the model is trying to capture the overall shape of the object.

Further, we show results with an increased number of strokes, in the original game setting, in Table 4.4. Compared to the model trained to draw with 20 lines in the original game setting (see Table 4.3) which tries to cover the overall space occupied by the photograph's

Table 4.3: **The effect of the drawing complexity (5, 10 or 20 line strokes) on the emergent visual communication channel.** The communication success rate (*i.e.* receiver agent correctly guessing the target image) and standard deviation across 10 runs are shown next to sample sketches.

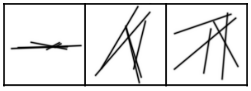
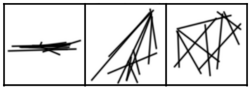
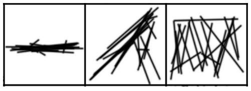
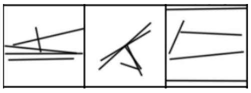
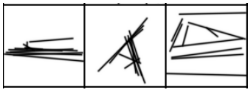
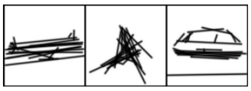
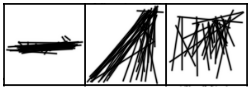
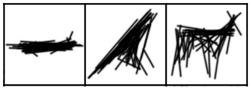
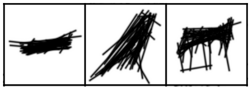
	5	10	20
Original game 	73.41% (±1.6) 	69.48% (±3.3) 	69.57% (±2.6) 
OO-game different 	80.69% (±1.1) 	80.9% (±0.6) 	81.09% (±0.6) 

Table 4.4: **The effect of increasing drawing complexity (30, 40 or 50 lines) in the original game setting.** Sketches become visibly more correlated with the input photographs as the increase in the number of line allows for shorter strokes to be used which help with the overall interpretability.

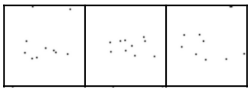
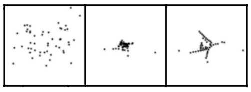
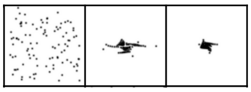
	30	40	50
Original game 	71.13% (±1.9) 	70.01% (±2.1) 	69.21% (±1.4) 

main object, the models trained with more strokes start to draw different lengths, and thus, the object becomes visually more recognisable.

4.4.5 How important is the rasteriser?

To further challenge our hypothesis about visual communication being possible between fully self-supervised agents, we ask the question of how important the rasteriser, and hence the sketch, is for the emergent communication protocol. Instead of line strokes, we constrain the agents to encode images into a cloud of points. We observe that communication between agents is definitely possible even when extracting as little as 10 points from an image, but the resulting sketch does not have any meaning to a human observer. When increasing the number of points to 50, or better 100, the communication success slightly drops to 0.71, 0.66 respectively, but object contours/shapes start to emerge in the sketches as shown in Table 4.5. Encoding to a cloud of points is possible but less efficient, as it requires more coordinates to be learned to create sketches that are interpretable to some extent for humans.

Table 4.5: **The effect of encoding the images into a cloud of points (10, 50, 100) in the original game setting.** Communication is possible with a points rasteriser, but more inefficient. More interpretable sketches require a larger number of points and, hence, more parameters to be learned.

Points	10	50	100
Original game	75%	71%	66%
			

4.4.6 How much do sketches differ visually across seeds?

Throughout this chapter, the sample sketches are presented from one seed out of the 10 model runs. However, in Figure 4.6 we present an example of an overlay of the 10 seeds, normalised to look like a heatmap so that darker lines represent strokes generated by *more* models. As can be seen, the 10 different models trained on Caltech-101 [1] from different seeds are consistent in picking out key features of the input image, but show variation in finer details.

4.4.7 What effect does weighting the perceptual loss have on the sketches?

Next, we explore the effect of manually weighting the perceptual loss. More precisely, we look at what happens when the perceptual loss is applied to the features maps from just one layer of the backbone network. As previously mentioned in Section 4.3.2, the feature maps are extracted using a VGG16 CNN up to `ReLU5_3` layer. For example, we can discard all feature maps except those from the first layer by weighting the perceptual loss by $[1, 0, 0, 0, 0]$. The effect of the different weights, which allow only one block of feature maps to be used for drawing the sketch, is illustrated in Table 4.6. We apply these constraints in two setups, the *original* and the *OO-game different*. In both cases, the drawings are unrecognisable if the perceptual loss takes into account only the first or the second block of feature maps. Blocks 3 through 5 seem to provide increasing structure under both game setups. It is worth noticing that, similar to the results shown in Section 4.4.1, the communication success rate in the original setup is always lower than that from the *OO-game different* setup. Overall, the information provided by individual layers in the visual extractor network is enough for the agents to develop a visual communication strategy that can be used to play the game. For humans, however, the later layers contribute the most to the emergence of a communication protocol that humans can understand.

[1] It is worth mentioning that for this experiment, as well as for other questions such as those posed in Sections 4.4.9 and 4.4.10, results are presented on the Caltech-101 dataset instead of STL-10 because it has a greater variety of classes and the images have a higher resolution which makes it more appropriate for highlighting the findings.

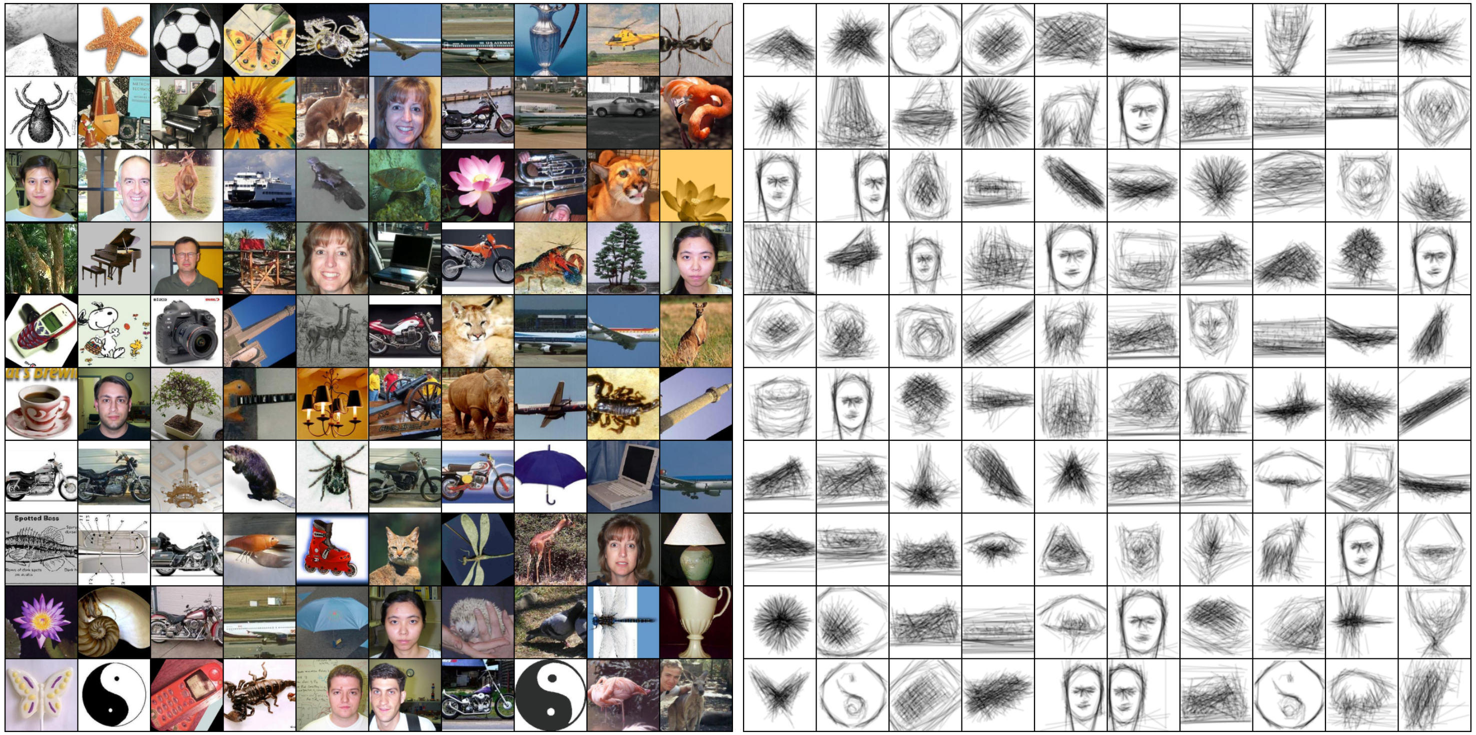
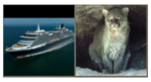
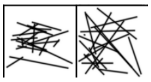
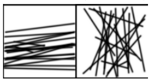
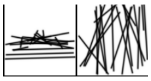
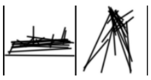
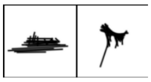
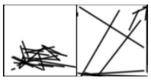
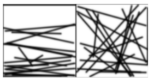
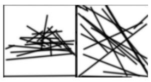
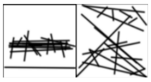
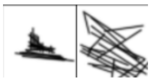


FIGURE 4.6: **An overlay of 10-seeds sketches** drawn by a model trained in the *original* game variant on Caltech-101, with Stylized-ImageNet weights.

Table 4.6: **The effect of weighting the perceptual loss such that only the feature maps from one backbone layer are used.** The features extracted in the last three layers of the visual system seem to capture information that leads to sketches which resemble to an extent the corresponding photograph.

Loss weights	$[1, 0, 0, 0, 0]$	$[0, 1, 0, 0, 0]$	$[0, 0, 1, 0, 0]$	$[0, 0, 0, 1, 0]$	$[0, 0, 0, 0, 1]$
Orig. game 	68.4% (±3.6) 	69.6% (±2.2) 	71.1% (±2.4) 	76.4% (±2.1) 	60.5% (±4.8) 
OO-game diff 	81.9% (±1.2) 	81.5% (±0.9) 	82.3% (±0.9) 	82.5% (±0.5) 	81.4% (±0.8) 

4.4.8 What is the impact of L in the computation of the perceptual loss on the emergent sketches?

When computing the additional perceptual loss to induce sketches to become visually more similar to the target photographs, we use the outputs of $L = 5$ feature maps, extracted from the VGG16 layers immediately before the max-pooling layers (`relu1_2`, `relu2_2`, `relu3_3`, `relu4_3` and `relu5_3`), we will refer to this set of feature maps as $fmaps$. Table 4.7 shows the effect of decreasing L and using feature maps only up to the specified layer. More concretely, in the table results for `relu4_3` show how the sketches look like when the perceptual loss is computed over features extracted after `relu1_2`, `relu2_2`, `relu3_3`, `relu4_3` only. We perform this ablation study in the original game setting with 20 line sketches, and show qualitative examples, the communication success rates averaged over 10 seeds and standard deviations. Note that these results are from when L is changed for both sender and receiver agents. We observe that there is a drastic drop in the communication success rate as L decreases from 5 to 4. Even more, if the perceptual loss is computed over the features extracted up to the third block of the VGG16 extraction network (*i.e.* anything up to `relu3_3`), the model no longer converges and the communication completely fails.

Similarly, Table 4.8 shows the effect of increasing L. To the original set of feature maps ($fmaps$) used in the computation of the perceptual loss, the outputs of the other convolutional layers in a certain block (5, 4 or 3) of the VGG16 feature extraction network are added. The results show that increasing the number of feature maps neither impacts the communication success rate nor makes the sketch visually more similar to the corresponding image.

Table 4.7: **Ablation study on the number of feature maps extracted from the visual system.** Studying the effect of decreasing the number of feature maps (L) extracted from the backbone VGG16 network. We present results by using features extracted from layers in *fmaps* up to the specified layer.

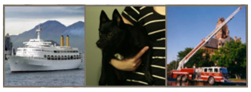
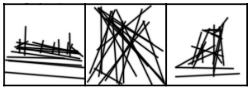
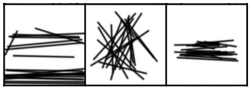
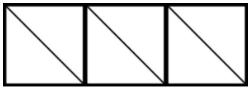
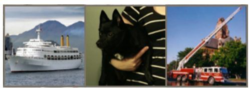
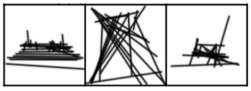
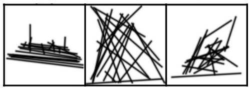
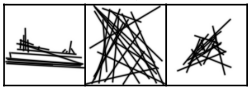
	`relu5_3`	`relu4_3`	`relu3_3`
Original game	69.57% (±2.6)	19.09% (±10.1)	1.0% (±0)
			

Table 4.8: **Ablation study II on the number of feature maps extracted from the visual system.** Studying the effect of **increasing** the number of feature maps (L) extracted from the backbone VGG16 network. To the original set of feature maps, *fmaps*, we add the following extra layers: `reluX_*` indicates that the other feature maps from the X^{th} block of convolutions in the VGG16 feature extraction network are being used to compute the perceptual loss.

	`relu5_*`	`relu5_*`, `relu4_*`	`relu5_*`, `relu4_*`, `relu3_*`
Original game	68.4% (±2.0)	69.5% (±1.8)	69.0% (±1.2)
			

4.4.9 How does the model's capacity influence the visual communication channel?

Regarding the model's architecture, we look into how drawings are influenced by the width of the model. In this experiment (results shown in Table 4.9), we compare the baseline model architecture detailed in Section 4.3.2 with a wider variant that has the following changes: the sender encodes the target photograph to a 1024-dimensional vector (baseline model encodes to 64-dimensional vector), the receiver's MLP capacity is also increased from 64 to 1024 in both layers. We present results for the *OO-game different* setup played with 128 × 128 Caltech-101 images [Fei-Fei et al., 2004]. The increased number of classes in Caltech-101 may explain the drop in the communication rate in this particular game setting, which compared to the same model played under the original game setup (see the ImageNet-pretrained model in Table 4.10), is with almost 30% lower. As one might expect, the wider model allows for more details to be captured, and, hence, conveyed in the sketches. Unlike the baseline model which, in this object-oriented setup, develops a communication system that is more representative of the class than of the instance (as discussed in Section 4.4.3), the wider model starts to draw distinctive representations for objects of the same type. For example, in Table 4.9 one can observe the difference between all images with chairs.

Table 4.9: **The effect of the model's capacity on its sketches.** Examples from training on 128 × 128 pixel Caltech-101 images, in the *OO-game different* setting. The wide model's sender encodes the photo into a 1024-dimensional vector (baseline 64), and the receiver's MLP linear layers have 1024 neurons each versus 64.

