Expanding the Generative Space: Data-Free Techniques for Active Divergence with Generative Neural Networks

Terence Broad

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy

Department of Computing
Goldsmiths, University of London

2024

Acknowledgements

First and foremost I would like to thank my supervisors Professor Mick Grierson and Professor Frederic Fol Leymarie for their dedicated support over the long journey of this research and my prior academic training, and to the wider community of academics in the Department of Computing at Goldsmiths who supported me over my many years there. I will be eternally grateful to those at Goldsmiths who had the dedication and vision to establish creative computing as an academic discipline and create a place for people like me to thrive in.

I would like to thank Professor Rebecca Fiebrink and Professor Matthew Yee-King for the invaluable feedback in my upgrade examination, which massively influenced the final shape of this thesis. Additionally, I would like to thank the examiners of this PhD, Professor Phillipe Pasquier and Professor Marco Gillies, for their time and the thoughtful questions they prepared for what was a very enjoyable viva.

I would like to thank my friend and PhD colleague Sebastian Berns for coining the term *active divergence* that succinctly defined and tied together what I had thought were very disparate pieces of research when I had originally conducted them. I would like to thank the principal investigators, admin team and the other fellow students from the Centre for Doctoral Training in Intelligent Games and Game Intelligence (IGGI; grant EP/L015846/1) for providing the support and community to make this PhD possible. I would also like to thank Sara Bodinar and Jan Birley from Writers Retreat UK for providing the space and mentorship during my retreat there in 2023 where the final framing and narrative of this thesis took shape.

I would like to thank the curators and other organisers who chose to exhibit and showcase artworks made during this thesis, which include Luba Elliot, Xavier Snelgrove, Gemma Murray, Susanna Pousette, Paula Perissinotto, Clarissa Oliveira, Mathieu Arbez Hermoso and Rita Hajj. I would also like to thank Georgie Hoare and Joe Bedell-Brill from the band 0171 who let me use their commission as a test-bed for some of this research and document that process in this thesis.

I would like to thank my good friend Joe for his generosity and support during the first COVID lockdown and the support he gave to this research during that pivotal time. I have the upmost gratitude for my partner Katie's patience and her resolute support that I so desperately needed in the final stages of this PhD. Finally, I would finally like to thank my Mum and Dad for the constant and unwavering support they have given me throughout my academic journey, none of this would have been possible without them.

Abstract

Generative neural networks offer powerful tools for the generation of data in many domains, given their ability to model distributions of data and generate high-fidelity results. However, a major shortcoming is that they are unable to explicitly diverge from the training data in creative ways and are limited to fitting the target data distribution. This thesis presents a body of work investigating ways of training, fine-tuning, and configuring generative neural networks in inference in order to achieve data-divergent generation. This goal of configuring generative neural networks to diverge from their original training data or any existing data distribution is referred to as *active divergence*. All of the approaches presented in this thesis are data-free in their implementation, which inherently distinguishes these approaches from the traditional orthodoxy of imitation-based learning that is widespread throughout most machine learning research. The research presented in this thesis represents three categorical contributions to achieving active divergence: training without data, divergent fine-tuning, and network bending. In addition to this, a formal survey and taxonomy of active divergence methods is presented as another contribution of this thesis. The overriding goal of the research in this thesis is to *expand the generative space* of generative neural networks. All three methods presented achieve this, and point to a new approach to working with generative neural networks that does not rely on the imitation of, and derivation from data, for extracting its value and creative possibilities.

Contents

List of Figures

List of Tables

List of abbreviations

These are the abbreviations used in this thesis:

- **AI -** Artificial Intelligence
- **CNN** - Convolutional Neural Network
- **CPPN** - Composition Pattern Producing Network
- **CST** - Creativity Support Tool
- **DSP** - Digital Signal Processing
- **EP** - Extended Play Record
- **FFHQ** - Fickr-Faces High-Quality (Dataset)
- **GAN** - Generative Adversarial Network
- **GUI** - Graphical User Interface
- **GMM** - Gaussian Mixture Models
- **GNN** - Generative Neural Network
- **GPU** - Graphics Processing Unit
- **GRU** - Gated Recurrent Unit
- **HCI** - Human-Computer Interaction
- **ICCC** - International Conference on Computational Creativity
- **ICCV** - International Conference on Computer Vision
- **KLD** - Kullback-Leibler Divergence
- **KNN** - K-Nearest Neighbour
- **LLM** - Large Language Model

- **LSTM** - Long-Short Term Memory Network
- **LSUN** - Large-scale Scene UNderstanding (Dataset)
- **MIDI** - Musical Instrument Digital Interface
- **MCMC** - Markov Chain Monte Carlo
- **ML** - Machine Learning
- **MLP** - Multi-Layered Perceptron
- **MNIST** - Modified National Institute of Standards and Technology (Dataset)
- **NerIPS** - Neural Information Processing Systems
- **NFT** - Non-Fungible Token
- **PCA** - Principle Components Analysis
- **RNN** - Recurrent Neural Network
- **RL** - Reinforcement Learning
- **RLHF** - Reinforcement Learning from Human Feedback
- **SGD** - Stochastic Gradient Descent
- **t-SNE** - t-distributed Stochastic Neighbor Embedding
- **VAE** - Variational AutoEncoder
- **VJ** - Video Jockey
- **VQ-VAE** - Vector-Quatised Variational Autoencoder
- **XAI** -eXplainable AI
- **xCoAx -** Conference on Computation, Communication, Aesthetics and X

List of first-author publications

This is the list of peer-reviewed first author publications that make up the work presented in this thesis.

- Broad, T. and Grierson, M., 2019. Searching for an *(un) stable equilibrium*: experiments in training generative models without data. NeurIPS 2019 Workshop on Machine Learning for Creativity and Design.
- Broad, T., Leymarie, F.F. and Grierson, M., 2020. Amplifying the uncanny. Proceedings of the 8th Conference on Computation, Communication, Aesthetics & X (xCoAx).
- Broad, T., Leymarie, F.F. and Grierson, M., 2021. Network Bending: Expressive manipulation of deep generative models. In International Conference on Artificial Intelligence in Music, Sound, Art and Design (Evo-MUSART, Part of EvoStar) (pp. 20-36). Springer, Cham.
- Broad, T., Berns, S., Colton, S. and Grierson, M., 2021. Active Divergence with Generative Deep Learning - A Survey and Taxonomy. Proceedings of The Twelfth International Conference on Computational Creativity, ICCC'21.
- Broad, T., Leymarie, F.F. and Grierson, M., 2022. Network Bending: Expressive Manipulation of Generative Models in Multiple Domains. Entropy, 24(1), p.28.
- Broad, T., 2024. Using Generative AI as an Artistic Material: A Hacker's Guide. XAIxArts: 2nd international workshop on eXplainable AI for the Arts at the ACM Creativity and Cognition Conference.

Chapter 1

Introduction

‘A Machine Learning algorithm walks into a bar.

The bartender asks, “What’ll you have?”

The algorithm says, “What’s everyone else having?”’ [Haase, 2017]

This joke by Chet Haase, typifies what is an almost universal axiom in machine learning practice and research. Real-world data, a.k.a the ground truth, contains all the information needed for our algorithms to learn from. These algorithms should learn to mimic and imitate this data in an unquestioning and uncritical fashion, because real-world data, collected, created or labelled by humans, is all they will need to achieve the aims that we determine they should strive for.

This ethos applies to almost all machine learning research and development. In the context of generative machine learning research, imitating data has led to great success. Realistic synthesis of images [Karras et al., 2019], text [Radford et al., 2018], audio [Oord et al., 2016] and video [OpenAI, 2024] were all greatly improved through this approach. Striving for realism, however, is not necessarily always a primary creative goal. In the late 19th century, the Impressionist movement in painting was a reaction against realism and rejected the notion of striving for naturalistic representation [Venturi, 1941]. Subsequent

Modernist movements in art and literature rejected the notion that art should be used for representation entirely and moved increasingly towards abstraction or nonrepresentational forms of art [Lewis, 2007].

In the context of digital and electronic media, realism is a common goal that drives the development of new techniques and technologies, but it is not the only one. Non-photorealistic rendering is widespread in video games and computer animation [Strothotte and Schlechtweg, 2002], and underpins the success of many of the most famous games and animated feature films [Kyprianidis et al., 2012]. In music, the creative (mis)use of electronic and digital musical instruments, many of which were originally designed to imitate traditional instruments, has spawned many musical genres [McGlynn, 2017]. In addition, tools like digital audio workstations, have fundamentally changed the way that people produce, perform, and listen to music [Ashbourn and Ashbourn, 2021].

Achieving realism is not the only goal of generative Artificial Intelligence (AI) research. A number of researchers in the field use datasets of paintings or recordings of musical instruments to train their AI systems. The art collective Obvious Art famously sold an AI-generated artwork *Edmond de Belamy* at the Christies auction house for $432,500 [Christies, 2018]. The project was completed by creating a dataset of traditional Western paintings and training a generative neural network on this dataset. A new *'painting'* was created by cherry-picking a generated output from this generative model and was then digitally printed onto canvas and adorned in a gilded frame, resurrecting an antiquated practice that dates back to the 14th Century and peaked in 18th and 19th century Europe, where aesthetic and cultural value is prescribed to painted works by placing them in ornate, highly decorated frames [Kiilerich, 2001].

While training generative AI on paintings is not the same goal as achieving photorealism (though they are still imitating digitised photographic images of physical works), this type of work still aims to imitate the representations of real-world phenomena. Here, the representations of traditional hand-crafted works are being imitated, often those that have historical and cultural value.

There have been some attempts to make generative AI produce more creative outputs. Continuing with the theme of generating paintings, the Creative Adversarial Networks (CAN) algorithm was designed to create 'original artworks' with 'new styles', by training the neural network to deviate from the categories of historical art movements but to still generate images that look like paintings [Elgammal et al., 2017]. This research was released to much fanfare and was even featured in an episode of HBO's Silicon Valley sitcom [Elhoseiny, 2019]. However, Jerry Salz, the art critic for the New York Times, was less enthusiastic about the originality of the works generated by this algorithm. In a video produced for Vice magazine, he describes one of these CAN generated 'paintings' as being:

> 'Incredibly dull, generic, boring [...] If the ultimate test is could this have been made by a human, the answer is yes, it has been a thousandth to the thousandth time [...] What I feel is bored when I look at it, what I feel is a lack of originality in the idea that generated it.' [Saltz, 2018]

1.1 The Backlash Against AI Art

'NO TO AI GENERATED IMAGES' was the caption on a widely shared meme (Fig. 1.1) that was posted to ArtStation, DeviantArt and other art platforms where traditional artists would share portfolios of their work as a protest to the proliferation of AI-generated artworks using text to image models which had been trained on data harvested from these very platforms.

The outrage was levelled at developments in text-to-image models from startups such as Midjourney [2023] and StabilityAI [2023], which had been trained on large swathes of data collected from the internet, including web platforms designed for people to share their art, as a means of having an online portfolio to raise their public profile, and in many cases, marketing their work for people

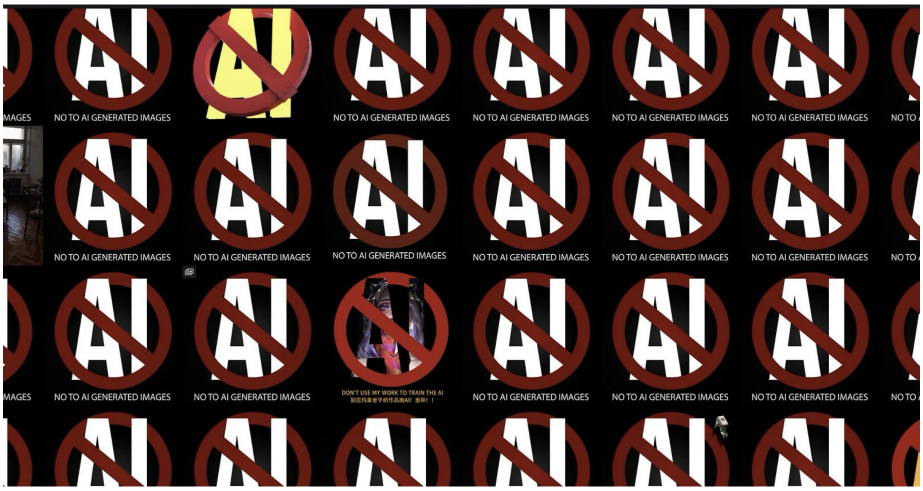

AI
NO TO AI GENERATED IMAGES
DON'T USE MY WORK TO TRAIN THE AI
别尼玛拿老子的作品跑AI！崽种！！

Figure 1.1: Screenshot from the art platform *ArtStation*, where memes with the caption ‘no to AI generated images’ were shared widely in a large backlash to generative AI from traditional creative communities.

to buy or to attract freelance work, or to gain employment as an artist, graphic designer, or illustrator.

While text-to-image models have been around for some time, the developments in 2022 with diffusion-based models such as *Dall-E 2* [OpenAI, 2022], *MidJourney v4* [Edwards, 2022] and *stable diffusion* [StabilityAI, 2022], and their ability to so successfully imitate the existing styles of individual artists, simply by listing the names of well-known creators on these digital platforms in the input text prompt, sparked outrage in the creative communities from which a lot of the data was sourced. 2022 was also the year that ChatGPT was launched [OpenAI, 2022], which catapulted Large Language Models (LLM) powered text generation into the mainstream, allowing users to quickly generate large passages of coherent text with ease. The widespread use of commercial generative AI services has already led to significant impacts in labour markets of professionals in the creative industries [Hui et al., 2024].

There has been outrage that entire bodies of individual artists’ works, and entire publishing companies’ outputs have been used for training data without consent and without remuneration. There are also legitimate and substantiated fears that these generative AI systems will put creative practitioners out of work and lower the barrier to entry for image generation so considerably to make it a trivial pursuit requiring little skill or training to produce commercially viable results. A recent statement, signed by tens of thousands of creative professionals and hundreds of creative industry organisations, expressed this sentiment in unequivocal terms:

> 'The unlicensed use of creative works for training generative AI is a major, unjust threat to the livelihoods of the people behind those works, and must not be permitted.' [aitrainingstatement.org, 2024]

Speaking as a researcher and practitioner in the direct field, it is my view the concerns and grievances of these artists are completely legitimate. I have been an active member of the 'CreativeAI' community since its early inception,[1] and it is disheartening to see tech startups entering this space and acting with such contempt for the communities of artists from which much of their value and power is sourced. The work in this thesis is positioned as an alternative to the practices of these large tech organisations. The goal of this thesis has been to find new ways of *making* with AI and find ways of creatively (mis)using these technologies in order to understand them better from the perspective of those in the arts [Salvaggio, 2023a]. Throughout the development of this PhD research, I have sought to explore how we can create using generative AI without imitating data, and in addition, how we can use generative AI to create new styles and sounds without rehashing what humans have already created, and further, how we can learn more about how generative AI works by trying to interfere with its intended operation (§5; §8.4).

[1] Though people have been doing creative things with AI since the 1960's, 'CreativeAI' refers to a self-styled community of artists, technical researchers, and hobbyists who's activities centered around the development and application of GPU-accelerated neural network based methods in artistic and creative practice. To the best of my knowledge, the term was first coined by Samim Winiger for his now defunct blog `creativeai.net` that he setup in 2015. In the British context, Luba Elliot popularised the term with the London CreativeAI meetups that she organised, of which I was the first speaker at the first event in 2016.

1.2 Intellectual Property and AI Art

My first encounters with the issues of copyright, ownership and authorship of work with generative AI predate these developments. In 2015-16, I was working towards a research Master's thesis in Creative Computing at Goldsmiths, University of London, just as generative AI research was beginning to demonstrate significant improvement in realism. In one of the experiments outlined in my Master's thesis, I used all the frames from the film *Blade Runner* as the training data for an autoencoder model, which after training I used to make a reconstruction of the film through the learned model [Broad, 2016] (Fig. 1.2). The film *Blade Runner – Autoencoded* garnered a large international interest and I was very lucky to have had the work exhibited around the world in major museums and galleries [Broad and Grierson, 2017] (Fig. 1.3).