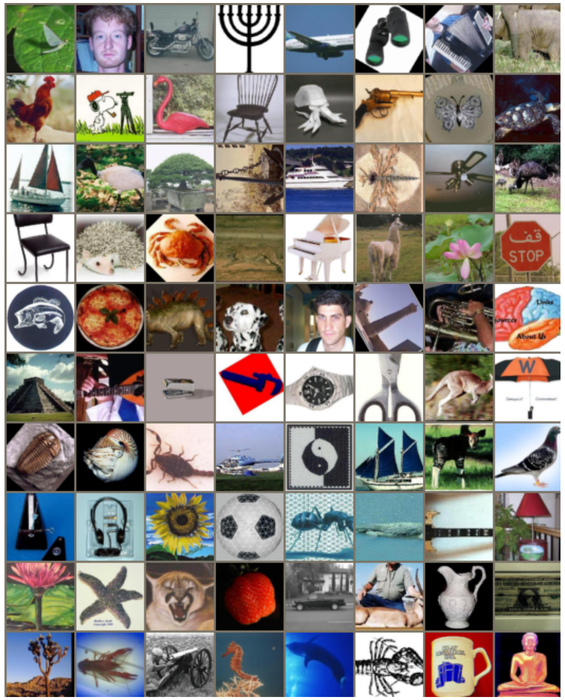
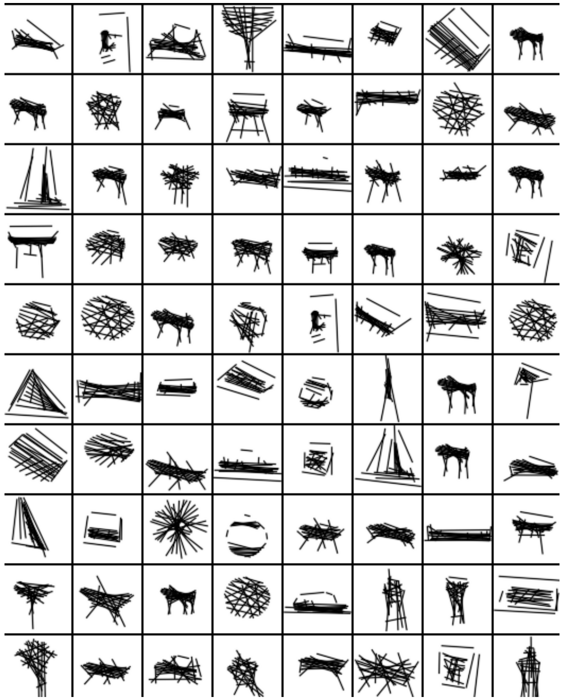
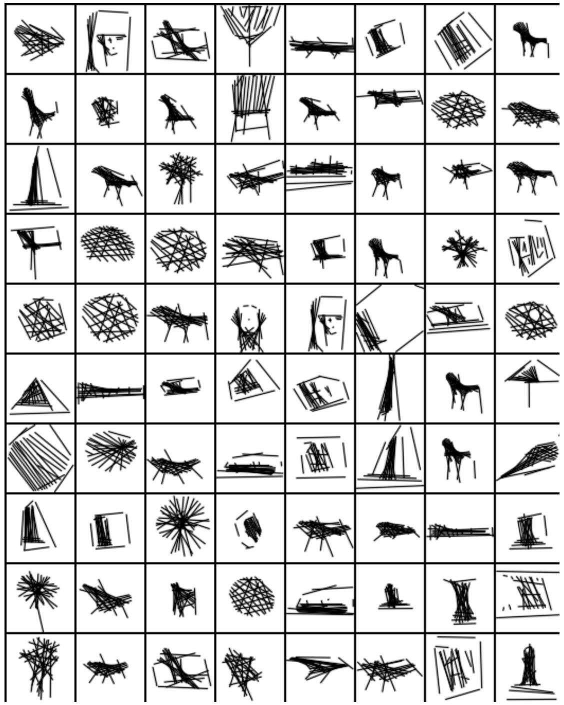
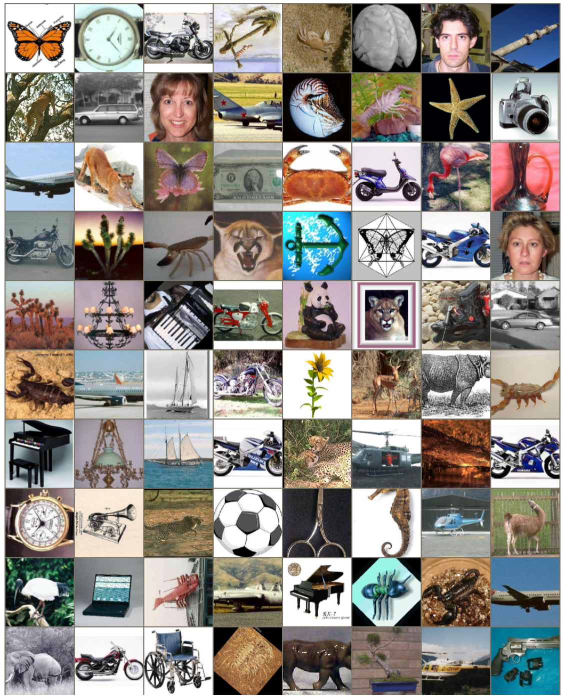
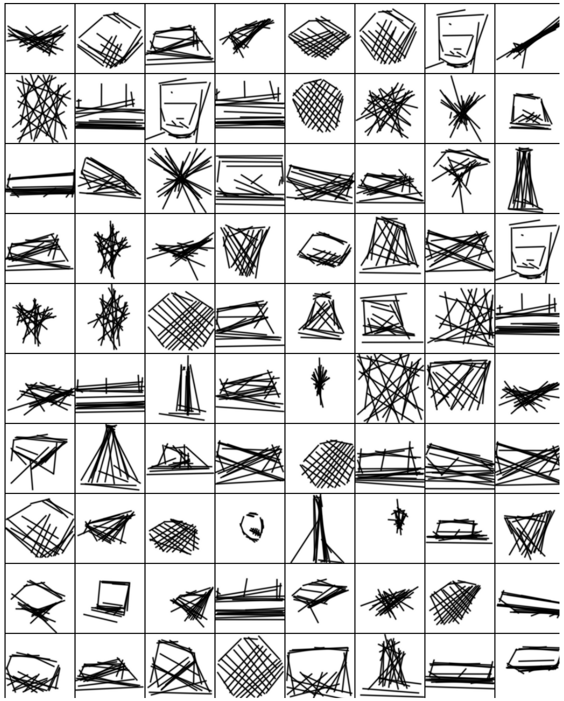
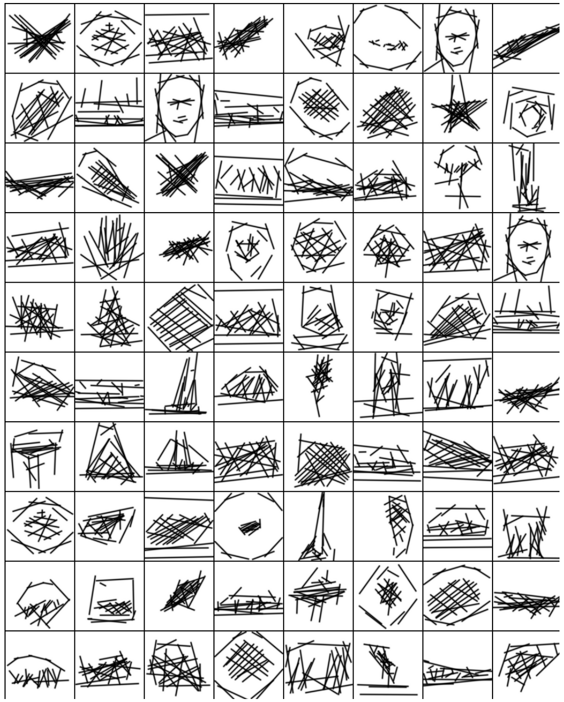
	Baseline	Wide
	50.46% (±1.5)	64.99% (±1.5)
		

Table 4.10: **The effect on the communication protocol of using a VGG16 feature extractor network pretrained on datasets that have texture (ImageNet) or shape (Stylized-ImageNet [Geirhos et al., 2019]) bias.** Examples are from agents trained in the *original* game setup with 128 × 128 Caltech-101 images. Shape-biased sketches are, visually, more similar to the objects they represent and are better at capturing the overall object form, particularly for things like faces.

	ImageNet weights	Stylized-ImageNet weights
	78.46% (±2.0)	77.09% (±1.9)
		

4.4.10 How does the texture/shape bias of the visual system alter communication?

Next, we show that a texture or shape bias of the visual system influences visual communication. This experiment was run under the original game setup with 128 × 128 Caltech-101 images [Fei-Fei et al., 2004]. The results shown in Table 4.10 suggest that inducing a “shape bias” into the model does not significantly improve the agent’s performance in playing the game, but produces more meaningful drawings. By using the VGG16 weights pretrained on Stylized-ImageNet [Geirhos et al., 2019], the communication protocol also becomes more faithful to the actual shape of the objects. A shape-based sketch is much more interpretable to humans, as it has been known for a long time that shape is the most important cue for human object recognition [Landau et al., 1988].

4.4.11 Do the models learn to pick out salient features?

From the results we have presented so far, it is evident that, particularly with the perceptual loss, the sender agent is able to broadly draw pictures of particular classes of objects. The high communication rates in the *original* game setting would also suggest that the drawings can capture something *specific* about the target images that allow them to be identified amongst the distractors. To further analyse what is being captured by the models we train the agents in the original game setting (using both normal and stylized backbone weights) with images from the CelebA dataset [Liu et al., 2015], which we take the maximal square centre-crop and resize to 112px. As this dataset contains only images of faces, messages between the agents will have to capture much more subtle information to distinguish the target from the distractors. Figure 4.7 shows the results;

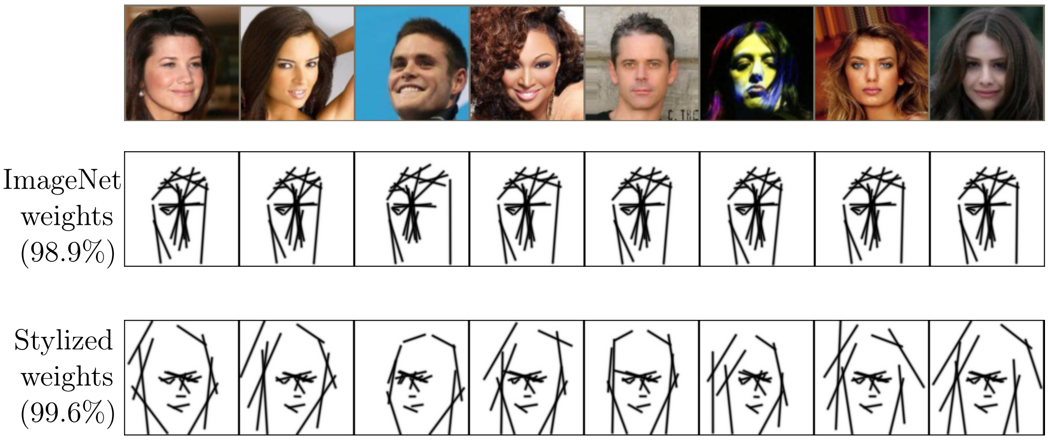


Figure 4.7: **Sketches from *original* variant games using the CelebA dataset with perceptual loss and different biases from backbone weights.** Both the texture-biased (ImageNet) and shape-biased (Stylized-ImageNet) settings exhibit near-perfect communication success, but the shape-biased sketches are considerably more interpretable and show visual variations correlated with the photos.

the communication rate is near perfect for both models, but the difference between the texture-biased and shape-biased models is striking. There is subtle variation in the texture-biased model's sketches which broadly seems to capture the head pose, but the overall sketch structure is similar. In the shape-biased model head pose is evident, but so are other salient features like hairstyle and (see Figure 4.8) head-wear and glasses.

4.4.12 What happens if the communication is constrained under an arbitrary, meaningless objective?

One might ask what happens to the communication protocol when the perceptual loss is replaced with some meaningless, arbitrary objective. To explore this scenario, we constrain sketches to look like a single image of a dog (shown in the top left of Figure 4.9) and train agents to draw in order to communicate about CelebA images. As one might expect, the artificial agents can still establish a successful communication strategy about the correct target even when constrained to draw dog-like sketches. Figure 4.9 shows results for models trained with such an additional objective, fully or partially, by scaling λ in $l = l_{game} + \lambda l_{arbitrary}$. These results indicate that it matters what the perceptual loss is: if it constrains sketches to look like the corresponding photographs, a human receiver might have a chance at recognising the person, but with such an arbitrary objective, humans stand no chance at understanding which image the sender agent tries to communicate about. Agents' communication success rate is also impacted (compared to the model with Stylized weights trained with our $\mathrm{l}_{\mathrm{perceptual}}$ and $\lambda = 1$, results shown in Figure 4.7).

4.4.13 What happens when injecting out-of-distribution images?

To further investigate the emergent visual communication protocol, we test a pair of agents pretrained in the proposed framework on out-of-distribution images. More specifically, we evaluate a pair of agents, previously trained in the *original* setting on CelebA dataset, on games played with STL-10 images (see Figure 4.10). Agents with ImageNet-pretrained visual systems, achieve a test communication rate on STL-10 of 15.8%. Similarly, agents initialised with Stylized-ImageNet weights achieve 30% test recognition accuracy. It is worth noting that even if these results are significantly lower, they are still better than random chance, particularly with the Stylized-ImageNet weights, where the sketches have considerably more diversity (but still all look like faces rather than the objects in the images).

A similar experiment is performed with models pretrained on STL-10, with either just the $\mathrm{l}_{\mathrm{game}}$ or with the additional $\mathrm{l}_{\mathrm{perceptual}}$. When testing these on Caltech-101 test data as shown in Figure 4.11, the communication success drops to 22.2% and 26.7% respectively. Interestingly, the perceptual loss helps improve generalisability in this case.

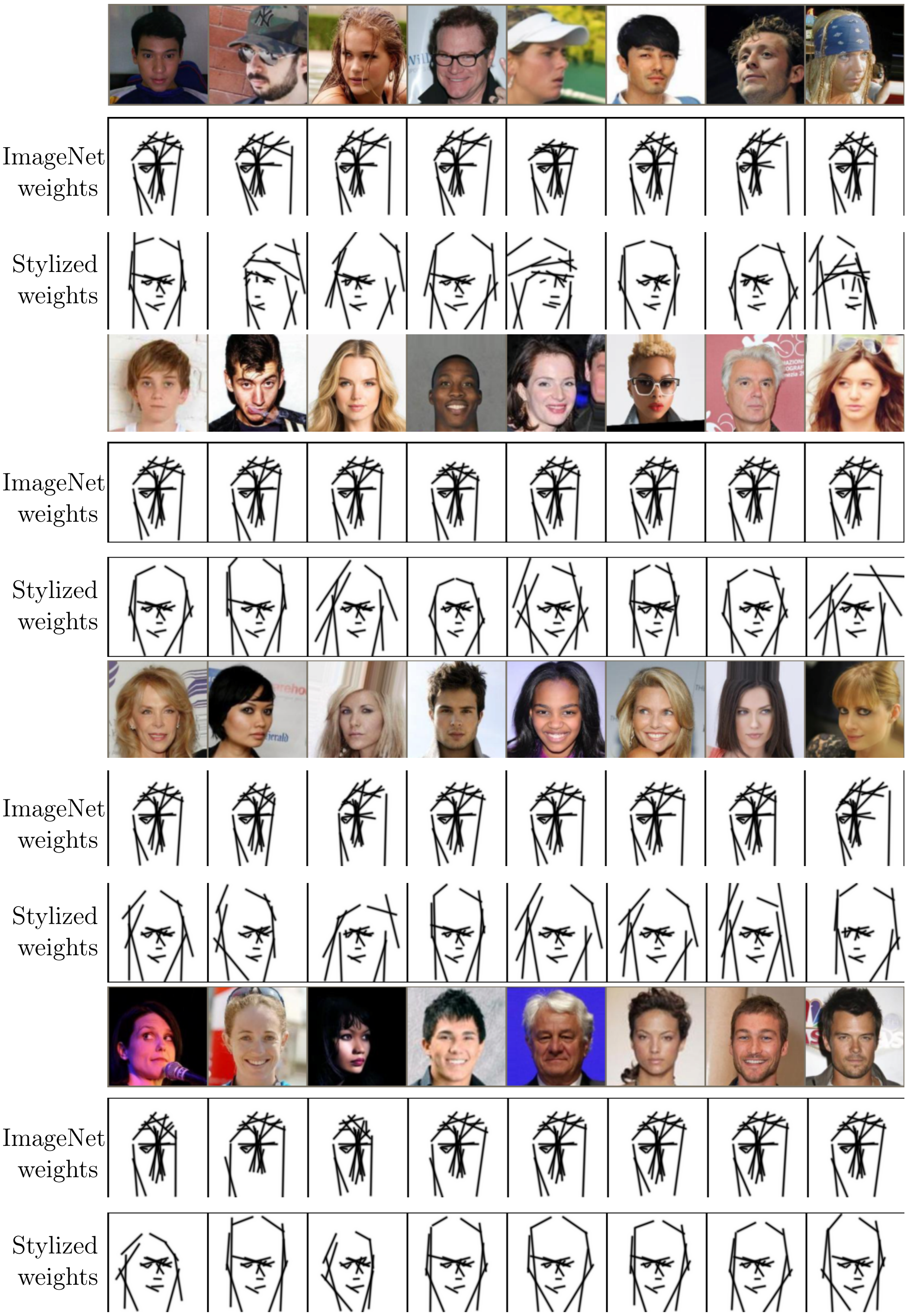


FIGURE 4.8: **Additional sketches from *original* variant game using the CelebA dataset, the perceptual loss and different biases from backbone weights.** Although both models have near perfect communication success, it is clear the inducing a shape bias helps bring out the most salient and distinctive features.

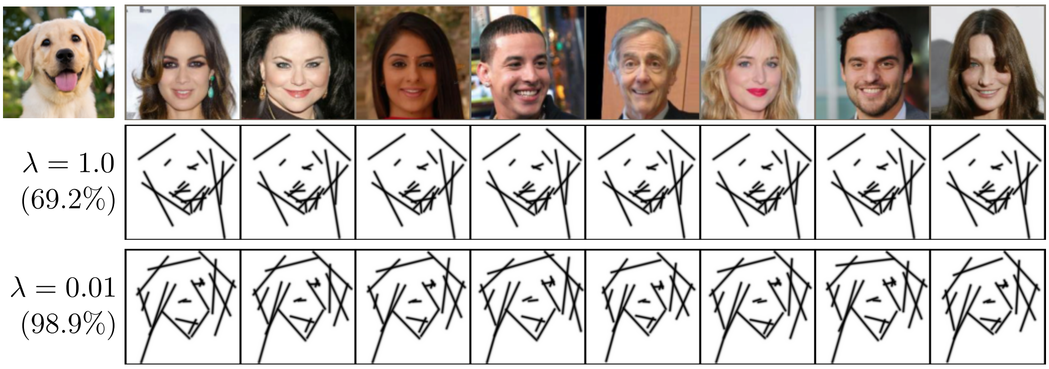


Figure 4.9: **Sketches from *original* variant game using the CelebA dataset with an arbitrary objective: the sketches are constrained to look like the image of a dog (fully or partially, by scaling the perceptual loss coefficient λ).** Results are shown for a model with the visual extraction network pretrained on Stylized-ImageNet.

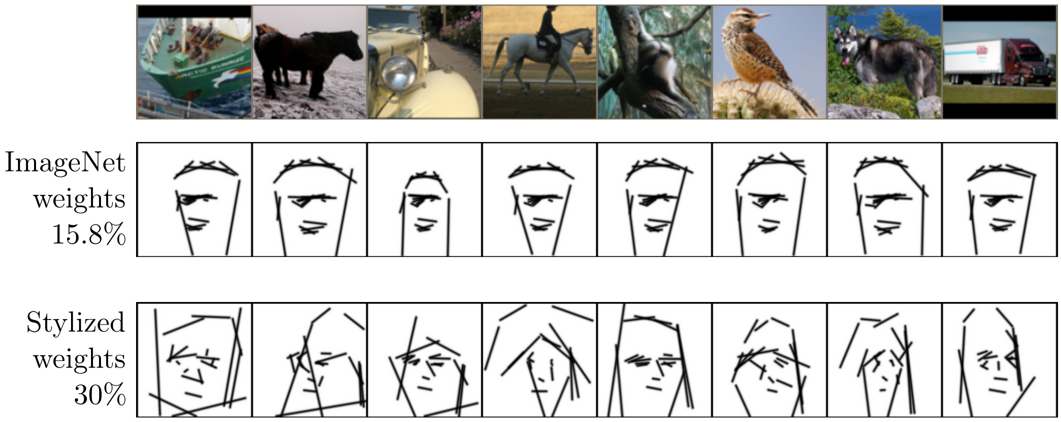


Figure 4.10: **Sketching agents, previously trained on CelebA (original game) tested on STL-10 test images.** We compare models with visual systems pretrained on ImageNet and Stylized-ImageNet.

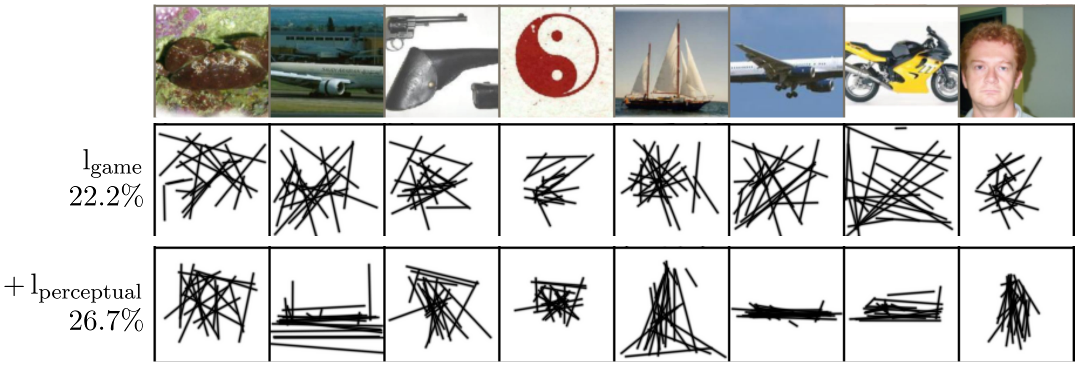


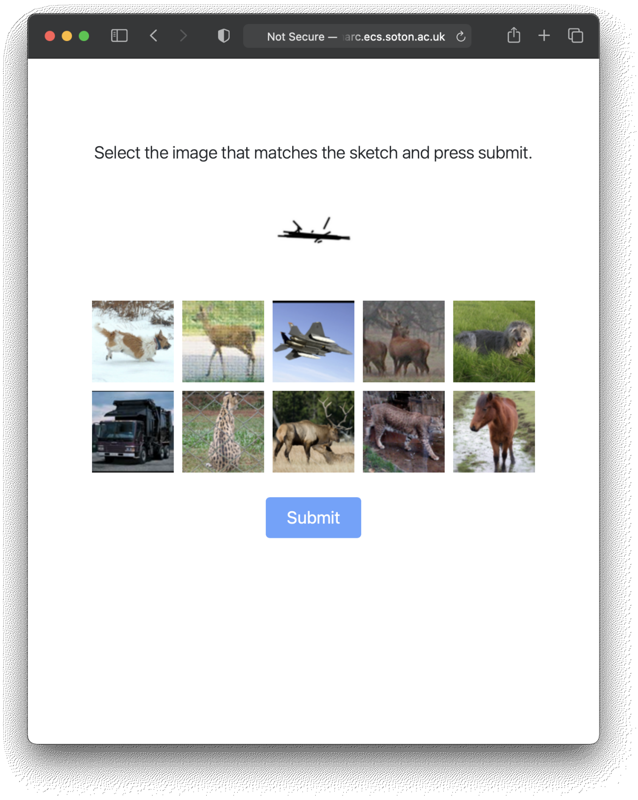
Figure 4.11: **Sketching agents, previously trained on STL-10 (original game) tested on Caltech-101 test set.** We compare models pretrained with l_{game} only with those that also use $l_{perceptual}$.

4.5 Do Agents Learn to Draw in a Fashion that Humans can Interpret?

In order to assess the interpretability of sketches drawn by artificial agents, we set up a pilot study in which a ‘sender’ agent, pretrained in five different game configurations on STL-10, is paired up with a human ‘receiver’ to play the visual communication game. For this pilot study, we collect results from 6 human participants. Each participant played a total of 150 games, *i.e.* had to select the target image for each of the 150 sketches drawn by a pretrained sender. Depending on the game setting, the list of options differs, but it is composed of distractors and the true target image. The experimental setup is detailed in Section 4.5.1 and results are shown in Section 4.5.2.

4.5.1 The experimental setup

The task. To reiterate the experimental task, the human participant is shown a sketch (previously generated by a trained sender agent during model evaluation) and is asked to select by clicking the corresponding target image from a grid of images, as illustrated in Figure 4.12.



Select the image that matches the sketch and press submit.

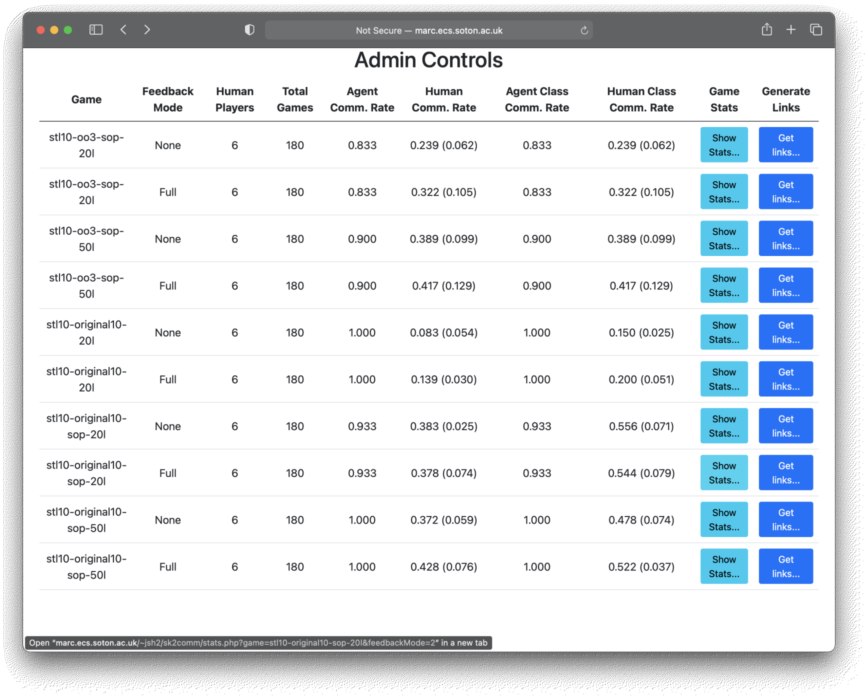
Figure 4.12: **Example of a game, original setting - the human participant has to pick from 10 images.**

The data. The sketches used in this experiment are generated in five different game configurations, varying the game setup, agents' training objective and the number of strokes. For the purpose of this study, the sketches are generated by models trained with the same fixed random seed. Whilst there is inevitably some variation in models from different seeds (see Section 4.4.6), this is not explored in the human evaluation.

For each game setting, the participant played 30 games, matching a total of 30 sketches to different target image sets. Each human participant played a total of 150 games, and the total amount of data collected in the pilot study corresponds to 1800 games. For this study, the games were chosen randomly from all those possible within the STL-10 test dataset. For all game settings used in the human pilot study, we limit the number of distractors to $K = 9$.

User interface. To allow human participants to play the game, a web interface was developed and each participant was provided with a set of 5 unique URLs corresponding to the 5 different game settings. Information on what the different settings involved was not provided to the participants. Each URL took the participant through 30 games and stored their answers in a database.

An example of such a game is shown in Figure 4.12. We do not impose a time limit per game but record how much time the participants take to make their guesses. Figure 4.13 shows our admin interface which summarises the averaged statistics based on the games played in this pilot study. Figure 4.14 shows the interface when feedback is given (see Section 4.5.3).

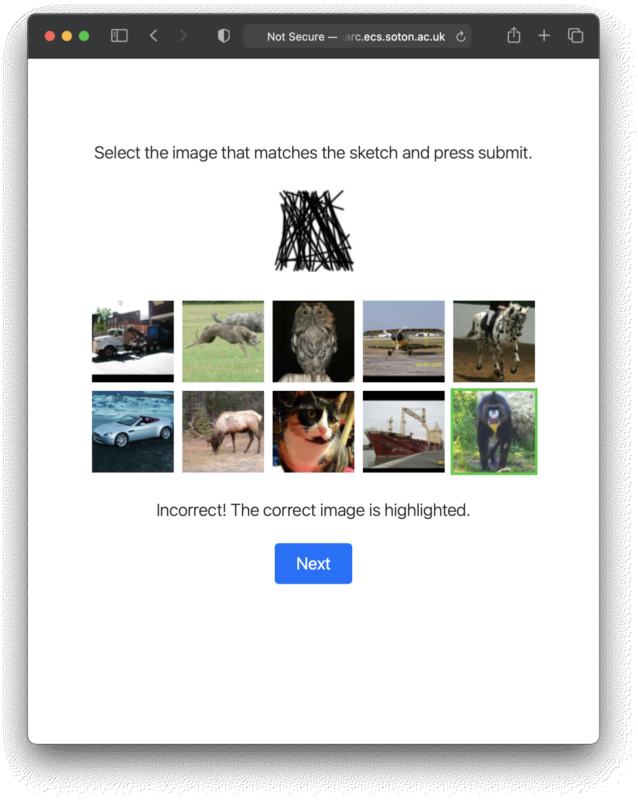


Admin Controls

Game	Feedback Mode	Human Players	Total Games	Agent Comm. Rate	Human Comm. Rate	Agent Class Comm. Rate	Human Class Comm. Rate	Game Stats	Generate Links
stl10-oo3-sop-20l	None	6	180	0.833	0.239 (0.062)	0.833	0.239 (0.062)	Show Stats...	Get links...
stl10-oo3-sop-20l	Full	6	180	0.833	0.322 (0.105)	0.833	0.322 (0.105)	Show Stats...	Get links...
stl10-oo3-sop-50l	None	6	180	0.900	0.389 (0.099)	0.900	0.389 (0.099)	Show Stats...	Get links...
stl10-oo3-sop-50l	Full	6	180	0.900	0.417 (0.129)	0.900	0.417 (0.129)	Show Stats...	Get links...
stl10-original10-20l	None	6	180	1.000	0.083 (0.054)	1.000	0.150 (0.025)	Show Stats...	Get links...
stl10-original10-20l	Full	6	180	1.000	0.139 (0.030)	1.000	0.200 (0.051)	Show Stats...	Get links...
stl10-original10-sop-20l	None	6	180	0.933	0.383 (0.025)	0.933	0.556 (0.071)	Show Stats...	Get links...
stl10-original10-sop-20l	Full	6	180	0.933	0.378 (0.074)	0.933	0.544 (0.079)	Show Stats...	Get links...
stl10-original10-sop-50l	None	6	180	1.000	0.372 (0.059)	1.000	0.478 (0.074)	Show Stats...	Get links...
stl10-original10-sop-50l	Full	6	180	1.000	0.428 (0.076)	1.000	0.522 (0.037)	Show Stats...	Get links...

Open "marc.ecs.soton.ac.uk/~jsh2/sk2comm/stats.php?game=stl10-original10-sop-20l&feedbackMode=2" in a new tab

FIGURE 4.13: **The admin interface.**



Select the image that matches the sketch and press submit.

Incorrect! The correct image is highlighted.

FIGURE 4.14: **Example of the game played with feedback.**

Participants. We divide the human evaluation into two disjoint study groups: participants who just play the game with no feedback and, hence, cannot learn during gameplay (results are presented in Table 4.11), and a second group which is allowed to learn from feedback. Details about the latter group are discussed in Section 4.5.3.

For the purpose of the study, we collect results from 6 participants per group. Overall, the study includes participants aged between 20 to 35 with various professions. Participation in the study does not require any specific skills.

4.5.2 Results

Table 4.11 compares the averaged human gameplay success to that of a trained 'receiver' agent. The results show that the addition of the perceptual loss leads to a statistically significant improvement in humans' ability to recognise the identity of sketches. For the original game setting, played in this study with $K = 9$ distractors which might be of the same category as the target, we also assess the ability of participants to recognise the class of the sketch. The human class communication rate shows that humans are better at determining the class of the sketch rather than the specific instance, even in the case of sketches generated with the game loss only.

Table 4.11: **Human Evaluation results, no learning allowed.** Trained agents communicate successfully between themselves in all settings. The addition of the perceptual loss allows humans to achieve significantly better than random performance (images from STL-10, original games have 9 distractors/game for these experiments & random chance is 10%). In addition, humans are better at guessing the correct image class when the models are trained with the additional perceptual loss.

Game	Loss	Lines	Agent comm. rate	Human comm. rate	Human class comm. rate
original	$l = l_{game}$	20	100%	8.3% (±5.4)	15.0% (±2.5)
original	$l = l_{game} + l_{perceptual}$	20	93.3%	38.3% (±2.5)	55.6% (±7.1)
original	$l = l_{game} + l_{perceptual}$	50	100%	37.2% (±5.9)	47.8% (±7.4)
oo diff	$l = l_{game} + l_{perceptual}$	20	83.3%	23.9% (±6.2)	23.9% (±6.2)
oo diff	$l = l_{game} + l_{perceptual}$	50	90.0%	38.9% (±9.9)	38.9% (±9.9)

4.5.3 Can human participants *learn* to play the game?

The principal pilot study (Table 4.11) looks at humans' ability to play the game with an agent, with no feedback involved. The human participants will not know what the correct target was or if they guessed correctly. We also pose a slightly different question: Can humans learn to play the game with an agent? For this secondary study, after participants select what they believe to be the target image, they will be told if their selection was correct or not and the correct target will be indicated (as shown in Figure 4.14).

Table 4.12 summarises the statistics computed over the participants in this secondary study. The participants were tested on the same set of games as the first group, and the same metrics are reported.

T-tests run between the averaged communication success rates of the same game setting in the group with feedback versus the one without feedback, do not show a statistically significant improvement when participants are allowed to learn from feedback, except for the original game with $l = l_{game}$ only. As we expected, participants in both study groups had the lowest scores in this game across all tested settings: without feedback the averaged $commrate = 8.3\%(\pm 5.4)$; with feedback, $commrate = 13.9\%(\pm 3.0)$. The sketches drawn by a sender agent pretrained without the perceptual loss are not "constrained" to resemble the target image, hence they are the least interpretable. However, the two-tailed P-value

Table 4.12: **Human Evaluation results, learning allowed from feedback.**

Game	Loss	Lines	Agent comm. rate	Human comm. rate	Human class comm. rate
original	$l = l_{game}$	20	100%	13.9% (±3.0)	20.0% (±5.1)
original	$l = l_{game} + l_{perceptual}$	20	93.3%	37.8% (±7.4)	54.4% (±7.9)
original	$l = l_{game} + l_{perceptual}$	50	100%	42.8% (±7.6)	52.2% (±3.7)
oo diff	$l = l_{game} + l_{perceptual}$	20	83.3%	32.2% (±10.5)	32.2% (±10.5)
oo diff	$l = l_{game} + l_{perceptual}$	50	90.0%	41.7% (±12.9)	41.7% (±12.9)

between the two groups' performance in this setting was less than 0.0001 which suggests that feedback can lead to a statistically significant improvement when the sketches are not visually interpretable. Still, this is by far the worst communication scenario. This is also indicated by the amount of time the participants spent on average on this game which is higher than in other settings, the majority taking between 1 minute and 2 minutes 30 seconds per sketch.

In the future, it would perhaps be interesting to explore if humans could learn with feedback if they were to play more games; the 30 games per setting used in this experiment is possibly too little to allow a human player to robustly learn the strategy used by the agent.

4.5.4 Does the addition of the perceptual loss give statistically significant improvement over games which use only the hinge loss?

All participants were asked to play the original game with 20 stroke-sketches produced when $l = l_{game}$ and also when $l = l_{game} + l_{perceptual}$. Performing t-tests between the averaged communication rates within each study group, with and without perceptual loss, resulted in P values less than 0.0001, which indicates that the perceptual loss leads to a statistically significant improvement in humans' ability to play the game with the agent.

4.5.5 Does the number of strokes influence human performance?

We tested the original game and the object-oriented game setup, each with 20 and 50 strokes. The results indicate that in both settings, a higher number of strokes leads to better communication. However, in the group without feedback (Table 4.11), the mean communication rate was similar for the original setting with 20 strokes and with 50 strokes. The same game setting tested by people with feedback, however, showed a small increase in overall communication success. One should take into account that in this game setting, the human player might have to choose between more images from the same class. For an artificial agent, this can be an easy task. However, we might envisage a scenario in which other characteristics of a drawing would be included, such as colour, which might help the human differentiate between multiple instances from the same class. For example, think of 3 different species of birds, which could all be represented by some very general sketch, but could become distinctive if the colour were to be included. In the object-oriented game setting, the gap between 20-stroke and 50-stroke games is a bit more significant for both study groups.

4.5.6 Are humans better at determining the broader class of a sketch than at recognising the specific instance?

In the original game setting, it is possible to encounter distractor images from the same class as the target. In addition to the communication rate measure, which shows the overall success of an agent (human in this case) selecting the correct target image, we also compute the class communication rate, which calculates the overall success of an agent selecting an image from the same class as the true target. T-tests run between human communication rate and human class communication rate in the original game settings showed a statistically significant difference in both study groups. Humans are significantly better at understanding the broad class than they are at determining a specific instance based on the sketch in the games where there are multiple targets of the same class. This effect is possibly weakened by an increase in the number of strokes, however, as evidenced by a consistent lowering of statistical significance.

4.6 Summary

In this chapter, we have demonstrated that it is possible to develop and study an emergent communication system between agents where the communication channel is visual. Further, we have shown that a simple addition to the loss function (that is motivated by biological observations) can be used to produce messages between the agents that are directly interpretable by humans.