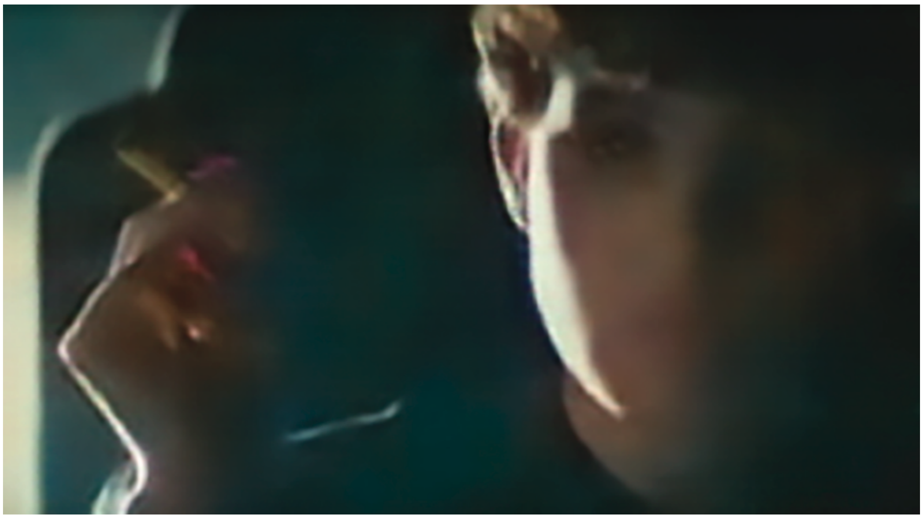


Figure 1.2: Still from *Blade Runner — Autoencoded.*

The training data, of course, was not intellectual property that I had permission to use.[2] Ironically, this project and the resulting (and later rescinded) DMCA copyright takedown notice given to the videos on the web platform

[2] It should also be noted that I was not even the first person to recreate *Blade Runner* with machine learning. Ben Bogart's work *Watching (Blade Runner)* was also created in 2016 [Bogart, 2016] (and built upon earlier research [Bogart, 2008, Bogart and Pasquier, 2013]), which I only learnt of its existence many months after completing my own recreation of the film.

Vimeo was what catapulted the work to international recognition after an account of these travails was detailed in the news website Vox [Romano, 2016].

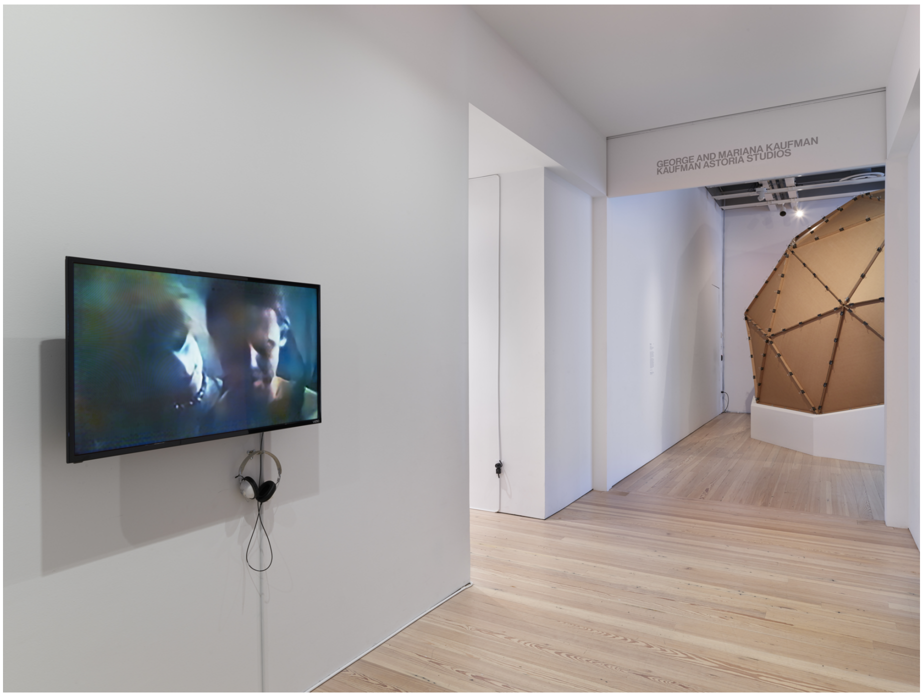


Figure 1.3: Installation view of *Dreamlands: Immersive Cinema and Art, 1905-2016* (Whitney Museum of American Art, New York, October 28, 2016-February 5, 2017). Left to right: Terence Broad, *Blade Runner - Autoencoded*, 2016; Liam Gillick, *Annlee You Proposes*, 2001. Photograph by Ron Amstutz. Image courtesy of the Whitney Museum of American Art.

Though I did not face any further legal action from Warner Brothers for disseminating the work, it was a major cause of personal stress, as I was very often anticipating some form of legal intervention (e.g. a cease and desist notice) from Warner Brothers prior to any exhibition where the work was going to be shown. An opinion published in the Columbia Journal of Law and the Arts predicted that the work would probably be dealt with as copyright infringement were it tested in an American court [Sobel, 2017].[3] I produced the work before the widespread emergence of NFTs and before there was a large market for AI-generated artworks. The money I made from exhibition fees and selling editions

[3] An alternative legal opinion was given by legal scholar Andres Guadamuz, who believed that this would be protected by fair-use or fair-dealing if it were to be tested in court on the basis of parody or pastiche [Guadamuz, 2024].

of the video work would have been relatively insignificant for a multinational media company. Nonetheless, this experience was instrumental in informing the subsequent research presented in this thesis. Finding ways of using generative AI that does not rely on data and the intellectual property of others was a key aim for the research presented in this thesis.

1.3 Motivation

My goal was to find ways of training or configuring generative AI models which did not rely on the creation of datasets to produce creative outcomes. The second was to find ways of achieving novel outcomes that did not rely on access to high-end resources, for example, those available to large technology companies, including Google DeepMind, NVIDIA, or artists such as Refik Anadol (who reportedly have access to considerable computational resources [Caulfield, 2022]). To this end, this research has focussed on exploring methods for training, configuring and customising very high-fidelity models that, when trained conventionally, require supercomputer-level resources. As such, this thesis presents a number of useful methods for manipulating, training and controlling these same models in much shorter time periods on consumer-level hardware.

Instead of relying on laboriously crafted or ethically questionable datasets to try and achieve creative outcomes, the work in this thesis details data-free methods that push the possibility space of what can be generated with contemporary neural networks. The approaches detailed are an attempt to use the intrinsic affordances of these neural networks to create original outputs that would not have been possible using any other technique or technology. The work detailed in this thesis is experimental image-making in its truest sense, and I have taken more inspiration from experimental photographers and filmmakers of the 20th Century (such as Harold Edgerton, Hiroshi Sugimoto, and Oskar Fischinger) than from academic researchers in regards to the process of enquiry used in this research.

The driving force that led to each technical breakthrough in this thesis has been technical curiosity. When considering a new possible configuration for training an AI or some other kind of intervention, if I couldn't imagine what the result of that experiment would look like, I would have to build it to find out, regardless of how many weeks or months of work it would take to get there. The results presented here are the experiments that produced the most surprising and striking results - sometimes beautiful and sometimes horrifying. There were a lot of failed experiments along the way that produced boring, predictable and uninspiring results. I've spared the reader details of most of these, apart from the few that led to key insights.

1.4 Research Methods

The research breakthroughs presented in this thesis have all come from a technological exploration of what is possible with these new technologies. Much of this research has been conducted in the vein of hacking, in its original meaning from the hacker culture at the Massachusetts Institute of Technology (MIT) in the 60s and 70s, where hacking meant 'exploring the limits of what is possible, in a spirit of playful cleverness' [Stallman, 2002]. This hacking ethos is not an approach that many people were taking in machine learning research when I started this PhD. The field was, and still is, very much dominated by orthodoxies and ideology, where theoretical mathematical underpinnings, achieving state-of-the-art performance on some widely used benchmark, and generalisation are most valued by the research communities developing these algorithms [Birhane et al., 2022].

Hacking was the primary means by which the algorithms in this thesis were discovered, but artistic exploration has also been central to the experimental work described in this thesis. When I started this PhD, my plan was to conduct primarily technical research and continue with an artistic practice on the side, maybe using some of the techniques developed in my research. Instead, it was an artistic enquiry that led me to the technical breakthroughs in the PhD, not the other way around.

In his paper *'Art in the sciences of the artificial'*, Stanley argues that in the fields of AI research and artificial life, subjective evaluation is a key driving force of progress for many researchers and practitioners in the field. There is a tendency in these research fields to discourage the dissemination of these observations in academic writing and in wider public discourse, something that Stanley worries might 'cut off some future discoverers from what could have been their inspirations' [Stanley, 2018]. In this thesis, I have sought to share my subjective position at various times in the thesis, and how that informed the direction of the following research experiments (§8.2 has a further reflection on this).

Being both guided by, and disseminating this kind of subjectivity is commonplace in research practices in many areas of the humanities, including research methods such as autoethnography [Reed-Danahay, 1997], or practice-based and practice-led research [Candy, 2006].

The goal of this research has been at its core, to advance the creative possibilities of these technologies. As a practising and internationally recognised visual artist, my subjective understanding of the visual potential and aesthetics of these systems has been one of the central guiding instruments in this research. To give any other account of how this research was conducted would be a failure of academic integrity.

The work I have done that is described in this dissertation and the contributions made in this thesis were the outcomes of practice-led research. The artistic outcomes are not presented as contributions to be assessed as outcomes of the thesis as such, but the process and practice that went into making them are described in an honest account in this thesis. Descriptions of artworks that have been made by myself and others using the techniques that have been described in this thesis are detailed in Chapter 7.

1.5 Overview and Contributions of the Thesis

The thesis is entitled *Expanding the Generative Space*. The throughline of all of the research presented here has been to find ways of going beyond the imitation of training data as the sole method for training generative neural networks. Instead, I have been trying to expand the possibility space that generative AI can produce, and the methods described in this thesis are but a few of the ways that this is possible.

1.5.1 Background

Chapter 2 provides a thorough review of relevant background literature and related research conducted prior to the work presented in this thesis. This review encapsulates both the technical aspects of machine learning relevant to this thesis, and also its application in creative contexts, whilst also drawing on the broader history of AI methods such as evolutionary algorithms, and their applications for generative processes. This chapter also describes notable prior work in relation to attempts to achieve novel outcomes with generative neural networks.

1.5.2 Training without Data

Chapter 3 documents the first peer-reviewed and published approach to training generative neural networks without data, one of the three categorical contributions to active divergence methods (§6.3.3) presented in this thesis. This work was first published in the paper *'Searching for an (un)stable equilibrium'*: experiments in training generative models without data' at the NeurIPS 2019 Workshop on Machine Learning for Creativity and Design [Broad and Grierson, 2019a].

1.5.3 Divergent Fine-Tuning

Chapter 4 documents the first peer-reviewed and published approach to divergent fine-tuning of generative AI models without relying on imitation-based learning. Divergent fine-tuning is another categorical contribution to active divergence methods (§6.3.4) presented in this thesis. This work was first published in the paper *'Amplifying the uncanny'* at the 8th Conference on Computation, Communication, Aesthetics & X (xCoAx) [Broad et al., 2020a].

1.5.4 Network Bending

Chapter 5, presents the network bending framework and is the third categorical contribution to active divergence methods presented in this thesis (§6.3.6). This work was first published in the paper *'Network Bending: Expressive manipulation of deep generative models'* at the International Conference on Artificial Intelligence in Music, Sound, Art and Design (EvoMUSART) [Broad et al., 2020b], and later extended in the paper *'Network Bending: Expressive Manipulation of Generative Models in Multiple Domains'* for the journal Entropy [Broad et al., 2021b]. Network bending has been widely reused and adopted by many other artists and researchers (detailed in §7.4 & §7.5).

1.5.5 Active Divergence Taxonomy

The final contribution of this thesis is the survey and formal taxonomy. The large majority of experimental work in this thesis falls under the umbrella term *active divergence*. This was first coined by a PhD colleague and friend, Sebastian Berns and his supervisor Simon Colton [2020]. The core experimental work in this thesis pre-dates this definition, and I am indebted to Sebastian for summarising the overarching theme of my research, which felt far more disparate when I was working on it until he was able to summarise it in a two-word definition. In collaboration with Sebastian and Simon, I expanded on this definition and the paper *'Active Divergence with Generative Deep Learning - A Survey* *and Taxonomy'* at the International Conference of Computational Creativity in 2021 [Broad et al., 2021a]. An updated summary of that survey is presented in Chapter 6 and details work completed concurrently by others during the time of this PhD to achieve similar goals.

1.5.6 Impact and Discussion

Chapter 7 details the impact of the research presented in this thesis and the subsequent work that this thesis went on to inspire. Chapter 8 reflects on the work undertaken, how artistic approaches to hacking AI models and training can lead to new forms of understanding, and how AI itself can be used as a material for artistic exploration and expression, a topic that I discussed in the paper *'Using Generative AI as an Artistic Material: A Hacker's Guide'* that I presented at the 2nd international workshop on eXplainable AI for the Arts (XAIxArts) at the ACM Creativity and Cognition Conference [Broad, 2024].

1.5.7 Conclusion

Chapter 9 concludes the thesis and reflects further on its contributions. This chapter also details the limitations of the research presented in this thesis and discusses possible future research directions to take this work further.

1.6 Summary

In the six years that I have been working on this PhD, there has been a huge amount of upheaval in the research field and its impacts on wider society. I have seen AI art and generative AI go from a small, quirky community of enthusiasts to a booming industry that has become pitted against the interests and livelihoods of the creative professionals that they are extracting value from. Hopefully, the approaches to working with AI described in this thesis can help others to find ways of using and working with generative AI which does not rely on the mass stealing and exploitation of creative professionals but instead fosters new ways for creative people to use generative AI in ways that creative people will always do: to deliberately break, misuse and adapt technologies far beyond the intended purpose to forge new forms of creative expression.

Chapter 2

Background

2.1 Introduction

This chapter serves as a survey of relevant background literature predominantly available prior to me undertaking my experimental research. The majority of the chapter outlines the technical basics of machine learning, neural networks and generative modelling. Further, the chapter also describes relevant research conducted in areas including computation, creativity, generative systems and divergent thinking prior to the advent of Graphics Processing Unit (GPU) powered deep learning circa 2011-12 [Krizhevsky et al., 2012].

2.2 Computation & Creativity

Since the advent of automated computing machines, and the idea of writing programs to give these machines instructions to follow, the idea of using computers to develop artefacts deemed creative has been long imagined. Ada Lovelace, the woman considered to be the first ever computer programmer, imagined that programmes for Charles Babbage's unfinished Analytical Engine could 'compose elaborate and scientific pieces of music of any degree of complexity or extent'. Lovelace however, did not think that computers could originate creativity themselves, declaring ‘The Analytical Engine has no pretensions whatever to originate anything. I can do [only] whatever we know how to order it to perform.’ [Lovelace, 1843].

Alan Turing took an opposing viewpoint to Lovelace on this question, stating that this objection would be better posed as ‘a machine can never take us by surprise’, countering that ‘Machines take me by surprise with great frequency [...] because I do not do sufficient calculation to decide what to expect them to do.’ [Turing, 1950] This reframing from originality to surprise shifts the emphasis from an action by the machine to an evaluation based on a human reaction. Turing develops this further, by describing a scenario called *The Imitation Game*, where a computer would be evaluated through a text channel and asked questions by an evaluator who would then attempt to differentiate whether it was human or not. If the evaluator considered the computer output to be from a human, this would be a threshold for determining simulated intelligence. This method for evaluating computational intelligence is commonly referred to as the Turing test.

In his description of the imitation game, Turing took seriously the idea of a computer being able to develop creative work. In the paper *‘Computing Machinery and Intelligence’* he muses about a machine writing a sonnet, and then, through the viva voce style of examination, being able to critically defend the work against a human interrogator based on criteria of aesthetic value, originality and of potential subjective readings of proposed changes to the language used in the work [Turing, 1950].

The idea that the bar for Artificial Intelligence (AI) is to convincingly imitate human behaviours, is one that has long been an anchor for research in the field. Imitation is central to much of how we train machine learning, neural networks and generative models, importantly imitation alone is not broadly considered a benchmark for intelligence. An alternative theoretical test for computational intelligence is the Lovelace test, where a computational program would pass the test a) it can generate an original artefact (poem, musical score, novel, idea) that can be reproduced and b) the creators of the program can not explain how it has found that solution [Bringsjord et al., 2003].

Ward [2020] argues that Turing's characterisation of Lovelace's lack of faith in the possibility for the analytical engine to produce origination (that he equates with surprise) is a mischaracterisation and misunderstanding of the debates around mechanisation and origination that were happening during the first industrial revolution. In the same notes where she makes her famous objection, she also goes on to say that the analytical engine has the power to offer new perspectives by combining theories in new ways [Lovelace, 1843]. Lovelace's remarks demonstrate the creative value of human-machine interaction, where she 'understand[s] mechanicity not as inherently creative or uncreative but as a mode through which new kinds of creativity are possible' [Ward, 2020].

2.2.1 Theories of Creative Processes

Creativity itself is broadly agreed as a well-defined concept, though there are some differences in definition. Narrower definitions of creativity refer to the cognitive processes involved in culturally understood creative activities, such as 'pieces of music, sculpture, painting, poems or other things that are taken or presented as art' [Wiggins et al., 2015]. Creativity though, is used much more broadly in common language. It can also be applied to acts, ideas or behaviours outside of the realm of art-making, such as scientific fields, sports, economic activities or even mundane, day-to-day activities.

A broader definition of creativity is that it is an act that produces something **new and original** [Kaufman and Glăveanu, 2021]. This act needs to be task-appropriate, fulfilling the requirements of whatever the original task set out. However, theories of how creativity is achieved, what facilities it, and how it is recognised and evaluated are far more disparate and less agreed upon.

Theories of what makes a person creative tend to focus on a summation of different elements. The componential model of creativity proposes that three interconnected variables are key to individual creativity. First, there are domain-relevant skills and knowledge, such as a technical skill or specific talent. Secondly, there are skills relevant to creative processes, such as a tolerance for ambiguity and a willingness to take certain risks. Finally, intrinsic motivation is needed to take part in an activity because it is enjoyable and meaningful [Amabile, 1983].

There are many other theories of creativity, pertaining to evaluating individual persons' creativity, creative collaborations, understanding traits of creative peoples and situations that best facilitate creativity and how creativity is evaluated from a historical or cultural perspective. The outline in the rest of this section will only cover theories or models of creative processes which have been developed in order to understand how to enhance and replicate creative acts, and in some cases, so that they can be partially or fully automated with computation.

2.2.1.1 Convergent and Divergent Thinking

The psychologist J.P. Guilford set out a series of traits and cognitive processes specific to creative activity. Those are ideation fluency, ideation novelty, synthesising ability and redefining ability, sensitivity to problems and evaluating ability [Guilford, 1950]. The fluency with ideas generated, the novelty of said ideas and the ability to then critically evaluate those ideas and pick the best one are some of the most important traits for creative people.[1]

Guilford later builds on this theory, expanding the thinking processes needed in creative thinking, in particular, the processes that are required for the production of creative ideas. He differentiates two kinds of productive thinking that are required for creativity; divergent and convergent thinking. Convergent thinking is the focusing of ideas down to a single correct answer. Divergent thinking is the diametric opposite, which is the ability to generate new and

[1] Notably, Guildford motivates this early research into the psychology of creativity because of the rise of *thinking machines* (aka digital computers). Imagining their eventual knock-on effect on the labour market and a future industrial revolution of intelligence being automated, Guildford muses that the only economic value left of human brains would be in the creative thinking they are capable of [Guilford, 1950]. A viewpoint I am not unsympathetic to.

different ideas. In the context of modelling creative acts, these two types of thinking are also called idea generation and idea evaluation [Guilford, 1957].

Of these two modes of productive thinking, Guilford believes divergent thinking is that which is more representative of and unique to the creative process. He considers factors of fluency, flexibility and original thinking as products of abilities in divergent thinking. Guilford's ideas about divergent thinking went on to inspire many other aspects of research, such as the Torrance test for creative thinking [Torrance, 1966].

2.2.1.2 Associative Creativity

Associative creativity is the theory that creative people or creative acts are made when connections are made between remote concepts or ideas [Mednick, 1962]. Koestler coined the term *bisociation* to describe a cognitive process where two or more concepts are combined to create a new concept [Koestler, 1964]. This model of creativity is also referred to as *combinatorial creativity* [Boden, 2004].

2.2.1.3 Evolutionary Theory of Creativity

Evolutionary theory states that the genetic structure of living beings is constantly changing through processes of random mutation and selection. Selection is carried out in two ways: **natural selection** is the process of fitness through living beings surviving long enough to reproduce sexually and transmit their genome. **Sexual selection** is the process by which organisms make preferential choices regarding which partners to mate with based on particular attributes.

An evolutionary approach to how ideas are generated and selected in creative acts was proposed by Campbell [1960], where he stated that the process of blind variation and selective retention in thought achieves innovation (aka creativity). According to Campell, this occurs through the internal emitting of thoughts, a process which lacks prescience and foresight. Campbell justifies this as a blind process, stating that 'once the process has blindly stumbled into a thought trail that "fits" the section criterion, accompanied by the "something clicked" or "Eureka" that usually marks the successful termination of the process' [Campbell, 1960].

2.2.2 Computational Creativity

Computational creativity is a subfield of AI research which investigates developing software that exhibits creative behaviours which unbiased observers would perceive as being creative [Colton and Wiggins, 2012]. Computational creativity research is usually preoccupied with artefact generation in domains that are culturally recognised as being creative, such as poetry, story generation, images or music. The mechanics of the system and how they are constructed to imitate the creative faculties of humans is the central area of exploration, whereas the quality of the generative process and the outputs from them is usually a secondary concern.

Computational creativity differentiates itself from the practice of building and evaluating creativity support tools, such as those commonly researched in the field of Human-Computer Interaction (HCI) (§2.2.5). Famously, the tagline at the 3rd International Conference on Computational Creativity in 2012 was 'scoffing at mere generation for more than a decade', though this has become an increasingly divisive phrase within the computational creativity community [Ventura, 2016].

2.2.2.1 Human-AI Co-Creation

Human-AI co-creation (also referred to as co-creativity) is a subfield of computational creativity research where the creative responsibility is shared between the software and the human interacting with it [Candy and Edmonds, 2002]. This framing positions the software as a creative collaborator, as opposed to an independent creative agent or tool only for supporting human creativity [Feldman, 2017].

2.2.3 Metacreation

Metacreation is the practice of developing software that demonstrates creative behaviour [Whitelaw, 2004]. In metacreation practice, the objective is not just to develop software, but to produce and present artistic works derived from the software, to validate their success.

Eigenfeldt et al. [2012] describe five viewpoints that should be considered when evaluating a metacreation system: (1) the designer of the system, (2) the audience for the derived artworks, (3) academic experts, (4) domain experts, (5) results from controlled experiments. This emphasis on audience evaluation and domain expertise differentiates metacreation research from computational creativity, where the emphasis is on the inherent soundness of the creative processes encoded in the system architecture [Colton, 2008].

2.2.4 Creative Computing

Creative computing is an academic discipline and is a practice-orientated approach to using and developing computing technologies to create expressive artefacts rather than something that is strictly functional [Yang and Zhang, 2016]. In creative computing, programming is the main tool that the creator uses to generate an artefact, and coding is the medium used to express human creativity. Creative coding is often carried out with creative coding frameworks, which are libraries, programming languages, or visual programming interfaces (such as node-based programming). Creative coding frameworks tend to focus on supporting visual rendering, audio processing and supporting human interaction with these frameworks.

Creative computing as an academic discipline has its roots in the Department of Computing at Goldsmiths, University of London. The first ever Creative Computing degree (BSc) was launched at Goldsmiths in 2007, it was initially designed and ran by Michael A. Casey for it's first year before being taken over by Mick Grierson for the remainder of it's fledgling years[2]. Creative computing

[2] As well as being the primary supervisor of this PhD, Mick was also my course leader and

is now an established academic discipline and taught at many universities around the world.

Clemente [2025] draw parallels between creative computing as a practice and academic discipline and the alternative computing scenes of the latter half of the 20th Century, which include *hacking* and *the demoscene*. Hacking and the creation of a *hack*, is a specific sense of creative invention with given materials in the context of electrical engineering and the academic environments researching this in the 1960s at MIT [Wark, 2006]. Though not exclusively used to describe computer code or a technical system, a hack had to 'be imbued with innovation, style and technical virtuosity' [Levy, 1984].

Hacking later became associated with the breaking of digital security and performing acts of digital trespassing and accessing confidential information, a practice that has retrospectively been called cracking. Cracking copy protection on home computer systems, for the distribution of games led to the evolution of the demoscene. In the demoscene, visual and audio programs were written and freely shared, where value was determined, for example, as follows: 'more graphical elements, more mathematical effects and more sounds made a better demo, while bugs, glitches, and irregularities made the demo worse' [Carlsson, 2019].

2.2.5 Creativity Support Tools

Creativity support tool (CST) is the term given to software programs that are designed to facilitate creative acts or enhance a user's creativity [Shneiderman, 2002a]. For CSTs, the graphical user interface (GUI) is of high importance [Shneiderman, 1999]. CSTs can be used to facilitate many varieties of tasks such as searching, visualising, consulting, thinking, exploring, composing, reviewing and disseminating [Shneiderman, 2002b].

With creativity support tools, the code is neither seen as acting in a creative

dissertation supervisor when I was an undergraduate student on the Goldsmiths BSc (and later MSci) Creative Computing programme.

way in its own right nor is it a medium for humans to be creative. Creativity support tools are independent pieces of software that can facilitate creativity but are not seen as being responsible for contributing to the creative process in their own right, as is the case with human-AI co-creativity (§2.2.2.1).

2.2.6 Computational Models of Creative Processes

There is a large existing literature on computational models that encode specific theories of creative processes, or specific attributes that are deemed essential for creative people to have. This section covers a non-exhaustive selection of these methods described in the literature.

2.2.6.1 Evolutionary Computation

Evolutionary computation refers to algorithms that are inspired by the process of biological evolution to perform some form of optimisation. The most commonly used evolutionary algorithm is the *genetic algorithm*, which is inspired by many of the processes present in biological evolution, such as random mutation, sexual selection, or (chromosomal) crossover.

Genetic algorithms require some kind of *fitness function*, that determines the quality or performance of an individual solution to whatever optimisation problem is trying to be solved. This is analogous, and in some ways similar to the *loss functions* used in machine learning algorithms (§2.3).

A genetic algorithm requires a genetic representation of the solution domain. This representation is usually a linear vector, and is often binary, with a fixed length representation, which allows for easy implementation of genetic operations such as crossover. The parameters in the genetic representation need to be carefully selected as they determine the *solution space.* Genetic algorithms are a stochastic optimisation process for exploring a *fitness landscape* and converging onto a high-quality solution [Back and Schwefel, 1996].

2.2.6.2 Novelty Search

Novelty search is an algorithm developed by Lehman and Stanley [2008], first used to guide evolutionary algorithms, where there is no set objective or fitness function. Instead, the search for novelty in the behavioural space of an evolutionary agent is the sole criterion. Lehman and Stanley [2010, 2011a,b] argue that by abandoning prescriptively defined objectives, novelty search algorithms can better search the possibility space of an evolutionary landscape, and they show that this approach can lead to both unexpected and more optimal behaviours in evolutionary agents. This approach to open-ended learning in evolutionary algorithms has inspired more recent developments in the space of open-ended reinforcement learning [Wang et al., 2020a] (§2.3.3 gives a definition of reinforcement learning).

2.2.6.3 Bayesian Surprise

Bayesian Surprise [Itti and Baldi, 2005, 2009] is an algorithm that takes inspiration from information theory and Bayesian statistics, that gives a mathematical formulation for surprise. This computational measure closely correlates with human attention, through the measurement of gaze shift of human participants watching television broadcasts.

2.2.6.4 Intrinsic Motivation

In agent-based AI modelling, intrinsic motivation is defined as goals or objectives for the agent that are not determined by external stimuli or reward functions (aka external motivation), but by the internal state of the agent itself. Intrinsic motivation can be applied in reinforcement learning agents [Chentanez et al., 2004] (§2.3.3), and is motivated by the proposition that not all objectives are universally useful [Barto, 2013]. Intrinsic motivation in AI is grounded in evolutionary theory [Singh et al., 2010], where an otherwise single mathematical function that defines a universal fitness function does not account for all behaviours and evolutionary strategies exhibited in real-world biological evolution.

2.2.6.5 Compression Progress

Compression progress [Schmidhuber, 2008] is a theoretical approach to training agents that relates to novelty search [Lehman and Stanley, 2008] and intrinsic motivation in RL agents [Chentanez et al., 2004]. Schmidhuber defines compression progress as an agent that is constantly trying to efficiently represent prior actions in an environment, whilst constantly seeking out new experiences that would satisfy an 'intrinsic curiosity reward', which would drive the agent toward seeking out novel and unpredictable experiences. Schmidhuber argues that this framework could be used for mathematical discovery, as well as art-making through 'subjective beauty'.

2.2.6.6 Combinatorial Creativity

Combinatorial creativity is the description of a creative process where two or more concepts are combined together to make new ones [Boden, 2004]. This concept is essentially a rehashing of the concept of bisociation first proposed by Koestler [1964]. Combinatorial creativity has been explored extensively in the context of computational creativity research [Zarraonandia et al., 2017, Guzdial and Riedl, 2018a,b] and in explaining the psychology of scientific discovery [Simonton, 2021, 2022].

2.2.7 Enacting Creative Processes in the Computational Arts

Many artists have explored models of creativity and creative processes through practice-based enquiry. Artists have made substantial contributions to this field and our understanding of non-human creativity, and the interactions between people and computers in exploring new possibilities for creative autonomy and creative collaboration. The rest of this section details a selection of these efforts.

2.2.7.1 Human-AI Co-Creation

Harold Cohen, the pioneering computational artist, developed and worked with the AARON program and robotic painter (using an XY plotter), to co-create paintings between 1972-2010 [Cohen, 2016]. The program underlying AARON was developed by Cohen himself [Cohen, 1995], using deterministic software that was in a constant process of development. AARON has been described as a form of meta-art (or metacreation) [McCorduck, 1991], where the artwork itself is the creation of a process that creates art.

The artist Sougwen Chung extends Cohen's work to create a practice that is centred around performative works where Chung collaboratively and interactively [Benediktsson, 2019]. Chung uses this practice to explore themes of increasing automation and co-existence between humans, algorithms and machines [Voss et al., 2021].

2.2.7.2 Evolutionary Arts

Many artists have explored the biological theory of evolution and evolutionary computation for artistic experimentation. Karl Sims seminal experiments explored evolutionary computation for the creation of computer graphics [Sims, 1991] and 3D morphology and behaviours in artificial creatures [Sims, 1994, 2023].

The artist William Latham, alongside collaborator Stephen Todd, developed the FormGrow to evolve 3D computer models resembling organic life [Latham and Todd, 1992], with the aesthetic preference of the artist guiding the evolutionary process [Lambert et al., 2013]. Using aesthetic preference to guide evolutionary systems was also utilised in the work *Cellular Forms*, where Lomas [2014] built his own user interface to interactively explore the possibility space of simulations of growing cellular organisms.

2.3 Machine Learning

Machine learning algorithms are algorithms that automatically improve from training data or experience. Given training data, a machine learning algorithm will build a model to make predictions or decisions without explicitly being told what decisions to make. Most machine learning algorithms use some form of optimisation and are optimised to minimise a loss function that is generally predetermined and task-specific. The process of optimising a machine learning model is referred to as *training*. When training a machine learning model, the loss function on the training data will be minimised with the goal of maximising the accuracy for the specific task. The model is generally evaluated on a separate test dataset that contains data not used during the training phase [Murphy, 2012].

2.3.1 Supervised Learning

Supervised learning algorithms build models based on training data that contain the desired set of output and inputs. This type of training data is referred to as *labelled data*. A labelled dataset will usually consist of pairs of data, where every input will be given with a corresponding output. Labelled datasets are in most cases hand-labelled by human users who ideally have expertise in the subject domain. Three of the most common tasks in supervised learning are *classification*, *regression* and *metric learning*.

2.3.1.1 Classification

In classification, each data sample x will be paired with a vector c that represents the class label. The machine learning model will take x as an input and output a prediction c'. The objective during training is to maximise the probability of the prediction c' will match the value of the true label c [Murphy, 2012].

2.3.1.2 Regression

In regression, each data sample x will be accompanied by an output u, where the output values are numerical values within a given range. The model will learn to output predictions p' for input x. The goal of training a regression model is to learn a model that can generalise to unseen data, where the input and output values are not necessarily present in the training data but can be inferred based on the examples given in the training data [Murphy, 2012].

2.3.1.3 Metric Learning

The goal of metric learning is to learn a distance function between given samples, that can be used to estimate how similar or dissimilar samples are. The model learns to provide a distance function $d(x, y)$ for input examples x and y. In the labelled dataset, input examples are usually accompanied by a vector label c donating the identity of the input example. When training a model, the goal is usually to minimise the distance between samples that have the same identity but maximise the distance between samples that have separate identities [Kulis et al., 2013].

2.3.2 Unsupervised Learning

Unsupervised learning algorithms find patterns in data where there are no given labels. Unsupervised learning methods find patterns and structures within data without guidance, either by learning discriminative features or capturing patterns as probability densities. The two most common tasks in unsupervised learning are *clustering*, *dimensionality reduction* and *generative modelling*.

2.3.2.1 Clustering

Clustering is the task of grouping data into clusters such that data grouped together in the same cluster are more similar to each other than data in another cluster. Examples of algorithms used for clustering are K-means, Gaussian mixture models or density-based clustering methods [Xu and Wunsch, 2005].