The immediate next steps in this line of work are quite clear. It is evident from our experiments that the incorporation of the perceptual loss dramatically helps produce more interpretable images. One big question to explore in the future is to what extent this is influenced by the original training biases of the backbone network — are these drawings produced as a result of the original labels of the ImageNet training data, or are they in some way more generic than that? We plan to address this by exploring what happens if the weights of the backbone are replaced with ones learned through a self-supervised learning approach like Barlow Twins [Zbontar et al., 2021]. We would also like to explore what happens if the agents' visual systems had independent weights.

Going further, as previously mentioned, learning a perceptual loss would be a good direction to explore, but perhaps this should also be coupled with a top-down attention mechanism based on the latent representation of the input. An open question from doing this would be to ask if this allows for a richer variation in drawing, and for features to be exaggerated as in the case of a caricature. Such an extension could also be coupled with a much richer approach to drawing, with variable numbers of strokes, which are not necessarily constrained to being straight lines. Coupling feedback or attention into the drawing mechanism itself could also prove to be a worthy endeavour.

We hope that the research presented in this chapter lays the groundwork for more study in this space. Fundamentally our desire is that it provides the foundations for exploring how different types of drawing and communication — from primitive drawings through to pictograms, to ideograms and ultimately to writing — emerge between artificial agents under differing environmental and internal constraints and pressures. Unlike other work that 'generates' images, we explicitly focus on learning to capture *intent* in our drawings. We recognise however that our approach may have broader implications beyond just understanding how communication evolves. Could for example in the future we see a sketching agent replace a trained illustrator? In the domain of robot art, Pix18 [Lipson, 2016] is a trailblazer as it is not only a robot that paints oil on canvas but can also conceive its own art subject with minimal human intervention. The creation of messages for communication inherently involves elements of individual creative expression and adaption to the emotive environment of both the sender and receiver of the message. Our current models are clearly incapable of this, but such innovations will happen in the future. When they do we need to be prepared for the surrounding ethical debate and discussions about what constitutes 'art'.

Chapter 5

How do Different Biases Affect Human Interpretability and Intent?

"Drawing at its best is not what your eyes see but what your mind understands."

— Millard Sheets

"It is ten per cent how you draw and ninety per cent what you draw."

— Andrew Loomis

This thesis has so far explored emergent communication through a discrete token-based communication bottleneck (Chapter 2), then proposed a differentiable rasterisation method (Chapter 3) which can be used to model self-supervised artificial agents that learn to communicate through parameterised strokes (Chapter 4). This chapter extends the study presented in Chapter 4 and further explores the effect of different perceptual losses and visual encoders on making sketches produced by a drawing model more interpretable to a human observer. We replace the VGG16 feature extraction module with a more powerful network for encoding visual information, the pretrained Vision Transformer [Dosovitskiy et al., 2021] from the CLIP framework [Radford et al., 2021], and then explore different approaches to inducing the network to produce more understandable drawings. We compare against the pretrained VGG16 feature extractor used in the original model and develop an approach that enables us to ask what the main semantic content of the drawings is using "prompt engineering" [Radford et al., 2021] with the CLIP model.

5.1 Extended Model with CLIP

The experiments in this chapter follow the game setup presented in Section 4.3 which was inspired by Havrylov and Titov [2017]'s image guessing game. As illustrated in Figure

5.1, the game requires the sender to communicate the target image to the receiver, by sketching 20 black straight lines. The receiver has to guess the correct image from a pool of photographs consisting of K distractors plus the target. The model is trained end-to-end with a multi-class hinge loss, we refer to this game objective as $\mathrm{l}_{\mathrm{game}}$. However, the addition of a perceptual loss, $\mathrm{l}_{\mathrm{perceptual}}$, has been shown to improve humans' ability to recognise the object depicted in the sketch (see Section 4.5.4). An updated schema of the agents' architecture and game setup is shown in Figure 5.1.

In the "original" game setup, the photograph that the sender communicates about matches the target from the receiver's pool of images. In Section 4.3.1, however, we proposed two other game setups in which this requirement does not hold. For the purpose of this study, we only explore the original game variant for which we set the number of distractors, K, to 99. It is worth noting that this sort of game would be very difficult for humans to play. Guessing the target from a set of 100 images, which could contain multiple examples from the same class as the target, based only on a 20-line black and white sketch seems impossible for humans. The trained agents, however, manage to establish a visual communication protocol that can be used to successfully solve the task as shown in Section 5.3.

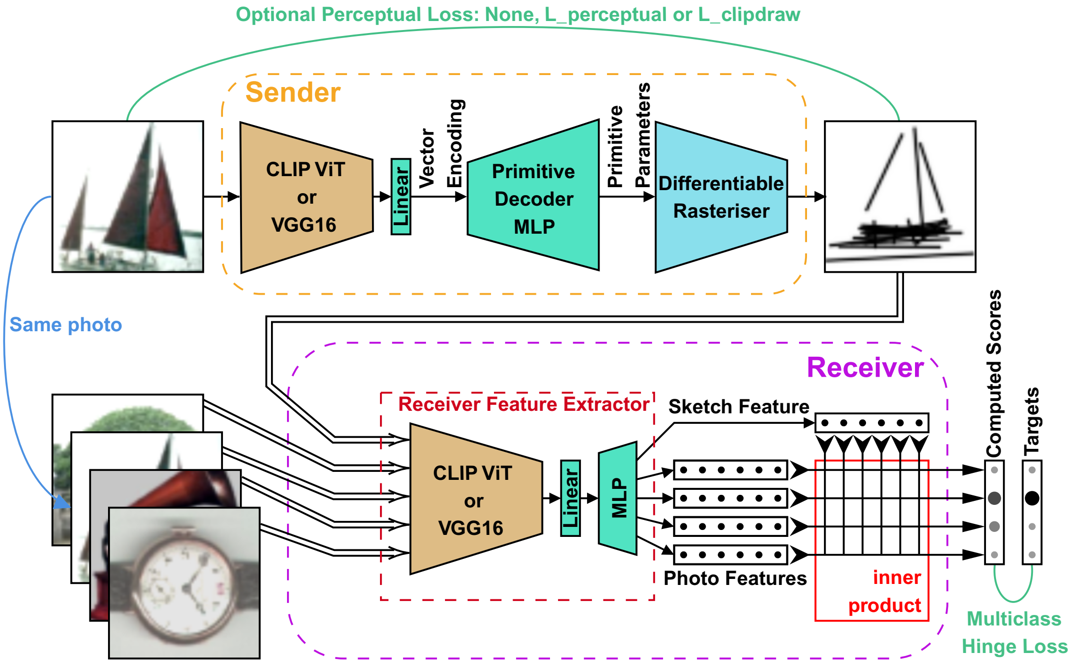


FIGURE 5.1: **Extended model overview.** Two agents are trained to play an image guessing game in which they communicate through a simple line drawing. This is an extension to the model shown in Figure 4.1 in which we experiment with a more powerful pretrained Vision Transformer encoder module (ViT-B/32 from CLIP [Radford et al., 2021]) and different perceptual biases. An additional perceptual loss between the sender's input photo and output sketch induces the sketch to be more understandable.

5.2 Studying Different Perceptual Losses

Zhang et al. [2018] demonstrated that a loss computed using the weighted difference of features extracted across a range of low and intermediate layers in a pretrained VGG16 [Simonyan and Zisserman, 2015] and AlexNet [Krizhevsky et al., 2012] CNN could predict the human perception of the similarity of images. Building upon this idea, we have shown in Section 4.5 that such a loss function could be used to induce a drawing agent to produce sketches that were significantly more interpretable by humans (in the sense of improved agent-human gameplay) than agents trained without such a loss. Without the perceptual loss, the agents could learn to play the game well and generalise to unseen images, but the drawings produced by the sender were essentially visual representations of hash codes with rather random sets of lines.

Moving from the VGG16 feature extraction network to the Vision Transformer (ViT) model of the CLIP framework [Radford et al., 2021], we first explored whether the perceptual loss yields similar results. We extracted the feature after each transformer residual block from both the sketch produced by the sender and the corresponding photo that was presented to the sender and used this to compute the loss. The loss itself involves normalising each layer's features, computing the sum squared difference between sketch ($\boldsymbol{S}$) and image ($\boldsymbol{I}$) features at each layer l and performing a weighted sum over the layers, L,

$$\mathrm{l}_{\text{perceptual}}(\boldsymbol{S}, \boldsymbol{I}, \boldsymbol{w}) = \sum_{l \in L} \frac{\boldsymbol{w}_l}{n_l} \|\hat{\boldsymbol{S}}^{(l)} - \hat{\boldsymbol{I}}^{(l)}\|_2^2 \,, \tag{5.1}$$

where n_l is the dimensionality of the l-th layer feature. This is the same as Equation 4.4, but made more general. For the experiments presented here, we used fixed uniform weights for each layer, $w_l = 1 \, \forall l \in L$. For the effect of different weights refer to Section 4.4.7.

We also investigate another method for generating interpretable drawings, inspired by the approach of Frans et al.. Instead of $\mathrm{l}_{\text{perceptual}}$, we incorporate $\mathrm{l}_{\text{clipdraw}}$ which is computed by the cosine distance (negative cosine similarity) between the encoded representation, $f(\cdot)$ (*e.g.* from the last layer of the ViT encoder, or `relu5_3` of the VGG16) of the generated sketch and input image. However, such a loss alone does not result in sketches that are perceptually similar to the input, so instead perceptual similarity is induced by computing the loss over a set of randomly *transformed* sketches, T:

$$\mathrm{l}_{\text{clipdraw}}(\boldsymbol{S}, \boldsymbol{I}) = -\sum_{t \in T} \frac{f(t(\boldsymbol{S})) \cdot f(\boldsymbol{I})}{\|f(t(\boldsymbol{S}))\| \|f(\boldsymbol{I})\|} \,. \tag{5.2}$$

Following Frans et al. [2021], T consists of four randomly sampled transformations created by applying a random perspective transformation and random resizing and cropping in sequence. This crude modelling of physical spatial constraints is sufficient to induce the sketches to be interpretable.

5.3 Results

This section first presents quantitative and qualitative insights into the drawing protocol evolved by the improved communicating agents. It then outlines our findings from a study in which humans interpret the agents' sketches.

5.3.1 Quantitative and qualitative results

We present results of the visual communication game played with STL-10 images [Coates et al., 2011] in the original game setup described in Section 4.3.1. Figure 5.2 shows test communication success rates and sketches produced by models constructed with either a VGG16 or ViT image encoder, trained with only the game objective $\mathrm{l}_{\mathrm{game}}$, or with the addition of either of the two perceptual losses, $\mathrm{l}_{\mathrm{perceptual}}$ and $\mathrm{l}_{\mathrm{clipdraw}}$, described in Section 5.2. For the experiments run with the ImageNet-pretrained VGG16 image encoder, the parameters specified in Section 4.3.2 are used. When replacing the image encoder with CLIP's pretrained ViT-B/32 model [Radford et al., 2021], we found that increasing the hidden sizes of the Primitive Decoder, shown in Figure 5.1, from 64 and 256 to 1024 each, significantly improves the quality of the sketches. It is also worth noting that a bigger learning rate is needed for this model to converge, more specifically 0.001.

Results show that models trained with only the game objective achieve the highest communication success rate, *i.e.* agents can successfully communicate about the target image, although, they do so by drawing what looks to us like random sets of lines. The addition of perceptual losses to $\mathrm{l}_{\mathrm{game}}$ leads to significantly more interpretable sketches. More sketches generated during testing are shown in Figures 5.3 and 5.4.

5.3.2 Human evaluation

To assess the level of interpretability, we extended the human evaluation presented in Section 4.5 to include the models studied in this chapter. To summarise this pilot study, it consists in pairing a pre-trained sender agent with a human receiver to play the visual communication game through a user interface that presents the human with a sketch and 10 possible photographs to choose from. Each human participant played 30 games (*i.e.* identified 30 sketches) with $K = 9$ distractors for each model configuration. The games are sampled randomly from all those possible within the STL-10 test dataset.

In Figure 5.2, we include human communication success rates averaged over the 6 participants taking part in this pilot study, for the games played with sketches generated by the corresponding models' sender agent. Table 5.1 shows the communication success between the agents playing the games included in this pilot study, the human success rate and an additional measure, the human class communication rate, that looks at

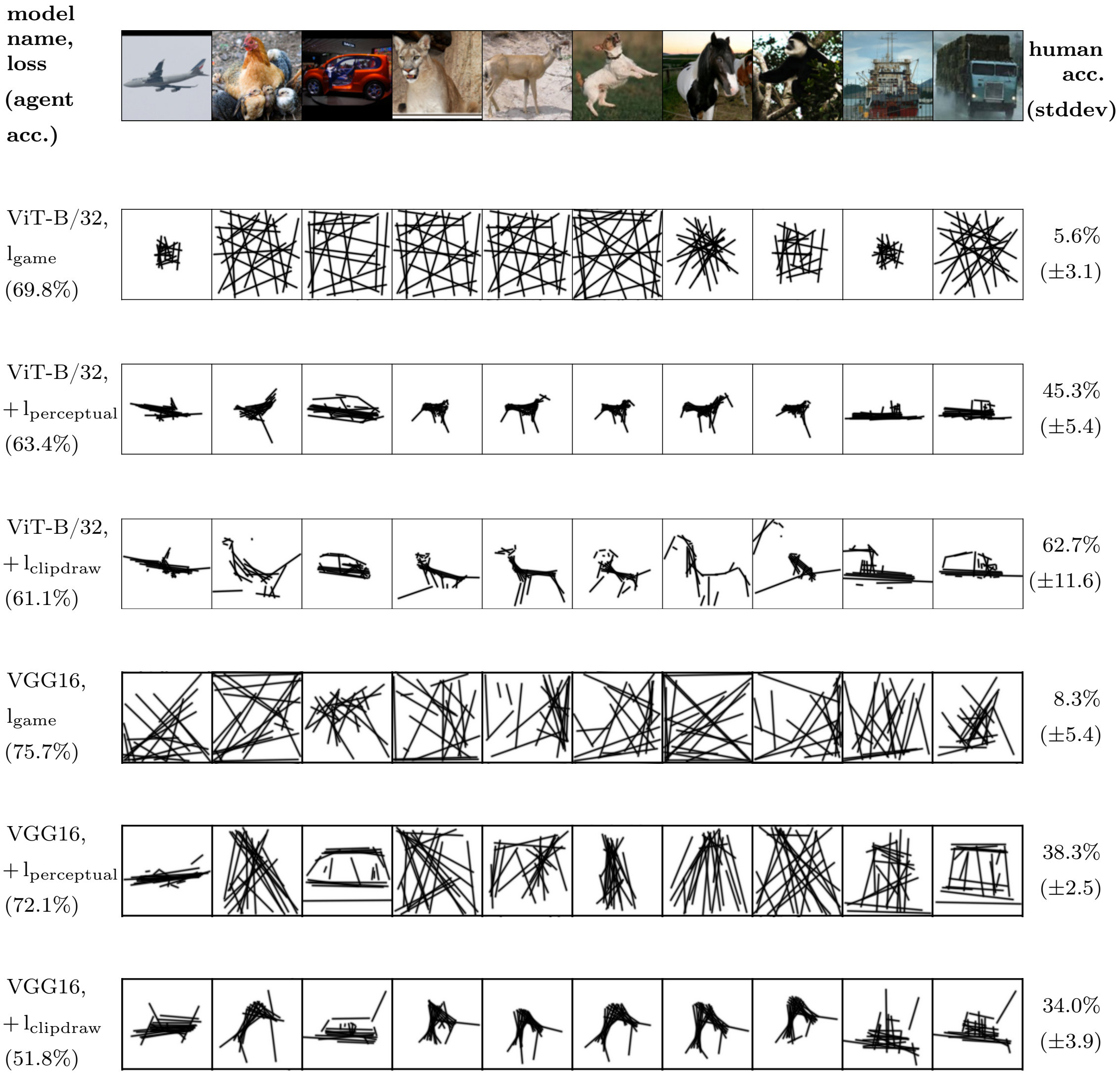


FIGURE 5.2: **Sketches from the visual communication game using STL-10 dataset with different image encoders and "perceptual" losses**. Models trained with the l_{game} only do not learn to draw in an interpretable fashion. For both ViT-B/32 and VGG16 image encoders, the addition of either perceptual loss induces more structure into the resulting drawings, making them more similar to the subject of the image, although it decreases that agents' communication success (shown in brackets on the left side). Perceptual losses also quantitatively improve human performance when pitched against agents (note that reported human accuracies on the right-hand side are for games with 9 distractors as opposed to the agent accuracies on the left with 99 distractors). CLIP-pretrained ViT-B/32 models have higher human performance than the VGG models.