2.3.2.2 Dimensionality Reduction

Dimensionality reduction is the task of transforming high-dimensional data into a lower-dimensional representation that still retains key characteristics present in the original data. Examples of algorithms used for dimensionality reduction are Principle Components Analysis (PCA) [Pearson, 1901], t-Distributed Stochastic Neighbour Embedding (t-SNE) [Hinton and Roweis, 2002] and autoencoders (§2.2.1).

2.3.2.3 Generative Modelling

Generative modelling is the task of learning a function that can generate a given data distribution. A generative model will give a joint probability distribution between the observable and target variables. Approaches to generative modelling include hidden Markov models [Baum and Petrie, 1966], Gaussian mixture models [Dempster et al., 1977], and neural networks (§2.4). An overview of generative model approaches using deep neural networks is given in Section 2.5.

2.3.3 Reinforcement Learning

Reinforcement learning (RL) is a form of agent-based modelling, where the agent learns how to behave in an environment by performing actions and receiving feedback in the form of penalties or rewards [Sutton et al., 1999]. The most commonly used algorithm in RL is Q-learning, where the agent learns the value of taking a specific action in a specific state in the environment [Watkins and Dayan, 1992]. Over the course of learning, a Q-table matrix records values for each state-action pair and gradually improves the policy.

2.4 Artificial Neural Networks

Artificial neural networks are ensembles of connected units that are meant to loosely model the synaptic structure of biological neural networks. The first artificial neural network that had a learning rule updated from data was the Mark 1 Perceptron [Rosenblatt, 1958], initially a physical circuit that was designed to perform the binary classification of images captured from a sensor array. The values of the weights between connections were encoded using potentiometers with a learning rule updated with electric motors. Later the perceptron architecture was modelled in software rather than hardware, with weight parameters and values encoded as real-valued numbers. The term perceptron was later used to denote a unit (or node) within a larger network and is used to this day in larger network architectures such as Muti-Layered Perceptrons (MLP).

The term MLP usually denotes a fully connected feed-forward neural network architecture with one or more hidden layers [Rosenblatt, 1958]. However, MLP can also be used to refer to neural networks with more complex topological arrangements between nodes. MLPs can employ different non-linear activation functions, such as *tanh* [Kalman and Kwasny, 1992] and the *sigmoid* function (σ) [Han and Moraga, 1995]. A significant advance in training MLP networks came with the backpropagation algorithm (first proposed by Werbos [1974] and popularised by Rumelhart et al. [1986]), where the gradient of the loss function is calculated and backpropagated through the network graph and used to adjust the weight parameters of the networks with respect to a single input pass of the network. This learning algorithm is referred to as *gradient descent* when the learning rule is applied after performing a forward pass on every datum in the dataset, and *Stochastic Gradient Descent* (SGD) when it is applied after every sample or every batch of samples.

2.4.1 Deep Learning

Deep learning is a term used to describe artificial neural networks with many hierarchical layers that produce *deep* network structures. Earlier attempts to make neural network architectures with many layers were hampered by computational resources and limited availability and storage of data. The first major breakthrough in the efficacy of training deep generative models was to train a hierarchy of restricted Boltzmann machines [Ackley et al., 1985] as pretraining for a deep autoencoder network [Hinton and Salakhutdinov, 2006]. The first major breakthrough in efficacy and efficiency of training deep neural networks on Graphics Processing Units (GPU) was with AlexNet [Krizhevsky et al., 2012] which won the 2011 ImageNet Large-Scale Vision Recognition Challenge (LSVRC) for image classification [Russakovsky et al., 2015]. Additional breakthroughs in novel activation functions such as the Rectified Linear Unit (RELU) [Nair and Hinton, 2010], and optimisation algorithms with improved performance and stability over SGD, such as *RMSProp* [Tieleman and Hinton, 2012] and *Adam* [Kingma and Ba, 2015] were key to improving the reliability of fundamental training methods for large scale deep neural networks.

2.4.2 Neural Network Architectures

Traditionally the most commonly used architecture for neural networks was the fully connected MLP, where every node in the layer of the network (perceptron) is connected to every other node in the previous and following layers. In more complex architectures, techniques like *skip connections* are used to connect nodes from layers that have intermediate layers in between them [Raiko et al., 2012, Graves, 2013, Hermans and Schrauwen, 2013] .

There is a range of other neural network architectures that have been found to have good performance for specific domains. What follows is a discussion of some of the most common.

2.4.2.1 Convolutional Neural Networks

A Convolutional Neural Network (CNN) [Fukushima and Miyake, 1982] is a network that uses a structure of shared weights based on convolutional kernel filter functions, where the parameters of the convolutional kernel are learned. The kernel functions are repeated across the breadth of the input (in a sliding window fashion where the gap between each instance of the filter being applied is known as the *stride*), ensuring that the learned features are equivariant to translation. CNN architectures are most commonly used in a 2-dimensional (2D) form for image processing, but 1D and 3D convolutional architectures are sometimes used for processing audio and 3D voxel data. In the architectures of generative models, transposed convolutions are commonly used to iteratively upsample learned features into a high-dimensional output.

2.4.2.2 Recurrent Neural Networks

Recurrent neural networks (RNN) are neural network architectures that have connections between nodes along a temporal sequence. Connections from a previous temporal state into an existing state allow RNNs to exhibit dynamic temporal behaviour. RNNs are trained on sequential data, with activations from the previous state of the network feeding into the current state, and are able to process temporal data of variable lengths. Traditional RNNs suffer from the exploding and vanishing gradient problem, where the error signal backpropagated through the temporal state of the network has a tendency to vanish completely and prevent the network from learning or to explode and catastrophically lose information that had been acquired in training [Hochreiter, 1998]. RNN architectures such as the Long-Short Term Memory network (LSTM) [Hochreiter and Schmidhuber, 1997] or the Gated Recurrent Unit (GRU) [Cho et al., 2014] are specifically designed to avoid this problem by using gates that can retain information within the network for long periods of time and allow the network mix information from high frequency and low-frequency components.

First introduced in the context of machine translation, the attention mechanism was introduced to improve the ability of RNNs to attend to different parts of the input and output sequences when predicting the next token [Bahdanau, 2014]. This attention mechanism has become widely used in many other domains and is now central to the functioning of large language models using the transformer architecture (§2.5.1.3).

2.5 Generative Neural Networks

Generative models are neural networks that learn a set of neural network parameters that approximately model a target data distribution. This was generally seen as a difficult problem, especially for images and audio, until advances were made in core techniques (listed below) and architectures were combined. Since 2015 there has been a lot of interest in generative models from varying research areas (computer graphics, audio Digital-Signal Processing (DSP), Human-Computer Interaction (HCI)) and creative communities, artists, musicians etc, because of their ability to produce artefacts of high cultural value.

When training a generative model, a network architecture will be defined and the parameters of the network will be randomly initialised. The network will generate a sample p' given an input vector x. Over the course of training using some learning rule, generated samples are optimised to resemble samples drawn from the target distribution P, eventually leading to a set of parameters that produces the approximate distribution P'.

All deep generative models, and in particular, ones that generate high dimensional data domains like images, audio and natural language, will have some level of divergence $D(P||P') \geq 0$ between the target distribution P and the approximate distribution P', because of the complexity and stochasticity inherent in high dimensional data. The goal of all generative models is to minimise that level of divergence, by maximising the likelihood of generating the given data distribution.

2.5.1 Approaches to Modelling Data

Approaches to modelling data distributions can be separated into three categories: explicitly - where modelling the likelihood of the data distribution is learned explicitly in the objective function, such as with autoencoders, autoregressive models; approximately - where an approximation of the target distribution is learned, as is the case with variational autoencoders and reverse diffusion models; or implicitly - where the target data distribution is not modelled directly but is learned implicitly through an indirect training process as is the case with generative adversarial networks. In the following section, the varying approaches to modelling data distributions is given in more detail.

2.5.1.1 Autoencoders

An autoencoder is a symmetrical neural network that learns to reduce the dimensionality of a data domain. The first part of the network, the *encoder*, takes data x from the input domain and compresses it into a latent representation $z \in Z \in \mathbb{R}$. The other half of the network is the *decoder*, which takes the latent encoding that reconstructs the input [Kramer, 1991]. An autoencoder is trained to minimise a reconstruction loss which is usually the Mean-Squared Error (MSE). The encoder can be thought of as a learned algorithm for dimensionality reduction, and the decoder as the inverse function. Autoencoders are used for the tasks of: dimensionality reduction, representation learning and generative modelling.

The Variational AutoEncoder (VAE) [Kingma and Welling, 2013, Rezende et al., 2014] advances the traditional autoencoder. It forces a distribution on the latent variable and uses Kullback-Liebler divergence (KLD) in the loss term to penalise the encoder if the posterior distribution $q_\phi(z|x)$ deviates too far from the prior distribution $p_\theta(z)$. Noise is injected into the latent space of the VAE during training. This means a VAE models a data distribution approximately, in contrast with a traditional autoencoder which models a distribution explicitly and can only therefore model the lower bound of the log-likelihood of the data.

Equation 2.1 shows the two terms that make up the VAE loss, the KLD and reconstruction losses:

$$L(x) = -D_{KL}(q_\phi(z|x)||p_\theta(z)) + E_{q_\phi}(z|x)(log_{p_\theta}(x|z)) \tag{2.1}$$

2.5.1.2 Generative Adversarial Networks

The Generative Adversarial Networks (GAN) training framework [Goodfellow et al., 2014] is a method of training generative models without directly approximating the target data distribution. Two networks, the generator G and discriminator D are set against each other in a zero-sum mini-max training regime. The discriminator is optimised to correctly classify real samples from a training set and fake samples from the generator, where the generator is optimised to fool the discriminator into predicting its samples are real, using the value function defined in Equation 2.2:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[1 - \log D(G(z))] \tag{2.2}$$

2.5.1.3 Auto-Regressive Modelling

An autoregressive generative model learns a conditional probability of a single output element based on the preceding elements. Like RNNs, autoregressive models can be used to model sequential data of variable length, however, unlike RNNs the internal state of the network from previous time steps is not fed back into the state of the present time step. When generating from a trained autoregressive model, generated samples can be fed back into the model to generate novel continuous sequences.

The transformer architecture used foremostly in Large Language Models (LLMs) [Vaswani et al., 2017], makes use of the attention mechanism first introduced in RNNs [Bahdanau, 2014] (§2.4.2.2), but removes the recurrent part of the model architecture. In transformers, a fixed window of tokens (aka context length) is both given as input and output to the model. In the generative context, the output to the model is offset from the input by a single token, and the model is trained to autoregressively predict the next token in the output based on the input, an approach most famously used in the GPT (Generative Pre-trained Transformer) class of models [Radford et al., 2018, 2019, Brown et al., 2020].

Autoregressive modelling has also been applied to image generation. PixelCNN uses convolutional neural networks to generate images pixel-by-pixel using neighbouring pixels to determine what the next pixel should be [Van den Oord et al., 2016]. This can be used for both image in-painting, completion and unconditional image generation. VQ-VAE (Vector-Quantised Variational Autoencoder) adapts the autoencoder approach so that the encoder produces discrete representations, and the decoder uses an autoregressive approach to sampling from these discrete codes, to improve the fidelity of generated images [Van Den Oord et al., 2017].

2.5.1.4 Reverse Diffusion Generative Models

Reverse diffusion generative models (aka denoising diffusion) take inspiration from thermodynamics, where they learn to reverse the process of diffusion applied to a particular data domain [Sohl-Dickstein et al., 2015] (such as images, audio or video). During training, noise is progressively added to data, to slowly destroy the structure in the data. A neural network model is trained to invert this diffusion process. This is usually some adaption of a U-Net architecture [Ronneberger et al., 2015] which is akin to autoencoders, but with skip connections between the paired layers of the encoder and decoder. After training the model is able to unconditionally generate images from noise, by iteratively denoising them, as well as perform tasks like noise reduction and in-painting. The iterative approach to diffusion greatly enhances the fidelity and flexibility of generative models to model very diverse datasets.

2.6 Analysis of Neural Networks

Developing methods for understanding the purpose of the internal features (aka hidden units) of deep neural networks has been an ongoing area of research. In computer vision and image processing applications, there have been a number of approaches, such as through visualisation, either by sampling patches that maximise the activation of hidden units [Zeiler and Fergus, 2014, Zhou et al., 2015], or by using variations of backpropagation to generate salient image features [Zeiler and Fergus, 2014, Simonyan et al., 2013]. The *deepdream* algorithm [Mordvintsev et al., 2015] extends this approach but uses gradient-based optimisation to progressively alter images to maximise activations of certain hidden units, which gives the resemblance of psychedelic experiences [Suzuki et al., 2017]. The artist Tom White uses gradient-based optimisation to generate abstract images that resemble visual objects using image classifier networks in the series of artworks *Perception Engines* [White, 2018]. By optimising towards an ensemble of classifier networks, White [2019] is able to show that there are shared visual representations between neural networks for image recognition and human visual representations.

2.6.1 Analysis of Generative Neural Networks

Understanding and manipulating the *latent space* of generative models has subsequently been a growing area of research. Semantic latent manipulation consists of making informed alterations to the latent code that corresponds to the manipulation of different semantic properties present in the data. This can be done by operating directly on the latent codes [Brock et al., 2016, Shen et al., 2020] or by analysing the activation space of latent codes to discover interpretable directions of manipulation in latent space [Härkönen et al., 2020]. Evolutionary methods have been applied to the problem of searching and mapping the latent space [Bontrager et al., 2018, Fernandes et al., 2020] and interactive evolutionary interfaces have also been built to operate on the latent codes

[Simon, 2018] for human users to explore and generate samples from generative models.

GAN dissection [Bau et al., 2018] is an algorithm where individual convolutional features in a GAN's generator are ablated one at a time. The generated outputs are processed by a bounding box detector trained on the ADE20K Scene dataset [Zhou et al., 2017], which led to the identification of a number of units associated with the generating of certain aspects of the scene. This approach has since been adapted for music generation [Brink, 2019].

2.7 Creative Practice with Generative Neural Networks

Generative neural networks have become widely adopted in creative practice through individual artistic practices and in interactive applications and installation artworks. These are detailed in the following sections.

2.7.1 AI-Art

Artists have been experimenting with and creating artistic practices with AI in its various incarnations since the 1970s, with artists such as Harold Cohen, Peter Beyls, and Naoko Tosa making art with classical forms of AI techniques [Grba, 2022]. Since the advent of modern generative AI characterised by the use of deep learning approaches to building and training neural networks, there have been many artists adopting these technologies in their creative practice from 2015 onwards (myself included).

Artists such as Helena Sarin, Anna Ridler, Gene Kogan, Mario Klingemann, Derrick Schultz, Tom White, Jake Elwes, Scott Eaton, and Sofia Crespo have all developed artistic practices centred around generative AI, often using these methods to create art around a particular social or environmental theme [Grba, 2022]. In many of these artistic practices, artists use generative models as artistic tools or as artistic materials (§8.3 has a further discussion on this approach), which are playfully experimented with in order to craft unique forms of expression and artistic styles. Often, artists will create their own custom datasets that they use to train individual models or sets of models that are chained together in order to produce unique artistic styles (§2.8.4; §6.3.5).

Many of these artists have created work under the 'CreativeAI' banner, a subcommunity of creative practitioners using early deep learning technologies like generative AI, as well as related techniques like *deepdream* and style transfer [Gatys et al., 2016] to make artworks. Lots of this work was disseminated on social media channels such as Twitter and Instagram, as well as in online digital art galleries such as the NeurIPS Creativity Workshop AI-Art Gallery[3] and Computer Vision Art Galleries[4], primarily curated and organised by the curator Luba Elliot. Many of these artists also disseminated and sold artworks on blockchain-based NFT (non-fungible token) art platforms like *hic et nunc*[5], *SuperRare*[6], *Foundation*[7], and *Feral File*[8]. It is now common for mainstream AI conferences to have their own art galleries showcasing AI-art or CreativeAI tracks, such as the CVPR AI art gallery[9], the SIGGRAPH[10] and SIGGRAPH Asia Art Galleries[11] or the NeurIPS CreativeAI track[12].

2.7.2 Interacting with Generative Neural Networks

This section details a number of projects (interactive installations and creativity support tools) that allow for users to directly interact with generative neural networks.

[3] https://www.aiartonline.com/

[4] https://computervisionart.com/

[5] https://hicetnunc.art/

[6] https://superrare.com/

[7] https://foundation.app/

[8] https://feralfile.com/

[9] https://thecvf-art.com/

[10] https://s2024.siggraph.org/program/art-gallery/

[11] https://asia.siggraph.org/2024/submissions/art-gallery/

[12] https://neurips.cc/Conferences/2023/CallForCreativeAI

2.7.2.1 GAN Paint

An interactive interface built upon the GAN Dissection approach [Bau et al., 2018] was presented with the GANPaint framework in 2019 [Bau et al., 2019]. This allows users to 'paint' onto an input image in order to edit and control the spatial formation of hand-picked features generated by the GAN.

2.7.2.2 GANBreeder

GANBreeder (now rebranded as ArtBreeder) [Simon, 2020] is a platform that allows users to collaboratively explore the latent space of a GAN. GANBreeder was directly inspired by the PicBreeder experiment [Secretan et al., 2008, 2011] which was an online platform that allowed users to collaboratively explore the generative space of an early form of generative neural network, Composition Pattern-Producing Networks (CPPNs) [Stanley, 2007]. Both of these approaches provide a collaborative, interactive evolutionary approach to exploring the generative space of neural networks. In PicBreeder this generative space is determined by the architecture of CPPNs, whereas in GANBreeder it was determined by the latent space of GANs. In GANBreeder, users could interactively mutate, and cross-breed latent codes, creating new generative samples that are published on a shared collaborative platform. The whole network of prior generations from the user and other users can be seen and interacted with.

2.7.2.3 Learning to See

Learning to see is a series of artworks by the artist and researcher Memo Akten [Akten et al., 2019, Celis Bueno and Schultz Abarca, 2021]. The artworks are presented as interactive installations. In *Learning to See: Hello, World!* the process of learning is performed in real time, using the live camera feed of the user as the training dataset, training a VAE [Akten, 2017a]. In *Learning to See: Interactive*, pretrained image-to-image translation model (presumable CycleGAN [Zhu et al., 2017], though the exact approach is not specified in Akten et al. [2019]) is used in conjunction with a live webcam feed pointed at mundane objects (such cloth, cables, and car keys). The image translation model takes the live feed of mundane objects and outputs a scene from whatever domain the image translation model was trained to output (such as oceans, flowers, and outer space) [Akten, 2017b].

2.7.2.4 Interactive Text Generation with RNN Ensembles

Akten and Grierson [2016] present a framework for real-time interactive text generation using an ensemble of recurrent neural networks. Here the outputs of many different models trained on different datasets (the Bible, the collected works of Aristotle, Jane Austen and Charles Baudelaire) are then interactively and fused together in a generative ensemble. Here the probability of outputs for the next character are interpolated based on the specified mix by the user (this project is also detailed in §6.3.7).

2.7.2.5 Text-to-Image Generation

Text-to-image generation using generative neural networks was first presented by Mansimov et al. [2015]. They utilised the Deep Recurrent Autogressive Writer (DRAW) [Gregor et al., 2015], which is a recurrent neural network that utilises attention layers for generating images in an autoregressive fashion. Mansimov et al. [2015] train this model on the MSCOCO dataset (Microsoft Common Objects in Context) [Lin et al., 2014], which provides pairs of images with text captions, to condition a DRAW network on the captions. After training, it is then possible to generate new images on image captions.

Conditioning the generation of an autoregressive generative model with an auxiliary network that can give a distance metric function for images and text was developed as part of the DALL-E model [Ramesh et al., 2021]. Here they condition the training and generation of a VQ-VAE [Razavi et al., 2019] with CLIP (Contrastive Language-Image Pretraining) [Radford et al., 2021]. Later approaches of text-to-image generation utilise CLIP conditioning for training and generation of diffusion and latent diffusion models (§2.5.1.4) for high fidelity text-to-image generation [Rombach et al., 2022].[13]

2.8 Data Divergent Generation with Generative Neural Networks

This section details data-divergent generation with generative neural networks. This section is limited to methods that predate the research conducted in this thesis. A full account of data-divergent generation with generative neural networks (aka active divergence) is given in Chapter 6.

2.8.1 Novelty Generation with Imitation Based Generative Modelling

Kazakçı et al. [2016] present an algorithm that performs novelty search over the learned generated samples of a sparse autoencoder trained on the MNIST (Modified National Institute of Standards and Technology) dataset of handwritten digits [LeCun et al., 1998]. After training the autoencoder, they generate and map out the entire generative space, then use clustering algorithms to cluster the latent space into discrete clusters based on visual similarity, and then disregard clusters that map to existing labelled classes in the training dataset, leaving only clusters that map to new modalities of generation not present in the original training set (this approach is covered in more detail in the active divergence survey §6.3.1). [Cherti et al., 2017] extend this approach to training a class-conditional generative model. They utilise hold-out classes, which automatically capture these modalities not present in the training set. This allows for these novel modalities to be generated without the need for searching the latent space (this approach is covered in more detail in the active divergence survey §6.3.2).

[13] Many of these developments in text-to-image models occured after most of the original experimental work in this thesis was conducted.

2.8.2 Creative Adversarial Networks

In the Creative Adversarial Networks (CAN) framework, Elgammal et al. [2017] train a class-conditional generative model (using GAN-based adversarial training [Goodfellow et al., 2014]) on the wikiArt dataset [Saleh and Elgammal, 2016]. The generator network is optimised to diverge from existing art styles and generate samples that sit within these existing art styles to generate new 'artworks' that have their own distinct styles that sit outside the art-historical framework. This approach is inspired by Martinale's theory of artistic change [Martindale, 1990] where artists push against the habituation of existing artistic styles, yet still aim to minimise negative reactions from observers. The CAN framework is discussed further in Section 6.3.2 in the discussion of its context in other approaches of active divergence.

2.8.3 CombiNets

The CombiNets framework [Guzdial and Riedl, 2018a] is inspired by combinatorial creativity [Boden, 2004] where the learned parameters of multiple neural networks are combined together in order to create a new network with parameters from multiple networks. This is done in a directed fashion with new samples outside of the training datasets of either of the existing models; for instance, a mythical creature like a pegasus, which combines characteristics of two real animals (horse and bird).

2.8.4 Data Divergent Generation in Creative Practice

One approach that many artists take to diverge from existing training data, and create bespoke artistic styles is to chain multiple models together. This is commonly done by combining unconditional generative models (such as GANs or VAEs) with image-to-image translation models (such as Pix2Pix [Isola et al., 2017] or CycleGAN [Zhu et al., 2017]) and other deep learning approaches such as style-transfer [Gatys et al., 2016]. Artists like Helena Sarin use this to great effect to create unique artistic styles by combining many custom-trained generative models on their own hand-crafted datasets [Sarin, 2018], deliberately utilising the imperfections of generative models to enhance their unique artistic style. This approach is detailed further in the active divergence survey as the 'chaining models' approach (§6.3.5).

Another approach to data divergent generation comes from Mario Kinglemann in the project *Neural glitch* [Klingemann, 2018]. Here Klingemann deliberately corrupts the learned weights of a generative model, by randomly deleting (zero-ing out) or swapping the learned parameters of different filters and layers within trained GAN models (this is further detailed in §6.3.8 & 8.3).

2.9 Summary

This chapter has surveyed the relevant background literature which was predominantly available prior to me undertaking my experimental research, in order to provide sufficient background for the investigation. The next three chapters will document the experimental work that I have undertaken for this thesis.

Chapter 3

Searching for an *(un)stable equilibrium*: Experiments in Training Generative Neural Networks Without Data

3.1 Introduction

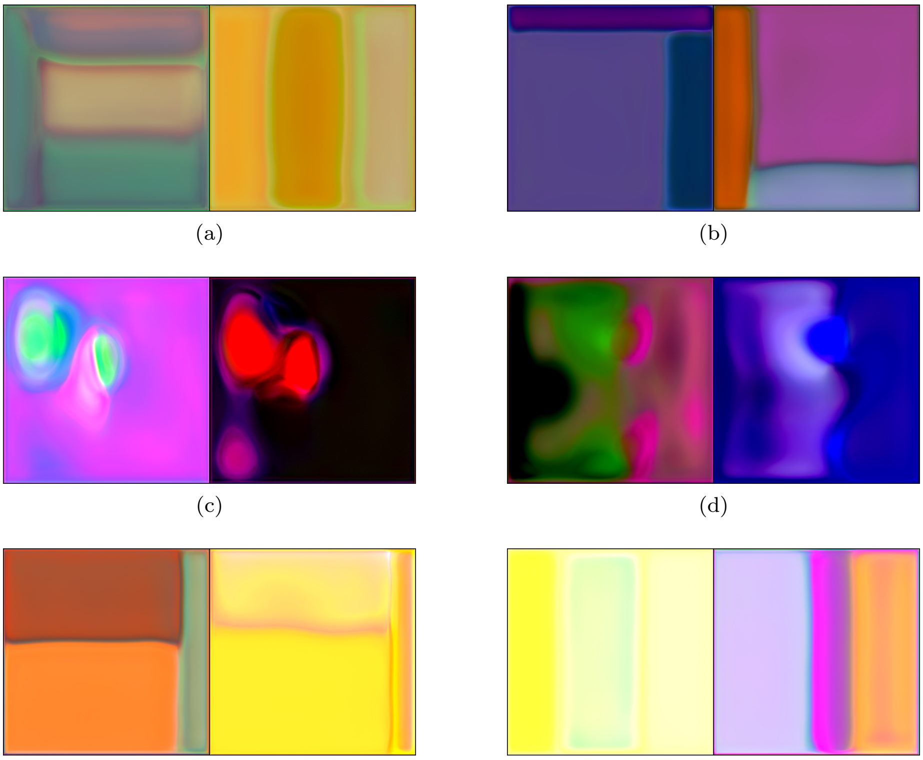
This chapter details the first successful attempt in my PhD research to find a way of training a generative neural network in a data-divergent (or data-agnostic) way. This work was the first published and peer-reviewed approach for training generative neural networks without data. The original experiments were published as a short paper at the NerIPS 2019 Workshop on Machine Learning for Creativity and Design, in Vancouver, Canada [Broad and Grierson, 2019a]. I also presented this work again at the Colors of AI workshop at the International Conference of Computational Creativity in Jönkoping in 2024 [Riccio and Schaerf, 2024]. Section 7.2 details the reception of the series of artworks *(un)stable equilibrium* that resulted from these experiments.

In that initial paper, I documented 6 experiments, the outputs of each training run becoming the artworks *(un)stable equilibrium 1:1* through to *(un)stable equilibrium 1:6* (Fig. 3.1). Unfortunately due to the passage of time and multiple computer and disk drive failures, I have lost the original logs and training samples from these original training runs. Therefore the experiments were rerun. This has meant that more data could be logged, and more variations of parameter settings and loss functions could be compared. The results are similar, but not identical to the ones in the original paper, as it was not possible to perfectly replicate these original experiments without the same initial training parameters, hyperparameters and sampling schedule. Section 7.2 details the reception and artistic presentation of the original training runs in more detail.

3.2 Motivation

This work came out of a deep frustration early on in my PhD journey, where I was trying to find ways of training generative models without modelling data. It took me far longer than it should have to come to the realisation that this is an oxymoron. That a generative model is simply a statistical model of a given data distribution, and nothing more. Any attempt to move past this notion needed a completely different approach, and the first approach I developed which was fruitful is presented in this chapter.

This work came out of a simple proposition: if it was possible to find a way of training a generative neural network without any training data, then by default, any outcome must be novel and could not resemble an existing training data distribution. There was little mathematical grounding to this approach. Through dogged trial and error, and some playful reconfiguring of the most common (at the time) way of training generative models, GANs, I was able to



(a) (b) (c) (d) (e) (f)

Figure 3.1: Original experiments in the *(un)stable equilibrium* series. (a) *(un)stable equilibrium 1:1*, (b) *(un)stable equilibrium 1:2*, (c) *(un)stable equilibrium 1:3*, (d) *(un)stable equilibrium 1:4*, (e) *(un)stable equilibrium 1:5*, (f) *(un)stable equilibrium 1:6*.

find a way to train generative networks without data, in ways that produced at the very least, aesthetically interesting outcomes.

From the early experiments with configurations of models and loss functions that led to unremarkable results, through to the final configuration of models and training runs that resulted in the artworks *(un)stable equilibrium*, this chapter is named as such because the journey I went on in producing those works was one of searching – through intuition and aesthetic exploration – for a (un)stable equilibrium. This is a balancing between finding a system just chaotic enough to produce enough randomness in the resulting training run that enough unpredictable dynamics would lead to the configuration of the weight parameters of the model such that unpredictable (and aesthetically compelling) results would come from the generator networks, but not enough for the loss functions to explode during training. The visual results of the training were monitored, both visually as I inspected the generated output as each training increment progressed, alongside the fluctuations of the various loss functions throughout training. Over the course of a couple of intensive weeks of working in this unorthodox way, the configuration that produced these results was discovered.

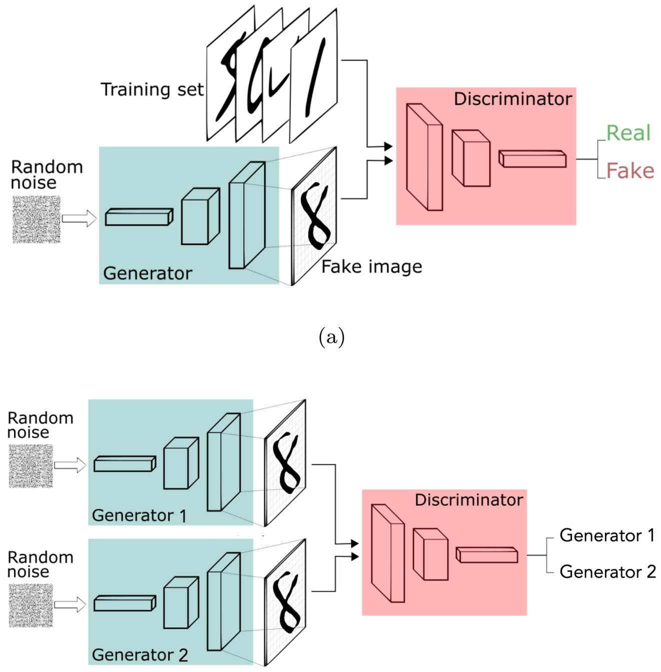
3.3 Initial Experiments

My first experiment took inspiration from the GAN framework (Fig. 3.2a) where a generator network imitates a training dataset, and the discriminator tries to tell them apart (Eq. 3.1).

$$Adv = \min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)]) + \mathbb{E}_{z\sim p_z(z)}[1 - \log D(G(z))] \tag{3.1}$$

My initial adaptation of this was to replace the training data with another generator (Fig. 3.2b). In this arrangement, the two generators are trying to imitate each other, whilst the discriminator is still trying to tell them apart (Eq. 3.2).

$$Adv = \min_{G_1}\max_{G_2}\max_{D} \mathbb{E}_{z\sim p_z(z)}[\log D(G_2(z))] + [1 - \log D(G_1(z))] \tag{3.2}$$



(a)

(b)

Figure 3.2: GAN architecture training diagrams. (a) The original GAN training framework where one generator imitates training data [Goodfellow et al., 2014]. (b) Novel training architecture where the training data is replaced with another generator in order to train two generative neural networks without data.

3.3.1 Adversarial Loss

Figures 3.3 & 3.4 show the samples during training and loss logs during training respectively. These experiments were done with the progressive growing GAN approach [Karras et al., 2017] used in the original StyleGAN [Karras et al., 2019]. The generator networks start training at a resolution of 32x32 pixels, and double in resolution for five steps, until reaching the final resolution of 512x512. Each generator is trained for 300 iterations at each resolution, totalling 1500 iterations for the full course of training. In all of these experiments, a fixed batch size of 10 was used for all resolutions in training (this was the largest I could perform at full resolution on an NVIDIA RTX 3090). The Adam optimiser was used for training [Kingma and Ba, 2014], with a learning rate of 0.01, and beta's $b_1 = 0, b_2 = 0.99$.