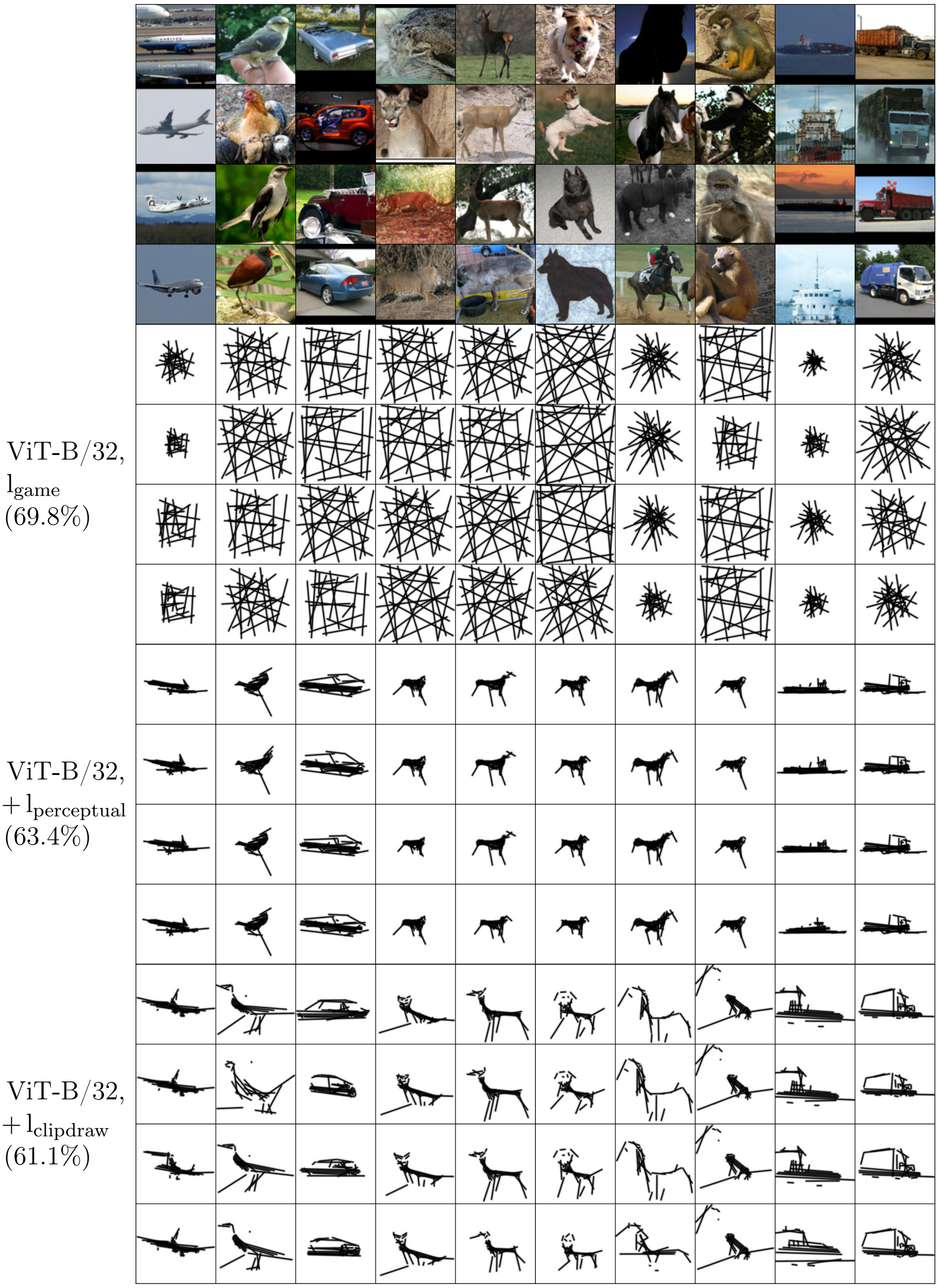


FIGURE 5.3: **More sketches of STL-10 test images produced by the model with ViT-B/32 encoder.**

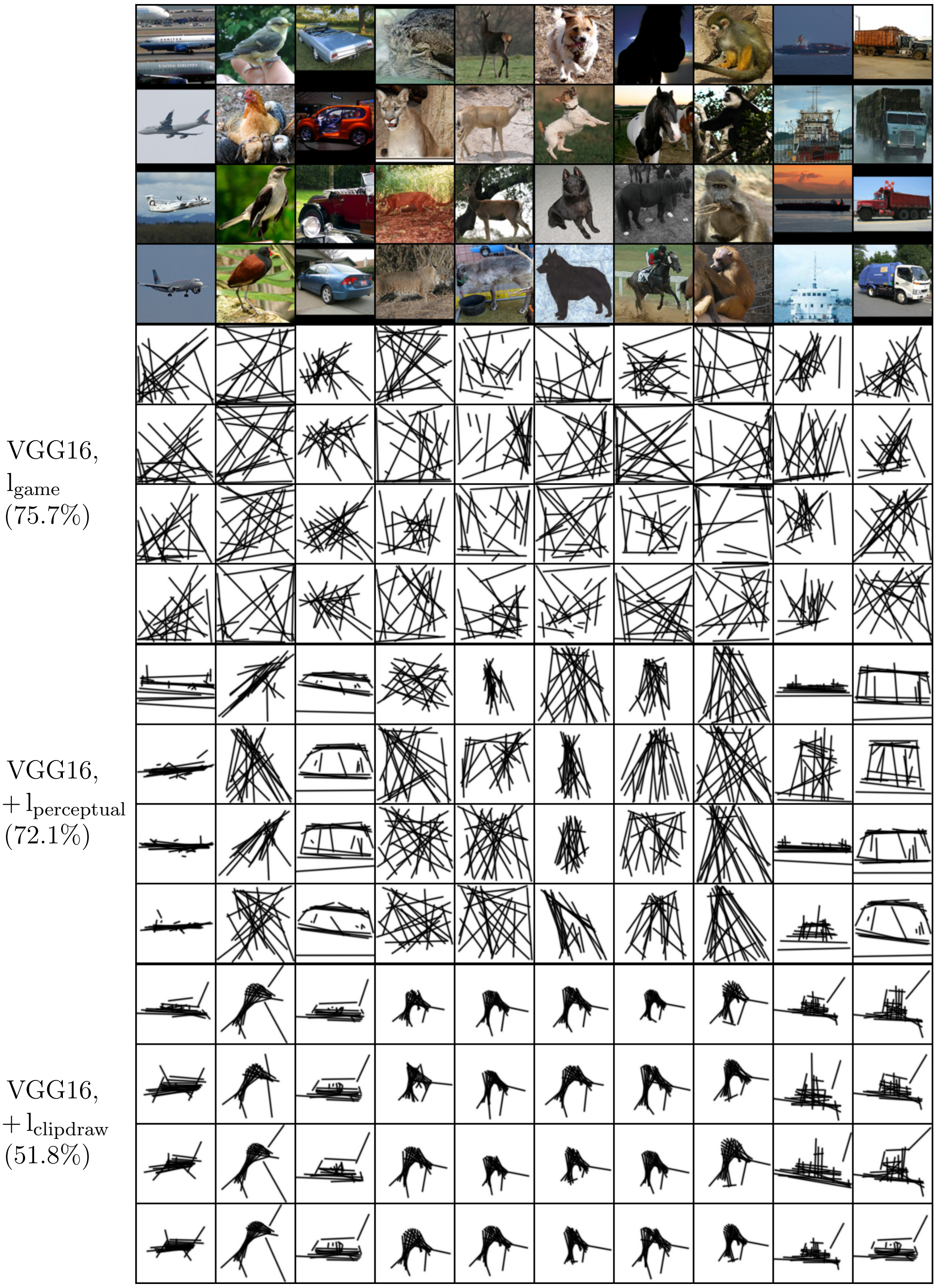

VGG16, $\mathrm{l}_{\mathrm{game}}$ (75.7%)
VGG16, $+\,\mathrm{l}_{\mathrm{perceptual}}$ (72.1%)
VGG16, $+\,\mathrm{l}_{\mathrm{clipdraw}}$ (51.8%)

FIGURE 5.4: **More sketches of STL-10 test images produced by the model with VGG16 encoder.**

Table 5.1: **Human Evaluation results - extended.** Trained agents communicate successfully between themselves in all settings. The addition of either perceptual loss allows humans to achieve significantly better than random performance (images from STL-10, original games have 9 distractors/game for these experiments & random chance is 10%). In addition, humans are better at guessing the correct image class when the models are trained with either of the perceptual losses.

Model	Loss	Agent comm. rate	Human comm. rate	Human class comm. rate
VGG16	l_{game}	100%	8.3%(±5.4)	15.0%(±2.5)
VGG16	$l_{game} + l_{perceptual}$	93.3%	38.3%(±2.5)	55.6%(±7.1)
VGG16	$l_{game} + l_{clipdraw}$	86.7%	34%(±3.9)	49.3%(±7.7)
ViT-B/32	l_{game}	93.3%	5.6%(±3.1)	15.6%(±3.1)
ViT-B/32	$l_{game} + l_{perceptual}$	96.7%	45.3%(±5.4)	63.3%(±7.0)
ViT-B/32	$l_{game} + l_{clipdraw}$	96.7%	62.7%(±11.6)	83.3%(±9.2)

the accuracy of humans at determining the class of the sketch rather than the specific instance.

The model using CLIP's image encoder, pretrained on the task of matching (image, text) pairs, leads to better image representations, and eventually, sketches that can be more easily interpreted by humans than those produced by a model with a VGG16 encoder pretrained for the supervised image classification task on ImageNet. Similarly, we observed that the addition of either of the perceptual losses significantly improves humans' ability to recognise the main category depicted in the sketch.

5.4 Investigating the Meaning of Drawings with Prompt Engineering

It would be beneficial to be able to understand what information the sender agent is trying to convey through its sketch and compare that to what a human playing the game might try to impart. In the particular game setting we are using, if one communicates only the object in the scene then the expected communication rate would be only 10%. To achieve higher rates much more nuanced information about the image contents needs to be conveyed. Ultimately understanding what is being communicated, and how it differs from humans would allow us to design better approaches to inducing more human-like behaviour in the model and the agent's internal representations.

To start to explore this in more detail, we demonstrate that we can begin to answer the question of what is being communicated by using the CLIP model as a probe. With the technique of prompt engineering, where a set of textual prompts are encoded with CLIP's language model, it becomes possible to ask basic questions about how CLIP perceives a sketch in terms of the semantic content. For the initial experiments presented here we use two prompt templates: “`a drawing of a XXX.`” and “`a photo of a XXX.`”. The placeholder (`XXX`) is replaced by the 10 different classes in the STL-10 dataset to create a complete set of 20 prompts. For each of the models, we then compute several statistics regarding CLIPs perception of the sketch, the target image (*i.e.* the receiver image that is the true answer) and the guessed image (the image that the receiver actually picked), averaged over all 8000 possible games in the STL-10 test set. More specifically we ask which class CLIP perceives an image $\boldsymbol{I}$ to be, using a function $\mathrm{c}(\boldsymbol{I})$ which returns the placeholder of the closest prompt (using cosine similarity in the embedding space), and compare CLIP‘s predicted class for the sketch, guess and target. In a similar way, we also compute which of the two templates `{photo, drawing}` CLIP predicts the sketch, target and guess to belong to. Finally, we also utilise a function $\mathrm{gt}(\mathit{input})$ which returns the STL-10 ground-truth class label of the sender agent’s input, to allow us to analyse to what extent CLIP’s perception of the images matches the true label. The results of this analysis are shown in Table 5.2.

Table 5.2: **Comparing models with CLIP using prompt engineering:** $\mathrm{c}(\boldsymbol{I})$ **returns which class CLIP perceives image** $\boldsymbol{I}$ **to be;** $\mathrm{gt}(\mathit{input})$ **returns the true class of the sender agent’s input photo;** $\mathrm{tp}(\boldsymbol{I})$ **returns the type (photo or drawing) CLIP predicts** $\boldsymbol{I}$ **to be.** There are significant differences between models, however, it is clear that the perceptual losses strongly encourage a more object-centric representation. CLIP is very good at telling the difference between sketches and photos in all cases, despite the perceptual losses pulling together the representations.

	VGG16 encoder			ViT-B/32 encoder		
	l_{game}	$+l_{\text{perceptual}}$	$+l_{\text{clipdraw}}$	l_{game}	$+l_{\text{perceptual}}$	$+l_{\text{clipdraw}}$
c(sketch)==gt(input)	7.3%	24.0%	41.2%	9.4%	96.4%	96.6%
c(sketch)==c(target)	7.3%	24.2%	41.5%	9.7%	96.4%	96.5%
c(sketch)==c(guess)	7.6%	24.6%	38.8%	10.4%	94.9%	96.1%
c(target)==gt(input)	97.3%	97.3%	97.3%	97.3%	97.3%	97.3%
c(guess)==gt(input)	85.6%	82.1%	76.5%	77.6%	94.8%	95.8%
tp(sketch)==’drawing’	100%	99.9%	99.9%	99.9%	96.2%	99.3%
tp(target)==’photo’	99.4%	99.4%	99.4%	99.4%	99.4%	99.4%
tp(guess)==’photo’	99.4%	98.4%	98.4%	99.0%	99.4%	99.4%

The results in Table 5.2 indicate that both forms of perceptual loss do a good job of making the sender agent produce sketches that capture the main class of object in the input image. There is an inherent bias towards CLIP generated sketches because the same model is being used to perform the generation and the probing. The fact that for all models `c(guess)==gt(input)` rates are so high suggests CLIP identifies the class of the guessed image to be correct, *i.e.* be the same as the ground truth label. On the other hand, `c(sketch)==c(target)` measures if the class CLIP thinks the sketch to be is the same as the class that CLIP thinks the target image to be. The fact that this measure is much lower across all VGG16 models suggests that sketches produced with this feature encoder are not as interpretable to the CLIP model as the sketches produced with the ViT-B/32 encoder. As can be seen in Figure 5.2, the CLIP generated sketches are qualitatively more interpretable than the VGG ones. When looking at these results bear in mind that the communication game itself is entirely self-supervised; the notion of object class is clearly not required for successful communication and is instead a side effect of inducing a perceptual loss between internal representations. The results also show that despite the perceptual losses forcing the representations of the sketch and image together, the CLIP-based probe is able to recognise the sketch as being a drawing and the receiver images from the dataset as being photos almost all of the time.

5.5 Summary

This chapter looked at how representational losses influence the sketches produced by artificial agents playing a visual communication game. We showed that the addition of either perceptual loss to the communication game leads to qualitatively more recognisable sketches than those produced by agents trained with the game objective only. Although the additional representational losses slightly decrease the agents' ability to communicate, they significantly increase the possibility to recognise the sketches as the semantic category of the photographs they represent (as shown in Table 5.2 and with the human evaluation presented in Figure 5.2). The striking differences between images from the two loss formulations raise lots of questions and this is definitely an area we would like to explore in future work. Undoubtedly, there are many other formulations that would be exciting to experiment with too. Going forwards it would also be interesting to explore if it is possible to minimise the drop in communication rates that arise from introducing this perceptual bias.

Our brief experiments with prompt engineering in Section 5.4 open up a number of doors for future analysis. The game being played by the agents is complex, and the communication success rates are far in excess of the 10% that would naïvely result from the models only communicating information about the class. The obvious next step is to question what additional information is being conveyed in the sketches; is it interpretable semantic information about the input image, or is it some kind of neural hash code that just happens to allow communication to succeed, or is it a mixture of both aspects? Further, similar to the causal interventions that are applied in emergent communication scenarios with non-visual channels, we would also wish to explore which parts of a sketch (perhaps bundles of strokes) contribute to particular aspects of semantic meaning. We hope that with richer datasets and considerably more engineering of prompts, the setup outlined in this chapter would allow these goals to be achieved.

Chapter 6

Conclusions

"Art is the queen of all sciences communicating knowledge to all the generations of the world."

— Leonardo da Vinci

"I do not want art for a few any more than education for a few, or freedom for a few."

— William Morris

The goal of this thesis has been to elucidate how various representations of the visual world influence the emergence and interpretability of communication protocols. In particular, this programme of research has explored a number of communication bottlenecks between artificial agents tasked to communicate and collaborate to achieve a shared goal. Communication among artificial agents and, eventually, with humans has become a subject increasingly important with the progressively automatised world that we live in. This final chapter attempts to summarise and highlight the main contributions of previous chapters and discuss a series of open questions. It ends with suggestions for future research following on from the findings presented in this thesis and with a look at where the field of self-supervised emergent communication is heading in the future.

6.1 Summary and Conclusions

Perceiving the visual world, learning to represent it internally and attribute meaning to it, and then using these representations in communication with other participants, are all very complex tasks that span across a number of research fields. The complexity of these tasks is reflected in Chapter 1 which gives an overall picture of what these processes entail, how they are achieved by humans, and how our understanding of them has changed over the years. This investigation then transitioned into how these processes have been modelled and become part of artificial intelligence. The field of emergent communication has seen a surge of interest in the last 5 years from a number of studies exploring the ability of artificial agents to cooperate on a shared goal and to develop communication protocols as part of the process. The last section of this chapter looked at gameplay as a framework for learning and developing skills such as language.

Starting from prior work in the field of emergent communication, Chapter 2 explored discrete token-based communication between artificial agents which learn to cooperatively solve an image-reference game. To reiterate, the task involves a *sender* agent communicating information about a visual scene through, in this case, a sequence of tokens chosen from a predefined vocabulary. The *receiver* is asked, based on the message from the sender, to pick the image it thinks the former has seen from a series of images which includes the target and a number of distracting photographs. Concretely, the research in this chapter focused on the factors which influence the human interpretability of emergent communication. We first investigated the biases introduced in the communication protocol by pretrained feature extraction networks and confirmed that networks pretrained on a supervised task lead to improved semantics, while fully learned or random-and-fixed networks learn to extract more low-level information which is not so decipherable for a human interpreter. Then, an analysis of the effect of different data augmentations decided for both the sender and the receiver together or independently, showed that choosing the right type and complexity for these can improve semantic alignment and reduce a hashing-like solution in end-to-end trained models.

The second part of Chapter 2 looked at multi-task learning as a form of grounding the agents' visual systems with semantic notions without the need for external human supervision. Likewise, we explored the effect of pre-training the visual feature extractor on a self-supervised task and demonstrated that such a task, which can be seen as nothing other than self-play of a different game before engaging in the main communication task, can further improve the quality of semantics captured by a fully learned model. The findings in this chapter suggested that creating the right environments for agents to learn by solving multiple tasks, either sequentially or concurrently, can be beneficial for developing a meaningful and interpretable communication protocol. If this is actually desirable, and the challenges of such an approach, are discussed in Section 6.3.2.