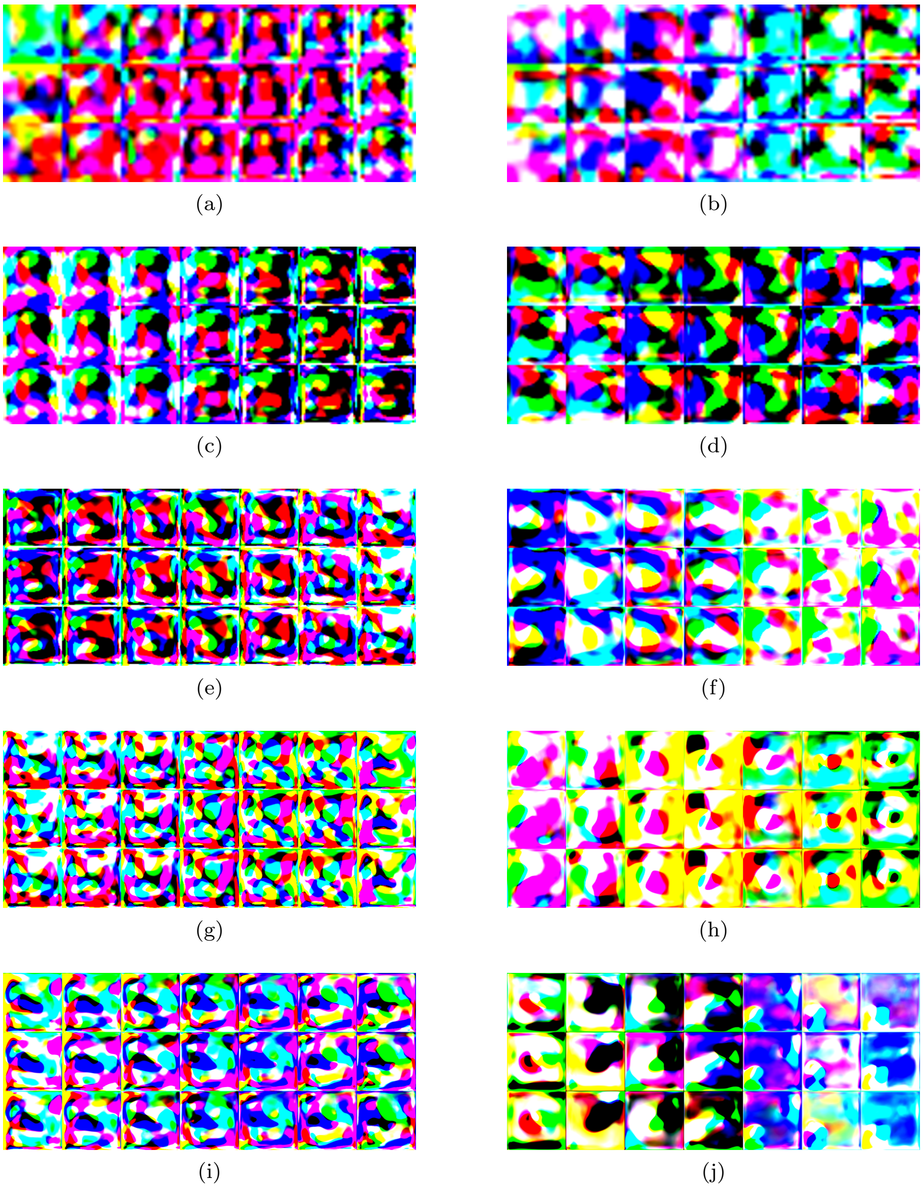
In this configuration, the visual results (Fig. 3.3) were not what I was hoping for. There is little diversity across the training batch for both generators. Essentially, this training regime suffered from mode collapse, which is a common failure state for GANs [Thanh-Tung and Tran, 2020].

In the normal GAN training regime, the diversity in the output images comes from mimicking the training data, which should have a wide variety of data samples in it (aka modes). Without training data, it is not surprising that these models quickly collapse to a single mode of generation for each of the respective generators. To overcome this I needed to add an additional term to force the model into producing more diverse outputs. This is detailed in the following subsection.

3.3.2 Colour Variance Loss Term

In response to the mode collapse suffered in the previous experiment I developed an additional term for the loss functions of the two generators. The additional loss term was designed to force increased colours to be used to measure the batch-wide variance of values for each pixel, in each channel of the tensor. This term calculates the variance across the sampled batch B for the respective channel c of the tensor for the samples drawn from both generators g_1 and g_2, which are then subtracted from each other (depending on which loss is being calculated for which generator) to enforce a relative variance that is higher than the other model across the different channels of the sample tensor.

$$Vdiff = Var(B^c_{g_1}) - Var(B^c_{g_2}) \tag{3.3}$$



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 3.3: Training samples for standard adversarial training with two generators, sampled at increments of 50 iterations. The left column shows the training samples for G_1 and the right column shows the training samples for G_2. (a,b) Training samples at resolution 32x32 for iterations 0-300. (c,d) Training samples at resolution 64x64 for iterations 300-600. (e,f) Training samples at a resolution of 128x128 for iterations 600-900. (g,h) Training samples at resolution 256x256 at iterations 900-1200. (i,j) Training samples at resolution 512x512 for iterations 1200-1500.

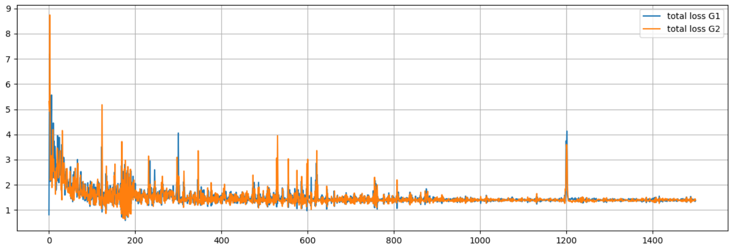


Figure 3.4: Loss plot for double adversarial training of the two generator networks. Note: the loss for the discriminator is the same as the second generator.

This is calculated for the 4 channels of the tensor present in the sample batch. The first channel is the overall batch bt, and the following channels are the colour channels of the output images: red r, green g and blue b.

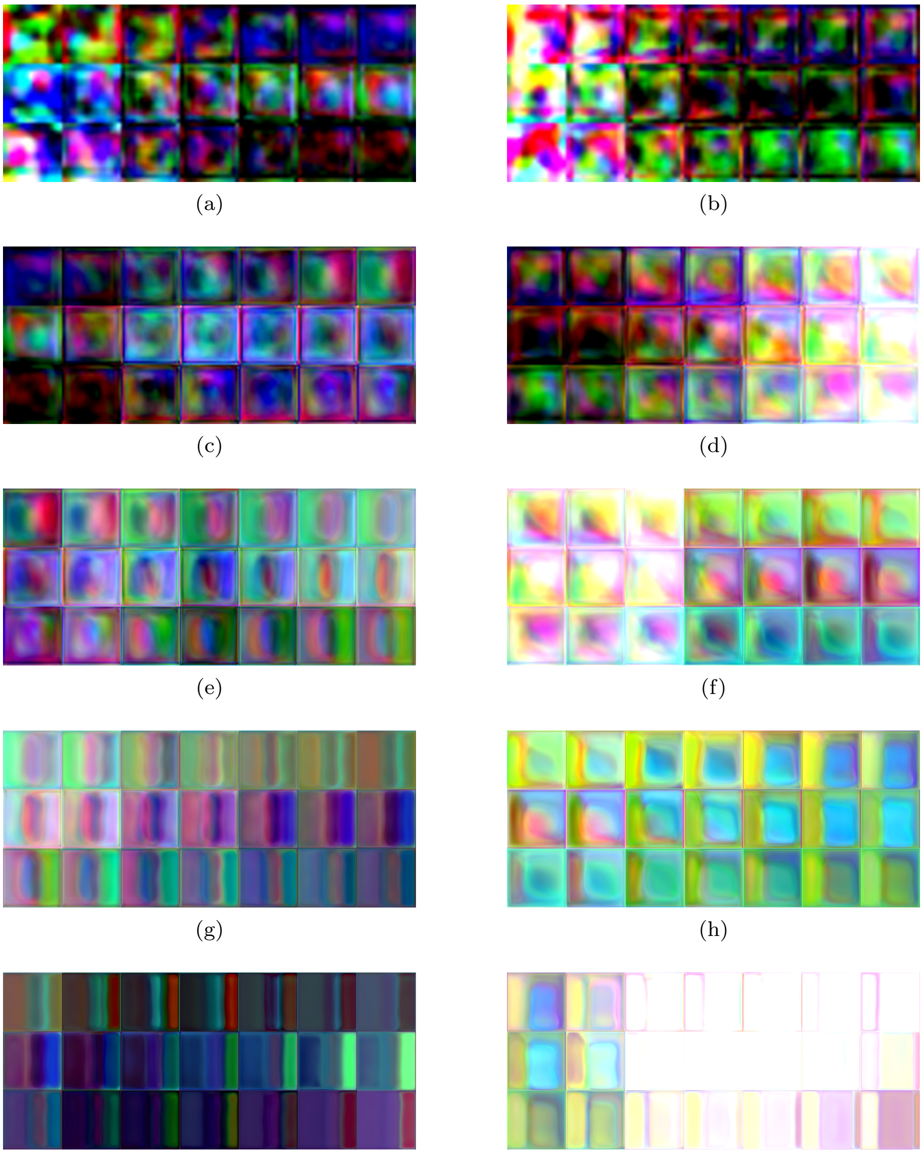
$$Vdiff(B^{bt}_{g_1}, B^{bt}_{g_2}) + Vdiff(B^{r}_{g_1}, B^{r}_{g_2}) + Vdiff(B^{g}_{g_1}, B^{g}_{g_2}) + dVdiff(B^{b}_{g_1}, B^{b}_{g_2}) \tag{3.4}$$

The loss penalty was calculated for each generator with respect to the other. Therefore each generator was optimised to have more mini-batch variance than the other. Adding this term had a significant impact on training. Equation 3.5 shows the training objective for G_1 and Equation 3.6 shows the training objective for G_2.

$$G_1\ loss = \min_{G_1} Adv + Var(B^c_{g_1}) - Var(B^c_{g_2}) \tag{3.5}$$

$$G_2\ loss = \max_{G_2} Adv + Var(B^c_{g_2}) - Var(B^c_{g_1}) \tag{3.6}$$

Figures 3.5 & 3.6 show the samples during training and loss logs during training respectively.


(a)

(b)

(c)

(d)

(e)

(f)

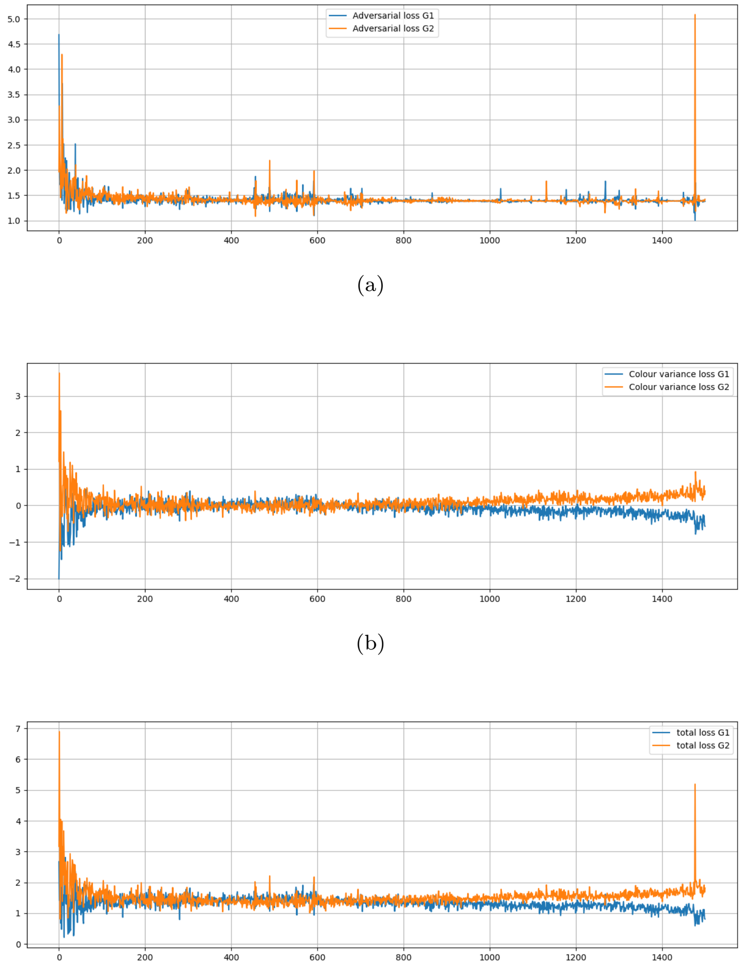
(g)

(h)

(i)

(j)

Figure 3.5: Training samples for adversarial training with two generators with the colour variance loss, sampled at increments of 50 iterations. The left column shows the training samples for G_1 and the right column shows the training samples for G_2. (a,b) Training samples at resolution 32x32 for iterations 0-300. (c,d) Training samples at resolution 64x64 for iterations 300-600. (e,f) Training samples at a resolution of 128x128 for iterations 600-900. (g,h) Training samples at resolution 256x256 at iterations 900-1200. (i,j) Training samples at resolution 512x512 for iterations 1200-1500.



(a)

(b)

(c)

Figure 3.6: Loss plot for double adversarial training with colour variation loss term. (a) Adversarial loss terms for both respective generators. (b) Colour variation loss term for both respective generators. (c) Total loss combining both terms for both respective generators.

The visual results (Fig. 3.5) with the additional loss term are much improved with respect to the variety across the generated data distribution when compared to the previous experiment without the colour variance loss term (Fig. 3.3). Whilst the visual results are simple in their composition, there is variety in colours generated across the generative space of both generators, which is exactly what I had intended would happen with this additional loss term.

In the following experiments, I take these ideas further, keeping the colour diversity loss term, but replacing the adversarial loss terms with other means of measuring distance and difference using commonly used in metric learning (§2.3.1.3).

3.4 Distance Functions

In these further experiments, I replace the adversarial loss with other means of measuring difference and distance using two common loss functions in machine learning, that come from metric learning. To calculate these losses efficiently the discriminator is kept in place, but here the discriminator network is used to calculate feature vector embeddings of each of the generated samples. Pair-wise distances are calculated from the generated sample from the respective generators, where both generators are sampled using the same fixed latent during training. The two pairwise distance loss functions used are the cosine distance and Euclidean distance, which are detailed in the following two subsections.

3.4.1 Cosine Distance

The cosine distance between two vectors is defined as the inverse of the cosine similarity (Eq. 3.7), which is used in lieu of the adversarial loss.

$$Cosine\ distance(\vec{u}, \vec{v}) = 1 - \frac{\vec{u} \cdot \vec{u}}{|\vec{u}||\vec{v}|} \tag{3.7}$$

The mean of the cosine distance is taken for the vector embeddings from the discriminator $\vec{d}$ of each respective generator $\vec{d}_{g_1}, \vec{d}_{g_2}$. This is calculated across the mini-batch $\vec{d}_{bg_1}, \vec{d}_{bg_2}$ and the mean is taken to calculate the loss for the mini-batch. The total loss for both generators is given in Eqs. 3.8 & 3.9.

$$G_1 \; loss = \overline{1 - \frac{\vec{d}_{bg_1} \cdot \vec{d}_{bg_2}}{|\vec{d}_{bg_1}||\vec{d}_{bg_2}|}} + Var(B^c_{g_1}) - Var(B^c_{g_2}) \tag{3.8}$$

$$G_2 \; loss = \overline{1 - \frac{\vec{d}_{bg_2} \cdot \vec{d}_{bg_1}}{|\vec{d}_{bg_2}||\vec{d}_{bg_1}|}} + Var(B^c_{g_2}) - Var(B^c_{g_1}) \tag{3.9}$$

The samples from training in this experiment are given in Figure 3.7 and the loss plots are given in 3.8.

3.4.2 Euclidean Distance

The Euclidean distance between two vectors is shown in Equation 3.10.