The motivation for developing intelligent agents that can communicate with each other and with us is founded on the idea that they will become useful in assisting humans in various tasks in the digital era. One could envisage coordination between self-driving cars, or smart appliances in one's home. Therefore, ensuring an interpretable communication protocol or one which can easily be processed into something that humans would understand, be it natural language, sound, or visual representations, is of increasing importance. The thesis direction shifts in Chapter 3 which lays the groundwork for visual communication between artificial agents. When humans look at any of the signs shown in Figure 6.1, no matter the language they speak, it is most likely that they will understand their meaning.

Figure 6.1: **Illustration of interpretable icons which could be understood no matter the country they appeared in.** Image sourced from Robinson [2002].

Training agents to communicate through drawing in an end-to-end fashion necessitates a differentiable rasteriser module that allows agents to draw using parameterised strokes.

In Chapter 3, a bottom-up differentiable rasterisation method was proposed to model the drawing act by generating pixel rasters from vector primitives. We showed that with this approach we can turn raster images into vector parameterised strokes, allowing for a variety of primitive parametrisations and composition operators, and then back into raster images. By directly optimising the chosen primitive's parameters against the original image, we can create sketches of any photograph using points, straight line segments, Bézier curves or Catmull-Rom splines, with or without learned connections. We illustrated the power of our differentiable drawing framework, which can be incorporated as part of any model, through a series of experiments. First, we looked at direct optimisation using different loss functions against raster images including cartoons, photographs and paintings. Most importantly, we demonstrated that our technique not only lets you uncover compact approximations of stroke definitions required to produce an image as a digital raster but also as *physical* continuous strokes with a drawing instrument manipulated by a robot. Lastly, we showed it can be used as part of larger frameworks such as autoencoder models performing autotracing, *i.e.* producing vector images from bit-mapped ones. We tested the efficiency of this application on a variety of raster image datasets (MNIST, KMNIST, QuickDraw and Omniglot).

Using the machinery described in Chapter 3, in Chapter 4 we explored a framework of cooperating agents that learn to communicate by drawing in order to solve an image-referential game. Limiting the agents to drawing only 20 line sketches in black and white, we first examined whether the task can be solved at all through such a bottleneck. Our experiments showed that agents can successfully communicate, although the resulting sketches in the baseline setup do not capture any semantics as understood by humans. We then investigated the communication protocol emergent in various configurations by modifying the game setup and objective (*e.g.* the object-oriented game variants, or training against an arbitrary objective), introducing additional loss terms (*e.g.* $l_{\text{perceptual}}$) or biases such as texture or shape (*e.g.* by training on different raster image datasets).

Most importantly, Chapter 4 presented a means of communication that can be directly interpreted by human observers. Through our human evaluation experiment, in which human participants were asked to play the same image referential game with sketches drawn by a pretrained *sender* agent, we analysed which are the factors that help increase interpretability and humans' success in the game. One of our main findings is the importance of the additional perceptual loss which encourages sketches produced by artificial agents to look more like the images they are meant to represent. Although agents could communicate successfully without such a loss, the resultant sketches would be difficult for a human to interpret, much like in the hashing-like scheme mentioned in Chapter 2. Likewise, we observed the importance of defining a game setup and objective which encourages representations symbolic for the object class rather than a specific instance; this is something which we observed humans are also better at, distinguishing between distinct classes rather than between instances of the same class.

Finally, Chapter 5 extended the framework for communication through sketching proposed in Chapter 4. In this chapter, we explored a more powerful visual encoder which has been shown to capture better semantics by having been pretrained on the task of matching pairs of image and text [Radford et al., 2021]. We compared against the models with a VGG16 pretrained encoder and show that models with a CLIP-pretrained visual transformer network improve the sketching quality which leads to an increased human gameplay success rate. Moreover, we extended the exploration of perceptual losses and introduce $\mathrm{l}_{\mathrm{clipdraw}}$ which measures similarity between encoded representations of randomly transformed sketches and the target image. Lastly, we proposed probing the semantic content of the drawings using the technique of prompt engineering. This was achieved with CLIP's language and vision encoders which can assess the similarity in terms of distance in the latent space between embeddings of sketches and embeddings of textual prompts which describe the semantic content of the possible target classes. Our results indicated that both perceptual losses strongly encourage a more object-centred, hence semantic, representation.

6.1.1 Revisiting contributions in this thesis

To conclude the summary of the research programme, this section reaffirms the contributions made by this thesis:

- This thesis has shown that, with the appropriate inductive biases and task objectives, meaningful symbolic representations of visual input can emerge in a fully self-supervised framework of communicating agents.
- This research bridged a gap between self-supervised learning and fully learned communication emerging between cooperative agents sharing a goal (*e.g.* an image signalling or a rotation prediction task). Our results highlighted the importance of combining or interleaving multiple self-supervised tasks to balance the trade-off between task success and the interpretability of the emergent protocol.
- This thesis proposed a technique for differentiable rasterisation of vector primitives which enables the physical act of drawing in a flexible way and which can be used as part of any learned model.

- Based on the approach of differentiable drawing, this thesis presented the first framework for fully self-supervised agents that learn to communicate through sketches. This research pioneers a new direction to be explored in future emergent communication studies.
- Through a study with human participants, it was demonstrated that with the appropriate perceptual biases and game objective it is possible for a trained sketching agent to successfully communicate with humans.

6.2 Open Questions and Future Work

While this thesis has covered much ground in the field of inter-agent communication, it also opened up a number of questions to be answered in future research. In this section, we discuss the potential avenues for further investigation following on from the research presented here.

Chapter 4 considered drawing as a much simpler and more interpretable form of communication than language. In the future, a more flexible definition of a sketch could be explored. In most of our experiments, we restricted the agents to communicate only through black and white sketches composed of a fixed number of line segments (20 is the baseline, although we looked at the effect of increasing/decreasing the drawing complexity in Section 4.4.4). However, one could envisage a model that could choose the primitive type (*e.g.* instead of straight lines, use points or curved segments) or that could learn to use a variable number of such primitives to communicate effectively.

Likewise, what if colour were to be included in the sketches? It is well known that colour, besides shape, plays an important role in humans' ability to recognise objects [Harris et al., 2021; Oliva and Schyns, 2000; Rousselet et al., 2005]. As discussed in Section 4.5.5, the sketches representing objects of the same class could be better differentiated if the colour were to be included for example. However, when humans play Pictionary, colour is not allowed as part of the game. So a natural question that arises is *how can one represent colour when drawing only in black and white*? Consideration of more abstract visual features, such as colour or quantity, *if* and *how* these can be learned and represented by sketching agents would be an interesting direction for future research.

Along the same lines, a direction for further investigation could follow from the prompt engineering experiment described in Section 5.4. Our results showed that drawings produced by self-supervised agents communicate more than just the object class. Therefore, in the future, it would be interesting to uncover which parts of the sketch correspond to the different aspects, of semantics or not, in the visual scene. Running causal interventions on the strokes, perhaps with a language model as done on non-visual communication channels, could be a possible approach.

Based on our study on inducing perceptual similarity between target photographs and sketches produced by artificial agents in Chapter 4, one question to be answered is whether it would be possible to learn individual weightings for the feature maps from each layer of the feature extraction network. In our perceptual loss formulation, we decided to weigh the feature maps from each layer equally. However, preventing a neural network from learning zero weights and thus ignoring the perceptual loss altogether is challenging.

As previously mentioned in Section 4.6, another open question related to interpretability based on our study of sketching is how much are the drawings produced by agents influenced by the training biases of their visual feature extraction network (*e.g.* the labels of classes present in ImageNet). What would the drawings look like if the visual backbone were to be pretrained with a self-supervised objective function [Zbontar et al., 2021] on ImageNet instead?

Lastly, we showed that it is possible to teach artificial agents to draw in order to communicate about visual stimuli. An interesting question building upon the sketching framework presented here is whether it would be possible to train agents to visually represent, *i.e.* draw to communicate the meaning of, other data modalities such as sound [Löbbers and Fazekas, 2022]. Could an artificial agent learn to represent the song of a bird as a symbolic sketch of one, or would it learn to associate the sound of a honk with a drawing of a car?

The ideas discussed in this section constitute just a selection. Overall, the research programme in this thesis encourages flexibility in modelling interaction and communication between artificial agents. Exploration of tasks that involve diverse data modalities and goals might help promote the intuitive knowledge humans develop about the world. Equally important is considering flexibility in representations and thorough analysis of biases "built in" the environments defined for studying communication.

6.3 The Future of Emergent Communication

The field of emergent communication is extremely interesting at the current time. The trends in the field can be separated based on the mode of communication, linguistic or visual. This section covers prospects for each modality tackled individually, but also combined.

6.3.1 Visual

Given the simplicity and endurance of drawing as a form of recording human thoughts and transmitting information since prehistoric times, future research could explore the transition from pictorial to linguistic communication among artificial agents that can draw. The historical graphical artefacts and knowledge we now have about the incentives behind them, as well as their role in the evolution of language over time should serve as conceptual touchstones. Future research on emergent visual communication between artificial agents should address the following questions: When and how do visual representations (*i.e.* sketches) emerging between pairs or among populations of artificial agents turn into symbolic conventions? By convention, a globally coordinated and agreed-upon representation is meant. The evolution of Chinese characters shown in Figure 1.7 illustrates such a scenario which would be interesting to explore in the future. Chinese, Japanese or Egyptian hieroglyphs are examples of writing systems based on logograms (*i.e.* written characters that represent whole words). Although initial attempts on emergent graphical conventions have been made by Qiu et al. [2021] whose work is concurrent to the research presented in Chapter 4, we believe the topic could be taken forward with the framework presented in this thesis as it enables flexibility in the drawing procedure. Hence, it would be intriguing to see what type of "written" symbolic system (be it alphabets, logograms or logo-syllabic) agents invent and what properties this system has.

Also from the point of view of language evolution, there exist various cultural accounts of the transmission of language as discussed in Section 1.3.1. Fay et al. [2010] contrasts two views on the evolution of sign systems: iterated individualistic [Kirby, 2002; Kirby and Hurford, 1997] and socially collaborative [Steels, 1997]. The framework proposed in Chapter 4 takes on elements from both views. As Fay et al. describes it, in the iterated learning account, communication emerges vertically; it is transmitted between generations, and hence is biased by individuals' prior representations. Same as in our approach, we employ a form of visual system pretrained on a specific dataset (ImageNet in most cases), hence, the communication protocol is shaped also by the biases confined within it. As this account predicts, we have observed that individual pairs trained on distinct datasets converge to different equilibrium points which make communication *across* dyads (individuals of a pair) not as effective and efficient as *between* dyads. This is due to the fact that the systems of pictorial representations evolved within distinct pairs do not correspond. By contrast, the collaborative learning account assumes a horizontal bidirectional transmission between agents such that communication emerges as the whole community agrees on the meaning. A straightforward extension to the research on graphical communication presented in this thesis would be to explore the development of a *global*, instead of a local, system shaped by populations of sketching agents and the role of "social" feedback at a larger scale.

Then, there exists potential in taking forward drawing as a means of communication to help us understand better the human ability to combine a wide variety of graphical and symbolical representations. It is well known that human intelligence is not something one is born with. Instead, intelligence becomes embodied and develops through exploration of the environment, interaction with its contents and with other social partners and by receiving feedback from these experiences [Smith and Gasser, 2005; Sachs et al., 1981]. Bisk et al. [2020] also discuss the importance of embodied experiences and social interactions for improving linguistic communication. In the future, it would be interesting to explore how visual communication, as a means of directly transmitting intent, would be shaped through the interaction between humans and artificial intelligent agents. We have shown in this thesis that it is possible for humans to understand the semantics of drawings produced by agents that can sketch. But could a trained receiver agent interpret sketches created by humans and how would this exchange of information influence the emergent graphical conventions?

Lastly, sketching can easily confer meaning by selecting and emphasising what is most relevant to the task and omitting irrelevant details. It is an iterative process and supports flexible, progressive and constructive thinking [Tversky et al., 2003; Goldschmidt, 1992; Fish and Scrivener, 1990]. There exist a vast number of applications for intelligent machines capable of understanding or being able to produce diverse sketched concepts like sheets, histograms, maps, engineering sketches or prototype designs [Tversky et al., 2003; Willis et al., 2021]. In the long run, developing sketching may enhance the quality of human-to-machine communication and hence assist humans in achieving their goals or even further expand their intellectual reach.

6.3.2 Linguistic

At the time of writing this thesis, a good review of outstanding trends in emergent linguistic communication in multi-agent environments was given by Baroni [2022]. The following section is structured based on this review. First, there exists the motivation for emergent communication as a method of achieving *interactive* agents, *i.e.* machines that can communicate efficiently with humans through language. A use case for such intelligent machines that can help humans through conversation can already be seen in smart virtual assistants (*e.g.* Amazon Alexa) or chatbots. Recent works in this line of research advocate that scaling up datasets, population size, task and environment complexity are all essential for building interactive AI and modelling human communication [Kalinowska et al., 2022; Chaabouni et al., 2021; Dessì et al., 2021; Lin et al., 2021; Lazaridou and Baroni, 2020]. In the future, we need to consider how we can leverage these aspects to help move the field closer to its goals. Moreover, based on the results regarding interaction and interpretability presented in this thesis, questions such as "what human priors are desirable in an emergent protocol?" should be answered in future research. Lastly, we recognise that the single-step referential game setup used in this thesis and in many current studies of emergent communication is not representative of real-world scenarios in which interaction through conversation occurs. In most real-life setups, the exchange of information follows in turns and is not limited to a certain number of referents. Hence future research motivated by the goal of interaction should consider more complex and dynamic setups [Kalinowska et al., 2022].

The second goal of researching emergent linguistic communication, although it can be considered for modalities other than language too, is that of machine-to-machine communication (M2M) [Weyrich et al., 2014; Amodu and Othman, 2018]. Today, we live surrounded by smartphones, smart devices in our homes and workplaces, as well as self-driving cars. One could envisage scenarios in which such smart machines could communicate to one another, with no human supervision, and thus make our lives less complicated and more enjoyable. Although ethics considerations will most probably be an issue, an example could be that of self-driving cars being able to communicate about road closures or accidents. Or autonomous delivery robots coordinating with one's grocery list or calendar to schedule the most convenient delivery time slot. The applications for autonomous interaction of smart devices without human intervention are countless. The methods and questions posed by emergent communication can serve M2M whose aim is to ultimately provide a shared protocol that can scale to any number or type of devices. Priorities in this line of research will then be centred on answering questions such as: What sort of communication system emerges from M2M? What properties does it have? Should it have, if any at all, human language principles such as discreteness or compositionality?

It is fair to say that these questions, as the whole field of emergent communication, are all very challenging, but equally exciting. With today's deep learning techniques, computing power and knowledge about human cognition we are getting closer than ever to reaching interactive intelligent agents and autonomous communication. By investigating different learning biases, objectives and communication modalities, we advance our understanding of how intelligent agents perceive, represent and decide to communicate about the visual world. In the long run, the fruit of these investigations may help inspire different approaches to modelling communication, linguistic or visual, and draw on lessons from human perception and cognition in doing so.

Bibliography

Diane Ackerman. *Deep play.* Vintage, 2011.

Christoph Adami. The use of information theory in evolutionary biology. *Annals of the New York Academy of Sciences*, 1256(1):49–65, 2012.

Adobe. Illustrator. `https://www.adobe.com/products/illustrator.html`, 2022. Accessed May 2022.

Shaaron Ainsworth, Vaughan Prain, and Russell Tytler. Drawing to learn in science. *Science*, 333(6046):1096–1097, 2011.

Oluwatosin Ahmed Amodu and Mohamed Othman. Machine-to-machine communication: An overview of opportunities. *Computer Networks*, 145:255–276, 2018.

Jaume Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial intelligence*, 201:81–105, 2013.

Natalie Angier. The purpose of playful frolics: Training for adulthood. *New York Times*, 20:B5, 1992.

Autodesk. Autocad software. `https://www.autodesk.co.uk/products/autocad/overview`, 2022. Accessed May 2022.

Samaneh Azadi, Matthew Fisher, Vladimir G Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. Multi-content GAN for few-shot font style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7564–7573, 2018.

S Balasubramanian, Vineeth N Balasubramanian, et al. Teaching GANs to sketch in vector format. *arXiv preprint arXiv:1904.03620*, 2019.

Adrien Bardes, Jean Ponce, and Yann Lecun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

Marco Baroni. Rat big, cat eaten! Ideas for a useful deep-agent protolanguage. *arXiv preprint arXiv:2003.11922*, 2020.

Marco Baroni. Deep net emergent communication: Why bother? ICLR 2022 Emecom Workshop (5th Workshop on Emergent Communication), 2022. URL `https://marcobaroni.org/publications/lectures/marco-emecomm-iclr-2022.pdf`.

Elizabeth Bates. Language development. *Current opinion in neurobiology*, 2(2):180–185, 1992.

Vineet Batra, Mark J Kilgard, Harish Kumar, and Tristan Lorach. Accelerating vector graphics rendering using the graphics hardware pipeline. *ACM Transactions on Graphics (TOG)*, 34(4):1–15, 2015.

Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR*, abs/1206.5538, 2012.

Mikhail Bessmeltsev and Justin Solomon. Vectorization of line drawings via polyvector fields. *ACM Transactions on Graphics (TOG)*, 38(1):1–12, 2019.

Derek Bickerton. *Language and species.* University of Chicago Press, 1990.

Derek Bickerton. *More than nature needs.* Harvard University Press, 2014.

Irving Biederman. Perceiving real-world scenes. *Science*, 177(4043):77–80, 1972.

Irving Biederman. On the semantics of a glance at a scene. In *Perceptual organization*, pages 213–253. Routledge, 2017.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, et al. Experience grounds language. *arXiv preprint arXiv:2004.10151*, 2020.

Ioana Boghian, Venera-Mihaela Cojocariu, Carmen V. Popescu, and Liliana Mata. Game-based learning. using board games in adult education. *Journal of Educational Sciences and Psychology*, IX (LXXI)(1), 2019.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

Diane Bouchacourt and Marco Baroni. How agents see things: On visual representations in an emergent language game. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 981–985, 2018.

Susan E Brennan and Herbert H Clark. Conceptual pacts and lexical choice in conversation. *Journal of experimental psychology: Learning, memory, and cognition*, 22(6):1482, 1996.

J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965. .

Karl Buhler. *The mental development of the child: A summary of modern psychological theory*. Routledge, 2013.

Roger Caillois. *Man, play, and games*. University of Illinois press, 2001.

Angelo Cangelosi and Domenico Parisi. *Simulating the evolution of language*. Springer-Verlag New York, Inc., 2002.

Kris Cao, Angeliki Lazaridou, Marc Lanctot, Joel Z Leibo, Karl Tuyls, and Stephen Clark. Emergent communication through negotiation. In *International Conference on Learning Representations*, 2018.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. Anti-efficient encoding in emergent communication. *CoRR*, abs/1905.12561, 2019.

Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. Compositionality and generalization in emergent languages. *arXiv preprint arXiv:2004.09124*, 2020.

Rahma Chaabouni, Florian Strub, Florent Altché, Eugene Tarassov, Corentin Tallec, Elnaz Davoodi, Kory Wallace Mathewson, Olivier Tieleman, Angeliki Lazaridou, and Bilal Piot. Emergent communication at scale. In *International Conference on Learning Representations*, 2021.

Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised GANs via auxiliary rotation loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12154–12163, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.

Dorothy L Cheney and Robert M Seyfarth. *How monkeys see the world: Inside the mind of another species*. University of Chicago Press, 1990.

E Cherry. A history of the theory of information. *Transactions of the IRE Professional Group on Information Theory*, 1(1):22–43, 1953.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. .

Jungwoo Choi, Heeryon Cho, Jinjoo Song, and Sang Min Yoon. Sketchhelper: Real-time stroke guidance for freehand sketch retrieval. *IEEE Transactions on Multimedia*, 21(8): 2083–2092, 2019.

Morten H Christiansen and Simon Kirby. Language evolution: Consensus and controversies. *Trends in cognitive sciences*, 7(7):300–307, 2003.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

Christopher Clark, Jordi Salvador, Dustin Schwenk, Derrick Bonafilia, Mark Yatskar, Eric Kolve, Alvaro Herrasti, Jonghyun Choi, Sachin Mehta, Sam Skjonsberg, et al. Iconary: A pictionary-based game for testing multimodal communication with drawings and text. *arXiv preprint arXiv:2112.00800*, 2021.

Jean Clottes. *Cave art.* Phaidon London, 2008.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

Thomas M Connolly, Elizabeth A Boyle, Ewan MacArthur, Thomas Hainey, and James M Boyle. A systematic literature review of empirical evidence on computer games and serious games. *Computers & education*, 59(2):661–686, 2012.

Antonia Creswell and Anil Anthony Bharath. Adversarial training for sketch retrieval. In *European Conference on Computer Vision*, pages 798–809. Springer, 2016.

Mihaly Csikszentmihalyi. Play and intrinsic rewards. In *Flow and the foundations of positive psychology*, pages 135–153. Springer, 2014.

Mihaly Csikszentmihalyi and Stith Bennett. An exploratory model of play. *American anthropologist*, 73(1):45–58, 1971.

Robert Cummins, Hilary Putnam, and Ned Block. *Representations, targets, and attitudes.* MIT press, 1996.

Dennis M Dake and Brian Roberts. The visual analysis of visual metaphor. 1995.

Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2951–2960, 2017.

Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning*, pages 1538–1546. PMLR, 2019.

Ayan Das, Yongxin Yang, Timothy Hospedales, Tao Xiang, and Yi-Zhe Song. Béziersketch: A generative model for scalable vector sketches. In *European Conference on Computer Vision*, pages 632–647. Springer, 2020.

Ayan Das, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Cloud2curve: Generation and vectorization of parametric sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7088–7097, 2021.

Terrence Deacon. The symbolic species, new york: W, 1997.

Laurent Demaret, Nira Dyn, and Armin Iske. Image compression by linear splines over adaptive triangulations. *Signal Processing*, 86(7):1604–1616, 2006.

Rene Descartes. Meditations on first philosophy. 1641.

Roberto Dessì, Eugene Kharitonov, and Baroni Marco. Interpretable agent communication from scratch (with a generic visual processor emerging on the side). *Advances in Neural Information Processing Systems*, 34, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *The IEEE International Conference on Computer Vision*, pages 1422–1430, 2015.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Tom Duff. Polygon scan conversion by exact convolution. In *International Conference On Raster Imaging and Digital Typography*, pages 154–168, 1989.

C von Ehrenfels. Über gestaltqualitäten. *Vierteljahrsschrift für wissenschaftliche Philosophie*, 14(3):249–292, 1890.

Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. Emergent communication in a multi-modal, multi-step referential game. *arXiv preprint arXiv:1705.10369*, 2017.

Michael W Eysenck. *Principles of cognitive psychology.* Lawrence Erlbaum Associates, Inc, 1993.

MW Eysenck and MT Keane. Cognitive psychology: a student's handbook hillsdale. *NJ: Earlbaum*, 1995.

Antonio Elias Fabris and A Robin Forrest. Antialiasing of curves by discrete pre-filtering. In *The conference on Computer Graphics and Interactive Techniques*, pages 317–326, 1997.

Paullex Fadirsair, Henderika Serpara, and Wilma Akihary. Application of mime and pictionary game methods on students'german vocabulary mastering. *HUELE: Journal of Applied Linguistics, Literature and Culture*, 1(2):93–100, 2021.

Judith E Fan. Drawing to learn: How producing graphical representations enhances scientific thinking. *Translational Issues in Psychological Science*, 1(2):170, 2015.

Judith E Fan, Daniel LK Yamins, and Nicholas B Turk-Browne. Common object representations for visual production and recognition. *Cognitive science*, 42(8):2670–2698, 2018.

Judith E Fan, Robert D Hawkins, Mike Wu, and Noah D Goodman. Pragmatic inference and visual abstraction enable contextual flexibility during visual communication. *Computational Brain & Behavior*, 3(1):86–101, 2020.

Masoumeh Farokhi and Masoud Hashemi. The analysis of children's drawings: social, emotional, physical, and psychological aspects. *Procedia-Social and Behavioral Sciences*, 30:2219–2224, 2011.

Nicolas Fay, Simon Garrod, Leo Roberts, and Nik Swoboda. The interactive evolution of human communication systems. *Cognitive science*, 34(3):351–386, 2010.

Li Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Conference on Computer Vision and Pattern Recognition Workshop*, pages 178–178, 2004. .

Li Fei-Fei, Asha Iyer, Christof Koch, and Pietro Perona. What do we perceive in a glance of a real-world scene? *Journal of vision*, 7(1):10–10, 2007.

Chrisantha Fernando, Daria Zenkova, Stanislav Nikolov, and Simon Osindero. From language games to drawing games. *arXiv preprint arXiv:2010.02820*, 2020.

Jonathan Fish and Stephen Scrivener. Amplifying the mind's eye: sketching and visual cognition. *Leonardo*, 23(1):117–126, 1990.

John Fiske. *Introduction to communication studies.* Routledge, 2010.

W Fitch. An invisible hand. *Nature*, 449(7163):665–667, 2007.

Luciano Floridi. Is semantic information meaningful data? *Philosophy and phenomenological research*, 70(2):351–370, 2005.

Jerry A Fodor. *The language of thought*, volume 5. Harvard university press, 1975.

Jerry A Fodor. *Representations: Philosophical essays on the foundations of cognitive science.* Mit Press, 1983.

Jerry A Fodor. *Psychosemantics: The problem of meaning in the philosophy of mind*, volume 2. MIT press, 1987.

Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *CoRR*, abs/1605.06676, 2016.

Douglas C Fox. Prehistoric rock pictures in Europe and Africa. *The Bulletin of the Museum of Modern Art*, 4(5):3–8, 1937.

Kevin Frans, Lisa B. Soros, and Olaf Witkowski. CLIPDraw: Exploring text-to-drawing synthesis through language-image encoders. *CoRR*, abs/2106.14843, 2021.

Johannes Fürnkranz. A study using n-gram features for text categorization. *Austrian Research Institute for Artifical Intelligence*, 3(1998):1–10, 1998.

Bruno Galantucci. An experimental study of the emergence of human communication systems. *Cognitive science*, 29(5):737–767, 2005.

Yaroslav Ganin, Tejas Kulkarni, Igor Babuschkin, S. M. Ali Eslami, and Oriol Vinyals. Synthesizing programs for images using reinforced adversarial learning. In *International Conference on Machine Learning*, pages 1652–1661. PMLR, 2018.

Yue Gao, Yuan Guo, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. Artistic glyph image synthesis via one-stage few-shot learning. *ACM Transactions on Graphics (TOG)*, 38(6):1–12, 2019.

Simon Garrod, Nicolas Fay, John Lee, Jon Oberlander, and Tracy MacLeod. Foundations of representation: Where might graphical symbol systems come from? *Cognitive science*, 31(6):961–987, 2007.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

Ignace J Gelb. *A study of writing.* University of Chicago Press, 1963.

James Jerome Gibson and Leonard Carmichael. *The senses considered as perceptual systems*, volume 2. Houghton Mifflin Boston, 1966.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

Gabriela Goldschmidt. Serial sketching: visual problem solving in designing. *Cybernetics and System*, 23(2):191–219, 1992.

Dorothy G Singer Roberta M Golinkoff and Kathy Hirsh-Pasek. *Play= Learning: How play motivates and enhances children's cognitive and social-emotional growth.* Oxford University Press, 2006.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

Richard Gregory. *The Intelligent Eye.* London: Weidenfeld and Nicolson, 1970.

Richard L Gregory. Eye and brain, 1966.

Richard Gross. *Psychology: The science of mind and behaviour 7th edition.* Hodder Education, 2015.

Shangmin Guo. Emergence of numeric concepts in multi-agent autonomous communication. *arXiv preprint arXiv:1911.01098*, 2019.

Yi Guo, Zhuming Zhang, Chu Han, Wenbo Hu, Chengze Li, and Tien-Tsin Wong. Deep line drawing vectorization via line subdivision and topology reconstruction. *Computer Graphics Forum*, 38(7):81–90, 2019. .

David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018.

John Haiman et al. *Talk is cheap: Sarcasm, alienation, and the evolution of language.* Oxford University Press on Demand, 1998.

Welliam Hamer and Ledy Nur Lely. Using Pictionary game to increase learners' vocabulary mastery in English language instruction. *Journal of English Education Studies*, 2(1): 43–51, 2019.

Jonathon S Hare, Paul H Lewis, Peter GB Enser, and Christine J Sandom. Mind the gap: another look at the problem of the semantic gap in image retrieval. In *Multimedia Content Analysis, Management, and Retrieval 2006*, volume 6073, page 607309. International Society for Optics and Photonics, 2006.

Ethan Harris, Daniela Mihai, and Jonathon Hare. How convolutional neural network architecture biases learned opponency and color tuning. *Neural Computation*, 33(4): 858–898, 2021.

Marc D Hauser. *The evolution of communication.* MIT press, 1996.

Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2149–2159. Curran Associates, Inc., 2017.

Robert D Hawkins, Megumi Sano, Noah D Goodman, and Judith E Fan. Visual resemblance and communicative context constrain the emergence of graphical conventions. *arXiv preprint arXiv:2109.13861*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

John M Henderson and Andrew Hollingworth. High-level scene perception. *Annual review of psychology*, 50(1):243–271, 1999.

Christopher S Henshilwood and Benoît Dubreuil. Reading the artefacts: gleaning language skills from the middle stone age in southern Africa. *The cradle of language*, 2:61–92, 2009.

A. Hertzmann. A survey of stroke-based rendering. *IEEE Computer Graphics and Applications*, 23(4):70–81, 2003. .

Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, page 453–460, New York, NY, USA, 1998. Association for Computing Machinery. ISBN 0897919998. .

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. Beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017a.

Irina Higgins, Nicolas Sonnerat, Loic Matthey, Arka Pal, Christopher P Burgess, Matko Bosnjak, Murray Shanahan, Matthew Botvinick, Demis Hassabis, and Alexander Lerchner. SCAN: Learning hierarchical compositional visual concepts. *arXiv preprint arXiv:1707.03389*, 2017b.

Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length and helmholtz free energy. In *Advances in Neural Information Processing Systems*, pages 3–10, 1994.

K Hirsh-Pasek, RM Golinkoff, and DE Eyer. Einstein never used flashcards: How our children really learn—and why they need to play more and memorize less, 2003.

Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500, 1952.

Dirk L Hoffmann, Christopher D Standish, Marcos García-Diez, Paul B Pettitt, James A Milton, João Zilhão, Javier J Alcolea-González, Pedro Cantalejo-Duarte, Hipólito Collado, Rodrigo De Balbín, et al. U-th dating of carbonate crusts reveals neandertal origin of iberian cave art. *Science*, 359(6378):912–915, 2018.

Homestyler. Homestyler 3d home design software. `https://www.homestyler.com/`, 2022. Accessed May 2022.

Ian P Howard, Brian J Rogers, et al. *Binocular vision and stereopsis.* Oxford University Press, USA, 1995.

Zhewei Huang, Wen Heng, and Shuchang Zhou. Learning to paint with model-based deep reinforcement learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8709–8718, 2019.

Johan Huizinga. *Homo ludens ils 86.* Routledge, 2014.

Glyn W Humphreys and M Jane Riddoch. *To See But Not to See: A Case Study of Visual Agnosia.* Psychology Press, 1987.

Michel Hupet and Yves Chantraine. Changes in repeated references: Collaboration or repetition effects? *Journal of psycholinguistic research*, 21(6):485–496, 1992.

James R Hurford. The evolution of the critical period for language acquisition. *Cognition*, 40(3):159–201, 1991.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*, pages 3040–3049. PMLR, 2019.

Xu Ji, João F. Henriques, and Andrea Vedaldi. Invariant information distillation for unsupervised image segmentation and clustering. *CoRR*, abs/1807.06653, 2018.

Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 26(11):3365–3385, 2019.

Gabe Johnson and Ellen Yi-Luen Do. Games for sketch data collection. In *Proceedings of the 6th eurographics symposium on sketch-based interfaces and modeling*, pages 117–123, 2009.

Philip Nicholas Johnson-Laird. *Mental models: Towards a cognitive science of language, inference, and consciousness.* Harvard University Press, 1983.

Emilio Jorge, Mikael Kågebäck, and Emil Gustavsson. Learning to play 'Guess Who?' and inventing a grounded language as a consequence. *CoRR*, abs/1611.03218, 2016.

Aleksandra Kalinowska, Elnaz Davoodi, Kory W Mathewson, Todd Murphey, and Patrick M Pilarski. Towards situated communication in multi-step interactions: Time is a key pressure in communication emergence. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2022.

Mahesh Karnani, Kimmo Pääkkönen, and Arto Annila. The physical character of information. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2107):2155–2175, 2009.

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Rhoda Kellogg. *Analyzing children's art.* McGraw-Hill Humanities, Social Sciences & World Languages, 1969.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Adam Kendon. *Gesture: Visible action as utterance.* Cambridge University Press, 2004.

Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. Entropy minimization in emergent languages. In *International Conference on Machine Learning*, pages 5220–5230. PMLR, 2020.

Mark J Kilgard and Jeff Bolz. GPU-accelerated path rendering. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012.

Andrew Kim, Maxim Ruzmaykin, Aaron Truong, and Adam Summerville. Cooperation and Codenames: Understanding natural language processing via Codenames. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 15, pages 160–166, 2019.

Byungsoo Kim, Oliver Wang, A Cengiz Öztireli, and Markus Gross. Semantic segmentation for line drawing vectorization using neural networks. In *Computer Graphics Forum*, volume 37, pages 329–338. Wiley Online Library, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, 2015.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014.

Simon Kirby. Natural language from artificial life. *Artificial life*, 8(2):185–215, 2002.

Simon Kirby and James Hurford. Learning, culture and evolution in the origin of linguistic constraints. In *Fourth European conference on artificial life*, pages 493–502. Citeseer, 1997.

Simon Kirby, Mike Dowman, and Thomas L Griffiths. Innateness and culture in the evolution of language. *Proceedings of the National Academy of Sciences*, 104(12): 5241–5245, 2007.

Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. *arXiv preprint arXiv:1905.00414*, 2019.

Satwik Kottur, José M. F. Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge 'naturally' in multi-agent dialog. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2962–2967, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, 2012.

Yu-Kun Lai, Shi-Min Hu, and Ralph R Martin. Automatic and topology-preserving gradient mesh generation for image vectorization. *ACM Transactions on Graphics (TOG)*, 28(3):1–8, 2009.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Barbara Landau, Linda B Smith, and Susan S Jones. The importance of shape in early lexical learning. *Cognitive development*, 3(3):299–321, 1988.

Angeliki Lazaridou and Marco Baroni. Emergent multi-agent communication in the deep learning era. *arXiv preprint arXiv:2006.02419*, 2020.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *International Conference on Learning Representations*, 2017.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In *International Conference on Learning Representations*, 2018.

Gregory Lecot and Bruno Levy. Ardeco: Automatic Region DEtection and COnversion. In *Symposium on Rendering*. The Eurographics Association, 2006. ISBN 3-905673-35-5. .

Yann LeCun, Corinna Cortes, and Christopher JC Burges. The MNIST database of handwritten digits. *URL http://yann. lecun. com/exdb/mnist*, 10:34, 1998.

Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *International Conference on Machine Learning*, pages 609–616. ACM, 2009.