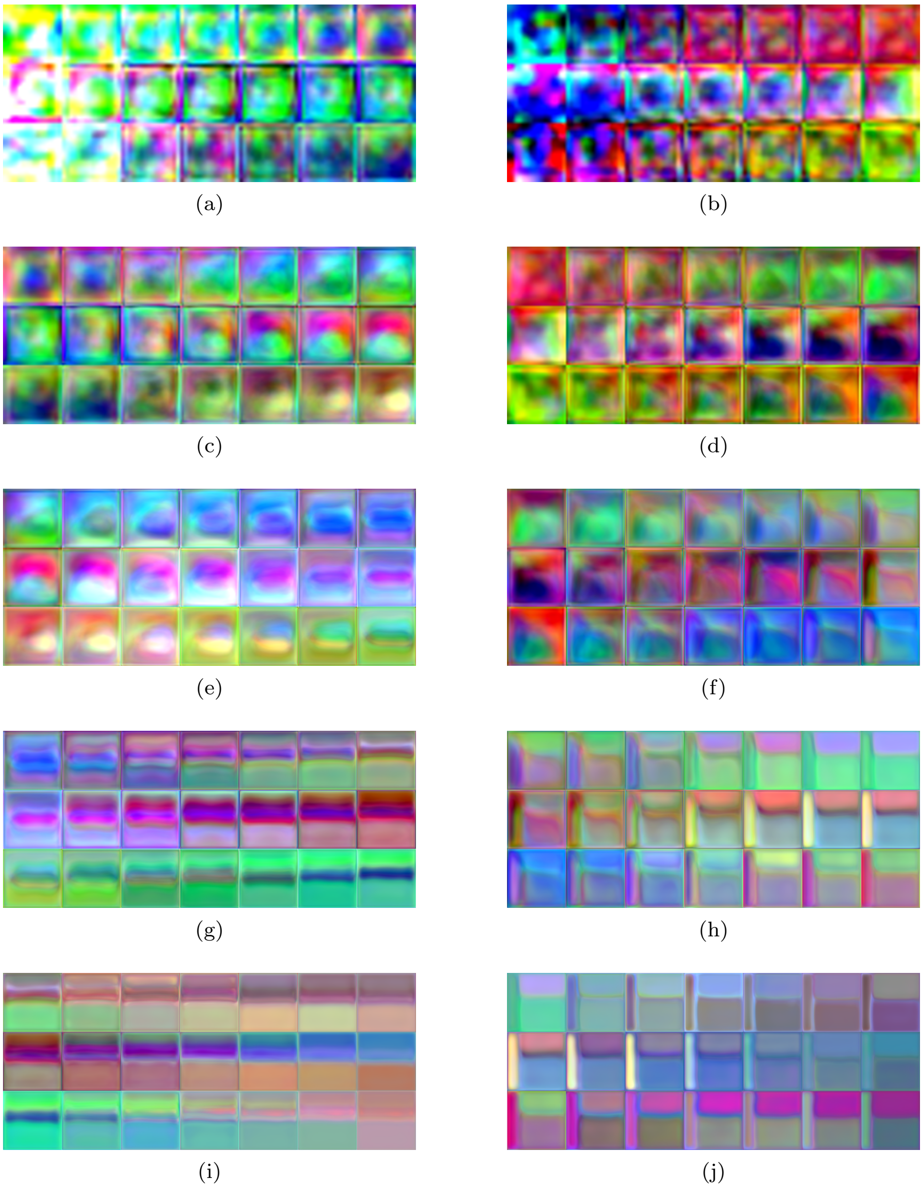
$$Euclidean \; distance(\vec{u}, \vec{v}) = \|\vec{u} - \vec{v}\|_2 \tag{3.10}$$

The mean of the Euclidean distance is taken for the vector embeddings from the discriminator $\vec{d}$ of each respective generator $\vec{d}_{g_1}, \vec{d}_{g_2}$. This is calculated across the mini-batch $\vec{d}_{bg_1}, \vec{d}_{bg_2}$ and the mean is taken to calculate the loss for the mini-batch. The total loss for both generators is given in Eqs. 3.11 & 3.12.

$$G_1 \; loss = \overline{\left\|\vec{d}_{bg_1} - \vec{d}_{bg_2}\right\|_2} + Var(B^c_{g_1}) - Var(B^c_{g_2}) \tag{3.11}$$

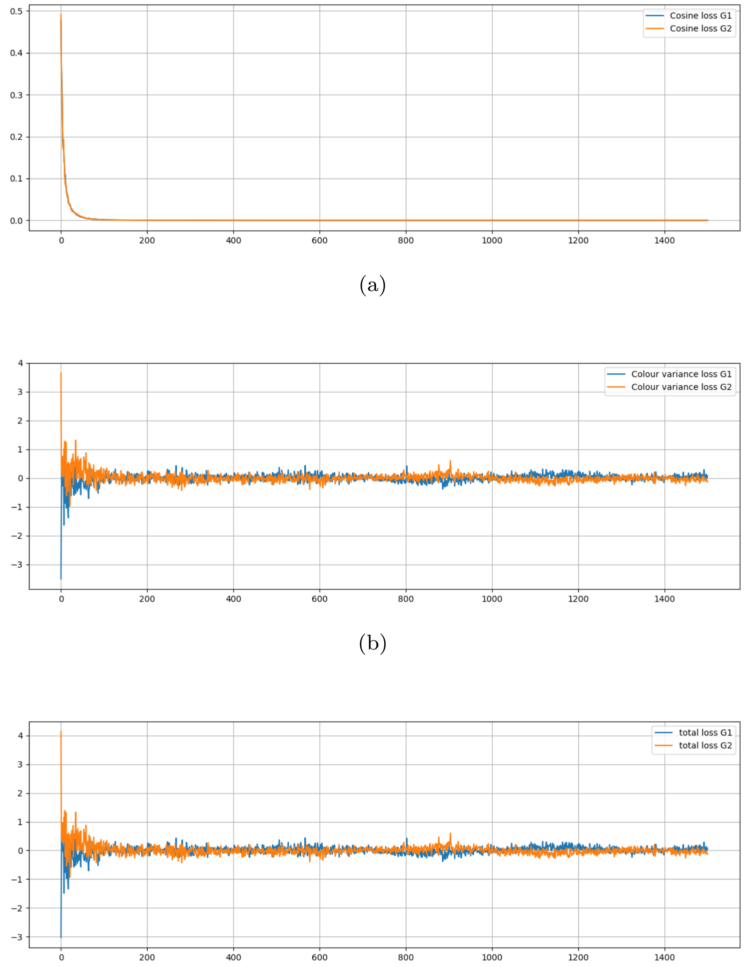
$$G_2 \; loss = \overline{\left\|\vec{d}_{bg_2} - \vec{d}_{bg_1}\right\|_2} + Var(B^c_{g_2}) - Var(B^c_{g_1}) \tag{3.12}$$

The samples from training in this experiment are given in Figure 3.9 and the loss plots are given in 3.10.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 3.7: Training samples for two generators with cosine distance with additional colour variance loss term, sampled at increments of 50 iterations. The left column shows the training samples for G_1 and the right column shows the training samples for G_2. (a,b) Training samples at resolution 32x32 for iterations 0-300. (c,d) Training samples at resolution 64x64 for iterations 300-600. (e,f) Training samples at a resolution of 128x128 for iterations 600-900. (g,h) Training samples at resolution 256x256 at iterations 900-1200. (i,j) Training samples at resolution 512x512 for iterations 1200-1500.

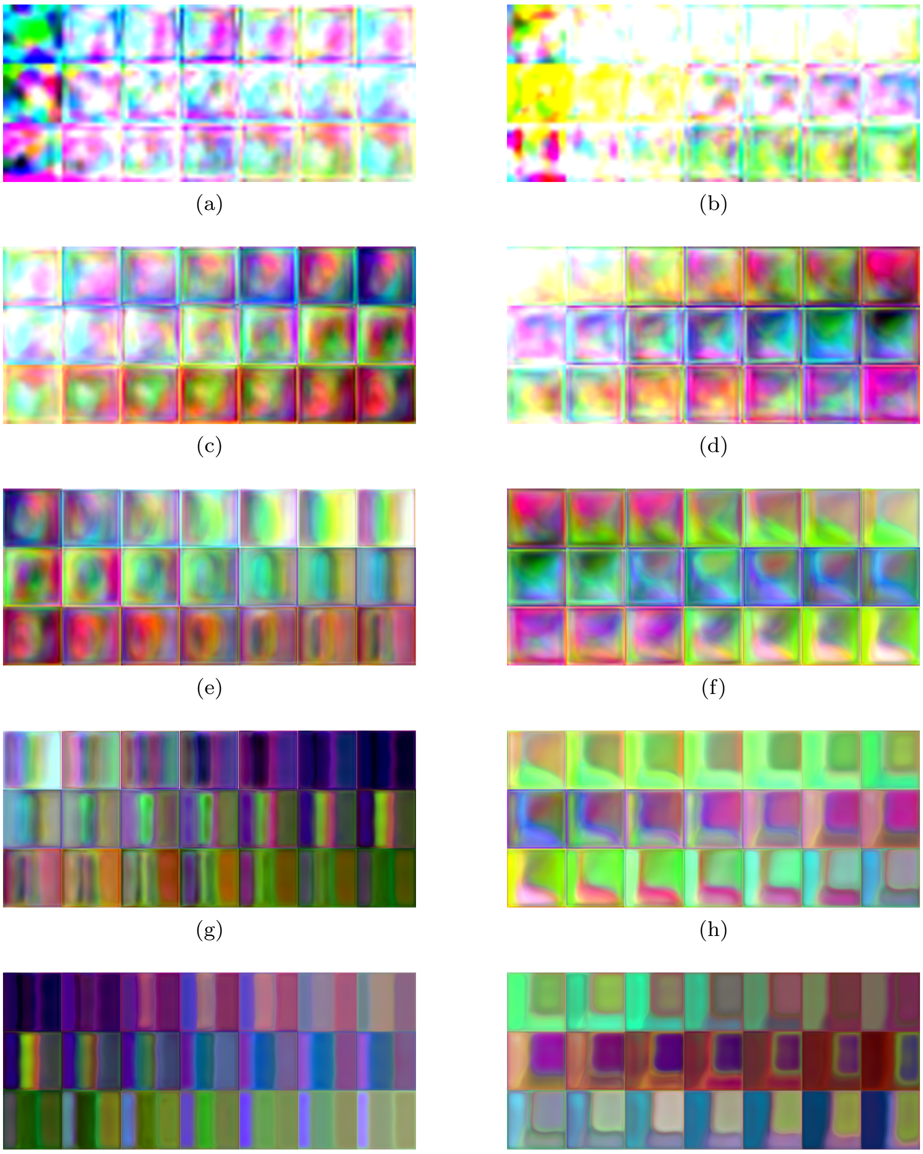


(a)

(b)

(c)

Figure 3.8: Loss plots for cosine distance training with colour variation loss term. (a) Cosine distance loss terms for both respective generators. (b) Colour variation loss term for both respective generators. (c) Total loss combining both terms for both respective generators.



(a)

(b)

(c)

(d)

(e)

(f)

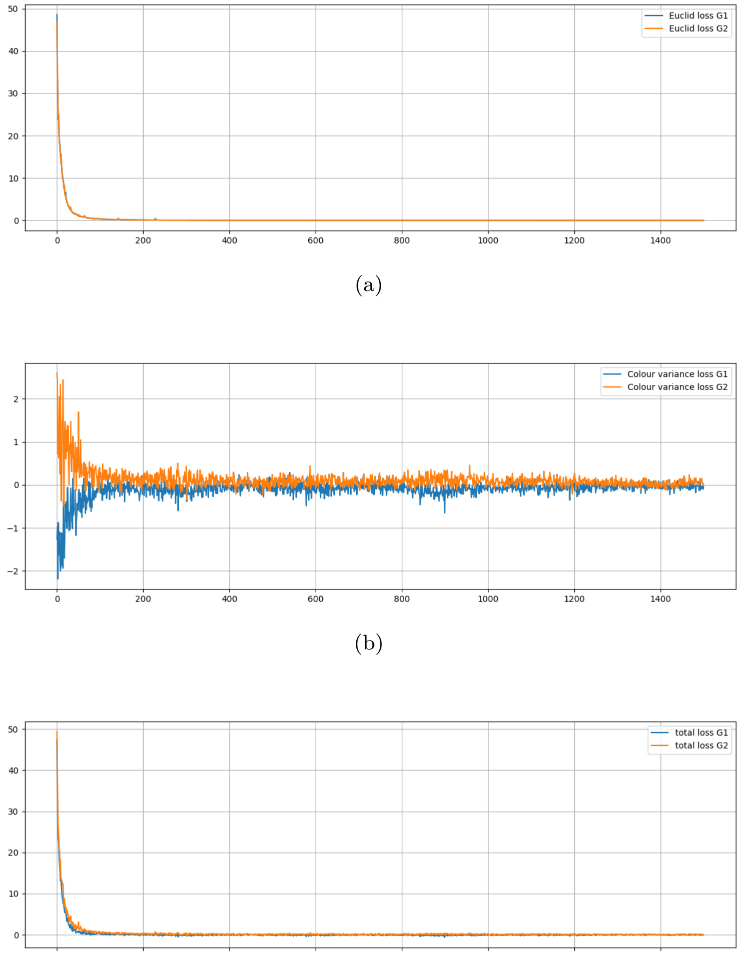
(g)

(h)

(i)

(j)

Figure 3.9: Training samples for two generators with Euclidean distance with additional colour variance loss term, sampled at increments of 50 iterations. The left column shows the training samples for G_1 and the right column shows the training samples for G_2. (a,b) Training samples at resolution 32x32 for iterations 0-300. (c,d) Training samples at resolution 64x64 for iterations 300-600. (e,f) Training samples at a resolution of 128x128 for iterations 600-900. (g,h) Training samples at resolution 256x256 at iterations 900-1200. (i,j) Training samples at resolution 512x512 for iterations 1200-1500.



(a)

(b)

(c)

Figure 3.10: Loss plots for Euclidean distance training with colour variation loss term. (a) Euclidean distance loss terms for both respective generators. (b) Colour variation loss term for both respective generators. (c) Total loss combining both terms for both respective generators.

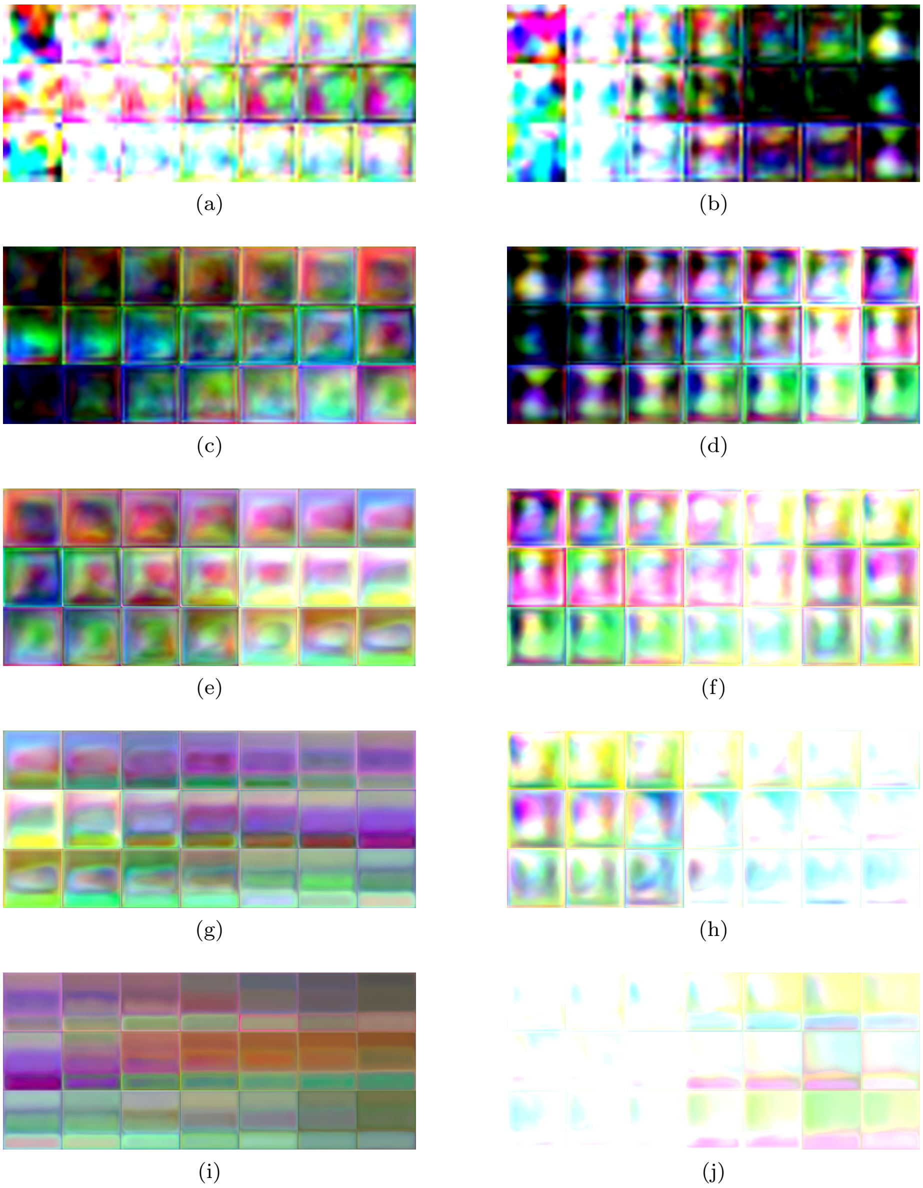
3.5 Mixing Generator Loss Functions

In the final three experiments the adversarial, cosine distance, and Euclidean distance are mixed for training the respective generators.

Figures 3.11 & 3.12 show the training samples and loss plots for mixing the cosine distance and adversarial loss for the two respective generators. For this training setup G_1 is trained with Equation 3.8 and G_2 is trained with Equation 3.6.

Figures 3.13 & 3.14 show the training samples and loss plots for mixing the Euclidean distance and adversarial loss for the two respective generators. For this training setup G_1 is trained with Equation 3.11 and G_2 is trained with Equation 3.6.