Jason Lee, Kyunghyun Cho, Jason Weston, and Douwe Kiela. Emergent translation in multi-agent communication. *arXiv preprint arXiv:1710.06922*, 2017.

Jason Lee, Kyunghyun Cho, and Douwe Kiela. Countering language drift via visual grounding. *arXiv preprint arXiv:1909.04499*, 2019.

David K. Lewis. *Convention: A Philosophical Study*. Wiley-Blackwell, 1969.

Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? End-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*, 2017.

Fushan Li and Michael Bowling. Ease-of-teaching and language structure from emergent communication. In *Advances in Neural Information Processing Systems*, pages 15825–15835, 2019.

Ke Li, Kaiyue Pang, Jifei Song, Yi-Zhe Song, Tao Xiang, Timothy M. Hospedales, and Honggang Zhang. Universal sketch perceptual grouping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph.*, 39(6), November 2020. ISSN 0730-0301. .

Erez Lieberman, Jean-Baptiste Michel, Joe Jackson, Tina Tang, and Martin A Nowak. Quantifying the evolutionary dynamics of language. *Nature*, 449(7163):713–716, 2007.

Toru Lin, Jacob Huh, Christopher Stauffer, Ser Nam Lim, and Phillip Isola. Learning to ground multi-agent communication with autoencoders. *Advances in Neural Information Processing Systems*, 34:15230–15242, 2021.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. *CoRR*, abs/1405.0312, 2014.

Hod Lipson. Pix18, 2016. URL `http://www.pix18.com/`.

S. Liu, W. Chen, T. Li, and H. Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7707–7716, 2019. .

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. .

Sebastian Löbbers and György Fazekas. Seeing sounds, hearing shapes: A gamified study to evaluate sound-sketches. *arXiv preprint arXiv:2205.08866*, 2022.

J Locke. An essay concerning human understanding, 1690.

Raphael Gontijo Lopes, David Ha, Douglas Eck, and Jonathon Shlens. A learned representation for scalable vector graphics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7930–7939, 2019.

David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

Ryan Lowe, Jakob Foerster, Y-Lan Boureau, Joelle Pineau, and Yann Dauphin. On the pitfalls of measuring emergent communication. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 693–701. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

Ryan Lowe, Abhinav Gupta, Jakob Foerster, Douwe Kiela, and Joelle Pineau. On the interaction between supervision and self-play in emergent communication. In *International Conference on Learning Representations*, 2020.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

Janet Mann, Richard C Connor, Peter L Tyack, and Hal Whitehead. *Cetacean societies: field studies of dolphins and whales.* University of Chicago Press, 2000.

Josiah Manson and Scott Schaefer. Analytic rasterization of curves with polynomial filters. In *Computer Graphics Forum*, volume 32, pages 499–507. Wiley Online Library, 2013.

David Marr. A computational investigation into the human representation and processing of visual information. *Vision*, 1982.

David Marr, Shimon Ullman, and Tomaso Poggio. Bandpass channels, zero-crossings, and early visual information processing. *JOSA*, 69(6):914–916, 1979.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Frans Mäyrä. The contextual game experience: On the socio-cultural contexts for meaning in digital play. In *DiGRA Conference*. Citeseer, 2007.

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

David McNeill. *Gesture and thought.* University of Chicago press, 2008.

John F. J. Mellor, Eunbyung Park, Yaroslav Ganin, Igor Babuschkin, Tejas Kulkarni, Dan Rosenbaum, Andy Ballard, Theophane Weber, Oriol Vinyals, and S. M. Ali Eslami. Unsupervised doodling and painting with improved SPIRAL. *CoRR*, abs/1910.01007, 2019.

Daniela Mihai and Jonathon Hare. Differentiable drawing and sketching. *arXiv preprint arXiv:2103.16194*, 2021a.

Daniela Mihai and Jonathon Hare. Learning to draw: Emergent communication through sketching. In *Advances in Neural Information Processing Systems*, 2021b.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 2013.

Marvin Minsky. A framework for representing knowledge, 1974.

Marvin Minsky and Seymour A Papert. Artificial intelligence progress report. Technical report, MIT Artificial Intelligence Laboratory, 1972.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Haoran Mo, Edgar Simo-Serra, Chengying Gao, Changqing Zou, and Ruomei Wang. General virtual sketching framework for vector line art. *ACM Trans. Graph.*, 40(4), jul 2021. ISSN 0730-0301. .

Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337): 508–513, 2017.

Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. *CoRR*, abs/1703.04908, 2017.

Umar Riaz Muhammad, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M. Hospedales. Learning deep sketch abstraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Reiichiro Nakano. Neural painters: A learned differentiable constraint for generating brushstroke paintings. *CoRR*, abs/1904.08410, 2019.

Usman Naseem, Imran Razzak, Shah Khalid Khan, and Mukesh Prasad. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. *Transactions on Asian and Low-Resource Language Information Processing*, 20(5):1–35, 2021.

Louis Albert Necker. LXI. Observations on some remarkable optical phænomena seen in Switzerland; and on an optical phænomenon which occurs on viewing a figure of a crystal or geometrical solid. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1(5):329–337, 1832.

Ulric Neisser. Cognition and reality san francisco: Vv h. *Freeman*, 1976.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Soma Nonaka, Kei Majima, Shuntaro C Aoki, and Yukiyasu Kamitani. Brain hierarchy score: Which deep neural networks are hierarchically brain-like? *Iscience*, 24(9):103013, 2021.

Martin A Nowak and David C Krakauer. The evolution of language. *Proceedings of the National Academy of Sciences*, 96(14):8028–8033, 1999.

Aude Oliva and Philippe G Schyns. Diagnostic colors mediate scene recognition. *Cognitive psychology*, 41(2):176–210, 2000.

Alexandrina Orzan, Adrien Bousseau, Holger Winnemöller, Pascal Barla, Joëlle Thollot, and David Salesin. Diffusion curves: A vector representation for smooth-shaded images. *ACM Trans. Graph.*, 27(3):1–8, aug 2008. ISSN 0730-0301. .

Mark Pagel, Quentin D Atkinson, and Andrew Meade. Frequency of word-use predicts rates of lexical evolution throughout Indo-European history. *Nature*, 449(7163):717–720, 2007.

Sue Taylor Parker and Kathleen Rita Gibson. A developmental model for the evolution of language and intelligence in early hominids. *Behavioral and Brain sciences*, 2(3): 367–381, 1979.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

Charles Sanders Peirce. *Collected Papers of Charles Sanders Peirce*. Cambridge, MA: Harvard University Press, 1931.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Jean Piaget. *Play, dreams and imitation in childhood*. Routledge, 2013.

Steven Pinker. *The language instinct: How the mind creates language*. Penguin UK, 2003.

David Pitt. Mental Representation. In *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2020 edition, 2020.

Thomas Porter and Tom Duff. Compositing digital images. In *Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH, page 253–259. Association for Computing Machinery, 1984. ISBN 0897911385. .

Marc Prensky. Fun, play and games: What makes games engaging. *Digital game-based learning*, 5(1):5–31, 2001.

Shuwen Qiu, Sirui Xie, Lifeng Fan, Tao Gao, Song-Chun Zhu, and Yixin Zhu. Emergent graphical conventions in a visual communication game. *arXiv preprint arXiv:2111.14210*, 2021.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

Renato T Ramos. The concepts of representation and information in explanatory theories of human behavior. *Frontiers in Psychology*, 5:1034, 2014.

Pradyumna Reddy, Michael Gharbi, Michal Lukac, and Niloy J. Mitra. Im2vec: Synthesizing vector graphics without vector supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7342–7351, June 2021.

Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B Cohen, and Simon Kirby. Compositional languages emerge in a neural iterated learning model. *arXiv preprint arXiv:2002.01365*, 2020.

Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. Sketchformer: Transformer-based representation for sketched structure. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14153–14162, 2020.

John M Roberts, Malcolm J Arth, and Robert R Bush. Games in culture. *American anthropologist*, 61(4):597–605, 1959.

Andrew Robinson. *Lost languages*. McGraw Hill New York, 2002.

Guillaume Rousselet, Olivier Joubert, and Michele Fabre-Thorpe. How long to get to the “gist” of real-world natural scenes? *Visual cognition*, 12(6):852–877, 2005.

David E Rumelhart. *The architecture of mind: A connectionist approach.*, chapter 8, pages 207–238. The MIT Press, 1989.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Jacqueline Sachs, Barbara Bard, and Marie L Johnson. Language learning with restricted input: Case studies of two hearing children of deaf parents. *Applied Psycholinguistics*, 2(1):33–54, 1981.

Manish Saggar, Eve-Marie Quintin, Eliza Kienitz, Nicholas T Bott, Zhaochun Sun, Wei-Chen Hong, Yin-hsuan Chien, Ning Liu, Robert F Dougherty, Adam Royalty, et al. Pictionary-based fMRI paradigm to study the neural correlates of spontaneous improvisation and figural creativity. *Scientific reports*, 5(1):1–11, 2015.

David Salomon. *Curves and surfaces for computer graphics*. Springer Science & Business Media, 2007.

Jerry Samet and Deborah Zaitchik. Innateness and Contemporary Theories of Cognition. In *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2017 edition, 2017.

Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)*, 35 (4):1–12, 2016.

Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2017.

Ravi Kiran Sarvadevabhatla, Isht Dwivedi, Abhijat Biswas, and Sahil Manocha. Sketch-Parse: Towards rich descriptions for poorly drawn sketches using multi-task hierarchical deep networks. In *Proceedings of the 25th ACM International Conference on Multimedia*, pages 10–18, 2017.

Ravi Kiran Sarvadevabhatla, Shiv Surya, Trisha Mittal, and R Venkatesh Babu. Pictionary-style word guessing on hand-drawn object sketches: Dataset, analysis and deep network models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(1): 221–231, 2018.

Andrew M. Saxe, Pang Wei Koh, Zhenghao Chen, Maneesh Bhand, Bipin Suresh, and Andrew Y. Ng. On random weights and unsupervised feature learning. In *International Conference on Machine Learning*, pages 1089–1096. Omnipress, 2011. ISBN 978-1-4503-0619-5.

Bilge Sayim and Patrick Cavanagh. What line drawings reveal about the visual brain. *Frontiers in Human Neuroscience*, 5:118, 2011. ISSN 1662-5161. .

Denise Schmandt-Besserat. *Before writing: From counting to cuneiform*, volume 1. University of Texas Press, 1992.

Daniel L Schwartz. The emergence of abstract representations in dyad problem solving. *The Journal of the Learning Sciences*, 4(3):321–354, 1995.

Thomas W Sederberg and Geng-Zhe Chang. Isolator polynomials. In *Algebraic Geometry and Its Applications*, pages 507–512. Springer, 1994.

Peter Selinger. Potrace: a polygon-based tracing algorithm. *Potrace (online), http://potrace. sourceforge. net/potrace. pdf (2009-07-01)*, 2, 2003.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 527–538. Curran Associates, Inc., 2018.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

Claude Elwood Shannon and Warren Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, 1949. ISBN 9780252725487.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, Shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Johannes Singer, Katja Seeliger, and Martin N Hebart. The representation of object drawings and sketches in deep convolutional neural networks. In *NeurIPS 2020 Workshop SVRHM*, 2020.

Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Individualized controlled continuous communication model for multiagent cooperative and competitive tasks. In *International Conference on Learning Representations*, 2019.

Erik Sintorn and Ulf Assarsson. Hair self shadowing and transparency depth ordering using occupancy maps. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*, I3D '09, page 67–74, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584294. .

Dmitriy Smirnov, Matthew Fisher, Vladimir G Kim, Richard Zhang, and Justin Solomon. Deep parametric shape predictions using distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 561–570, 2020.

Linda Smith and Michael Gasser. The development of embodied cognition: Six lessons from babies. *Artificial life*, 11(1-2):13–29, 2005.

Paul Smolensky. *Connectionist modeling: Neural computation/mental connections.*, chapter pages 44–67. The MIT Press, 1989.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

Dan Sperber. How do we communicate. *How things are: A science toolkit for the mind*, pages 191–199, 1995.

Luc Steels. The synthetic modeling of language origins. *Evolution of communication*, 1 (1):1–34, 1997.

Luc Steels. *Experiments in cultural language evolution*, volume 3. John Benjamins Publishing, 2012.

Katherine R Storrs, Tim C Kietzmann, Alexander Walther, Johannes Mehrer, and Nikolaus Kriegeskorte. Diverse deep neural networks all predict human IT well, after training and fitting. *BioRxiv*, 2020.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. *CoRR*, abs/1605.07736, 2016.

Jian Sun, Lin Liang, Fang Wen, and Heung-Yeung Shum. Image vectorization using optimized gradient meshes. *ACM Trans. Graph.*, 26(3):11–es, jul 2007. ISSN 0730-0301. .

Ivan E Sutherland. Sketchpad a man-machine graphical communication system. *Simulation*, 2(5):R–3, 1964.

Sriram Swaminarayan and Lakshman Prasad. Rapid automated polygonal image decomposition. In *35th IEEE Applied Imagery and Pattern Recognition Workshop (AIPR'06)*, pages 28–28. IEEE, 2006.

Yingtao Tian and Jesse Engel. Latent translation: Crossing modalities by bridging generative models. *CoRR*, abs/1902.08261, 2019.

Giulio Tononi. Consciousness as integrated information: a provisional manifesto. *The Biological Bulletin*, 215(3):216–242, 2008.

Alan Mathison Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

Barbara Tversky, Masaki Suwa, Maneesh Agrawala, Julie Heiser, Chris Stolte, Pat Hanrahan, Doantam Phan, Jeff Klingner, Marie-Paule Daniel, Paul Lee, et al. Sketches for design and design of sketches. In *Human behaviour in design*, pages 79–86. Springer, 2003.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al. Starcraft II: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

H von Helmholtz. Helmholtz's treatise on physiological optics, (southall jp, transl.). *New York: Optical Society of America*, 1925.

Dirk B. Walther, Barry Chai, Eamon Caddigan, Diane M. Beck, and Li Fei-Fei. Simple line drawings suffice for functional mri decoding of natural scene categories. *Proceedings of the National Academy of Sciences*, 108(23):9661–9666, 2011. ISSN 0027-8424. .

Joseph Walton-Rivers, Piers R Williams, and Richard Bartle. The 2018 Hanabi competition. In *2019 IEEE Conference on Games*, pages 1–8. IEEE, 2019.

William Warburton. *The Divine Legation of Moses Demonstrated: On the Principles of a Religious Deist, from the Omission of the Doctrine of a Future State of Reward and Punishment in the Jewish Dispensation. In Nine Books*, volume 1. 1742.

Jason Weston and Christopher Watkins. Support vector machines for multi-class pattern recognition. In *European Symposium On Artificial Neural Networks*, pages 219–224, 01 1999.

Michael Weyrich, Jan-Philipp Schmidt, and Christof Ebert. Machine-to-machine communication. *IEEE Software*, 31(4):19–23, 2014.

Labov William. Principles of linguistic change. vol. 1: Internal factors, 1994.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. ISSN 0885-6125. .

Karl DD Willis, Pradeep Kumar Jayaraman, Joseph G Lambourne, Hang Chu, and Yewen Pu. Engineering sketch generation for computer-aided design. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2105–2114, 2021.

Georges Winkenbach and David H. Salesin. Computer-generated pen-and-ink illustration. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, page 91–100. Association for Computing Machinery, 1994. ISBN 0897916670. .

X. Wu, Y. Qi, J. Liu, and J. Yang. Sketchsegnet: A rnn model for labeling sketch strokes. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2018. .

Xiaolin Wu. An efficient antialiasing technique. *ACM SIGGRAPH Computer Graphics*, 25(4):143–152, July 1991. ISSN 0097-8930. .

Tian Xia, Binbin Liao, and Yizhou Yu. Patch-based image vectorization with automatic curvilinear feature alignment. *ACM Trans. Graph.*, 28(5):1–10, dec 2009. ISSN 0730-0301. .

Guofu Xie, Xin Sun, Xin Tong, and Derek Nowrouzezahrai. Hierarchical diffusion curves for accurate automatic image vectorization. *ACM Trans. Graph.*, 33(6), nov 2014. ISSN 0730-0301. .

Ning Xie, Hirotaka Hachiya, and Masashi Sugiyama. Artist agent: A reinforcement learning approach to automatic stroke generation in oriental ink painting. *IEICE TRANSACTIONS on Information and Systems*, 96(5):1134–1144, 2013.

Peng Xu, Chaitanya K Joshi, and Xavier Bresson. Multigraph transformer for free-hand sketch recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Peng Xu, Timothy M Hospedales, Qiyue Yin, Yi-Zhe Song, Tao Xiang, and Liang Wang. Deep learning for free-hand sketch: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23): 8619–8624, 2014.

Lumin Yang, Jiajie Zhuang, Hongbo Fu, Kun Zhou, and Youyi Zheng. SketchGCN: Semantic sketch segmentation with graph convolutional networks. *arXiv preprint arXiv:2003.00678*, 2020.

Qian Yu, Yongxin Yang, Feng Liu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Sketch-a-Net: A deep neural network that beats humans. *International journal of computer vision*, 122(3):411–425, 2017.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

Ningyuan Zheng, Yifan Jiang, and Dingjiang Huang. Strokenet: A neural painting environment. In *International Conference on Learning Representations*, 2019.

Tao Zhou, Chen Fang, Zhaowen Wang, Jimei Yang, Byungmoon Kim, Zhili Chen, Jonathan Brandt, and Demetri Terzopoulos. Learning to doodle with stroke demonstrations and Deep Q-Networks. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 13. BMVA Press, 2018.

Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. .

Zhengxia Zou, Tianyang Shi, Shuang Qiu, Yi Yuan, and Zhenwei Shi. Stylized neural painting. *CoRR*, 2020.