Figures 3.15 & 3.16 show the training samples and loss plots for mixing the Euclidean distance and adversarial loss for the two respective generators. For this training setup G_1 is trained with Equation 3.8 and G_2 is trained with Equation 3.12.


(a)

(b)

(c)

(d)

(e)

(f)

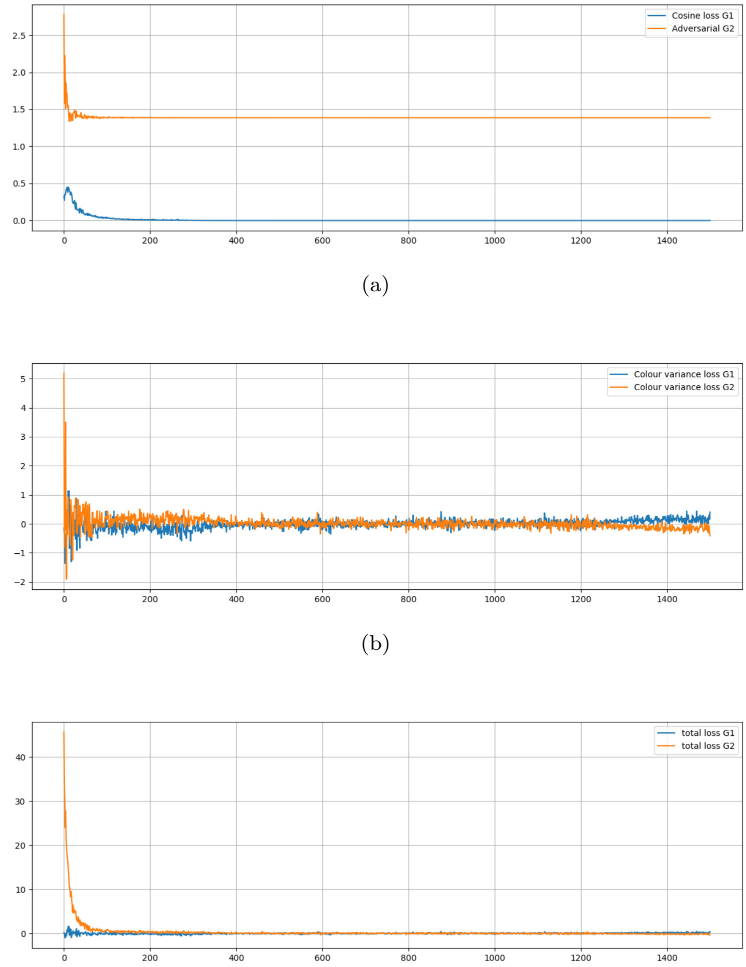
(g)

(h)

(i)

(j)

Figure 3.11: Training samples for two generators, one with cosine distance and one with adversarial loss, both with additional colour variance loss term, sampled at increments of 50 iterations. The left column shows the training samples for G_1 and the right column shows the training samples for G_2. (a,b) Training samples at resolution 32x32 for iterations 0-300. (c,d) Training samples at resolution 64x64 for iterations 300-600. (e,f) Training samples at a resolution of 128x128 for iterations 600-900. (g,h) Training samples at resolution 256x256 at iterations 900-1200. (i,j) Training samples at resolution 512x512 for iterations 1200-1500.

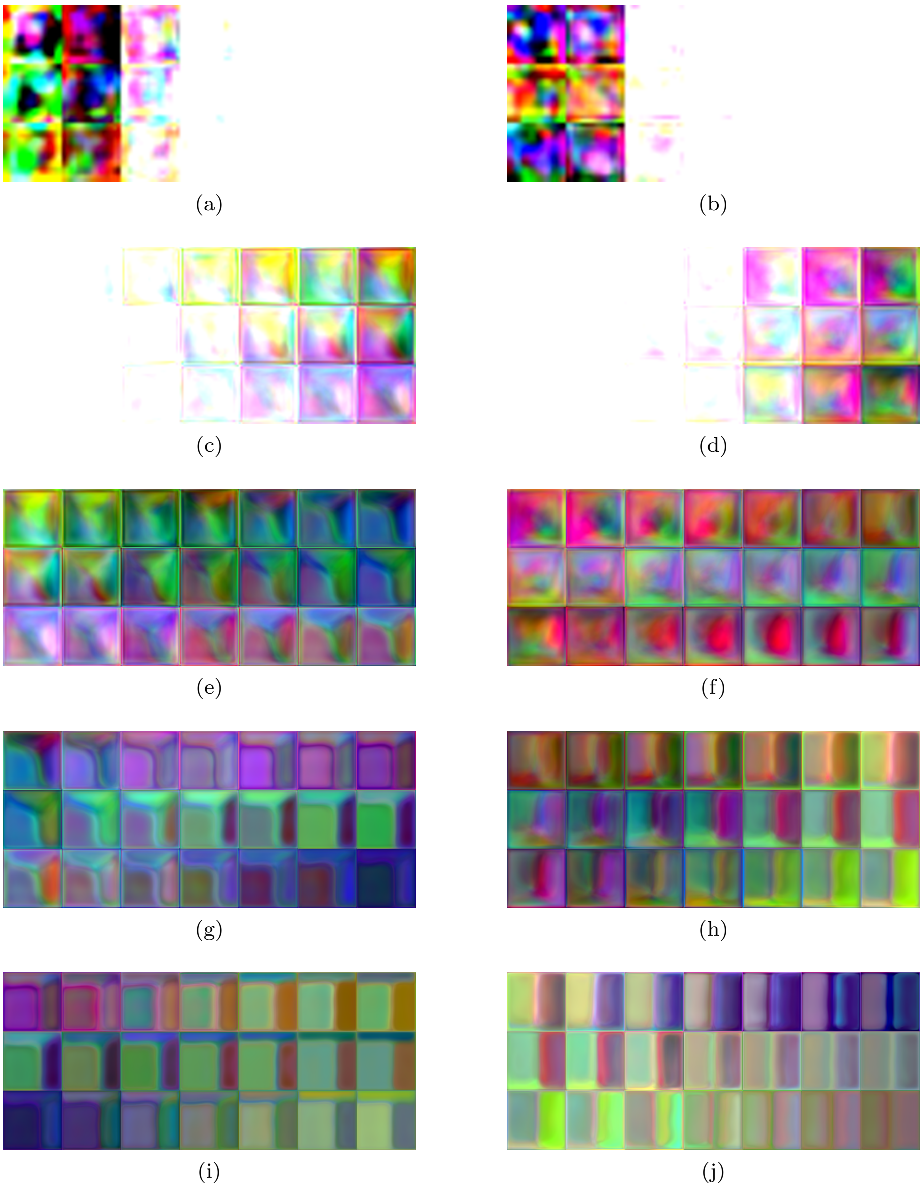


(a)

(b)

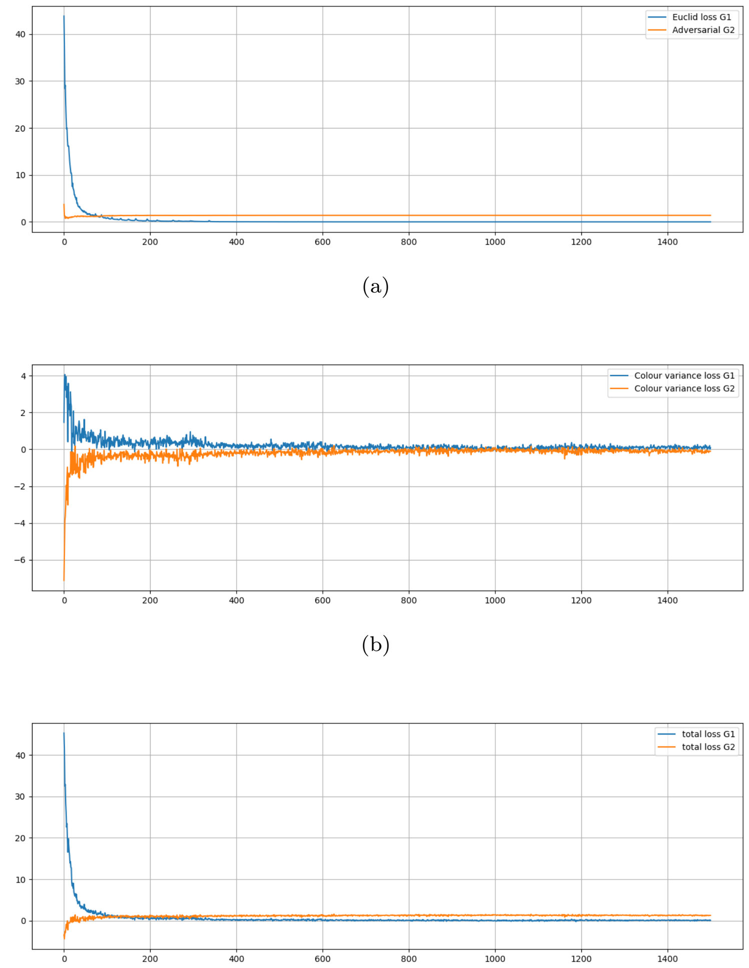
(c)

Figure 3.12: Loss plots for mixed generator losses with cosine distance and adversarial loss, both with colour variation loss term. (a) Losses for both respective generators, G_1 is trained with Cosine distance and G_2 is trained with the adversarial loss. (b) Colour variation loss term for both respective generators. (c) Total loss combining both terms for both respective generators.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 3.13: Training samples for two generators, one with Euclidean distance and one with adversarial loss, both with additional colour variance loss term, sampled at increments of 50 iterations. The left column shows the training samples for G_1 and the right column shows the training samples for G_2. (a,b) Training samples at resolution 32x32 for iterations 0-300. (c,d) Training samples at resolution 64x64 for iterations 300-600. (e,f) Training samples at a resolution of 128x128 for iterations 600-900. (g,h) Training samples at resolution 256x256 at iterations 900-1200. (i,j) Training samples at resolution 512x512 for iterations 1200-1500.

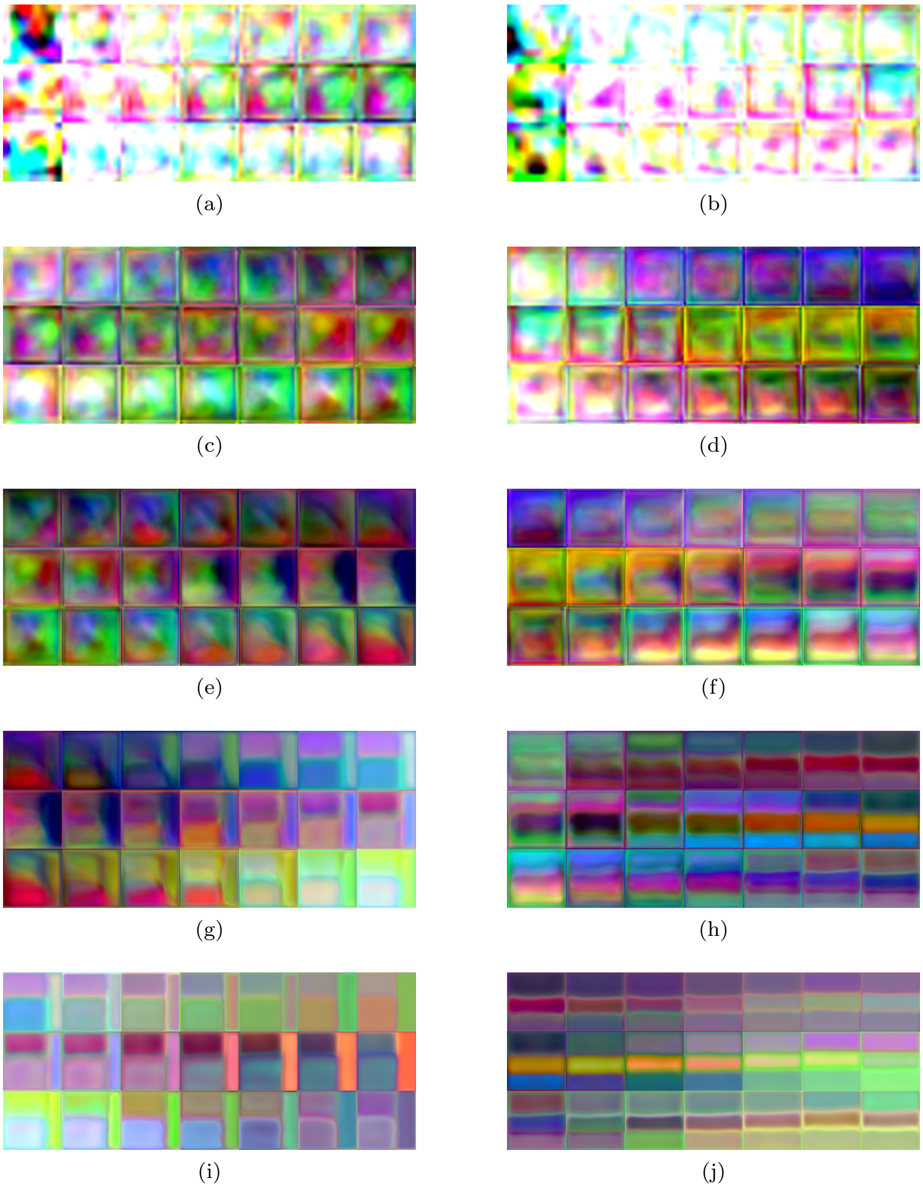


(a)

(b)

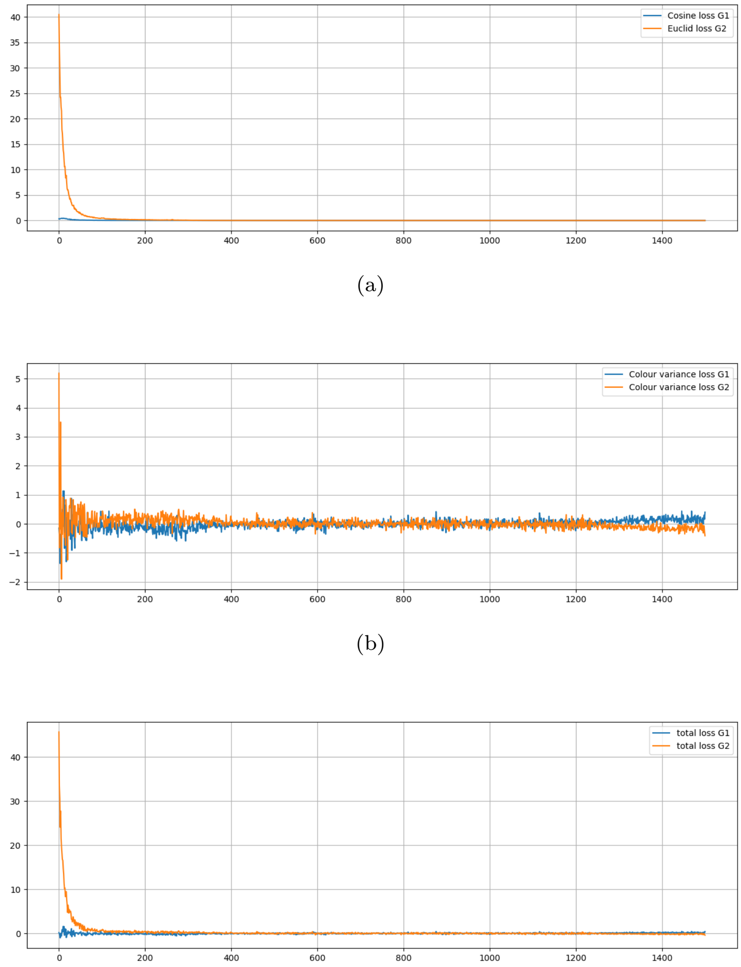
(c)

Figure 3.14: Loss plots for mixed generator losses with Euclidean distance and adversarial loss, both with colour variation loss term. (a) Losses for both respective generators, G_1 is trained with Euclidean distance and G_2 is trained with the adversarial loss. (b) Colour variation loss term for both respective generators. (c) Total loss combining both terms for both respective generators.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 3.15: Training samples for two generators, one with cosine distance and one with Euclidean distance, both with additional colour variance loss term, sampled at increments of 50 iterations. The left column shows the training samples for G_1 and the right column shows the training samples for G_2. (a,b) Training samples at resolution 32x32 for iterations 0-300. (c,d) Training samples at resolution 64x64 for iterations 300-600. (e,f) Training samples at a resolution of 128x128 for iterations 600-900. (g,h) Training samples at resolution 256x256 at iterations 900-1200. (i,j) Training samples at resolution 512x512 for iterations 1200-1500.



(a)

(b)

(c)

Figure 3.16: Loss plots for mixed generator losses with cosine distance and Euclidean distance, both with colour variation loss term. (a) Losses for both respective generators, G_1 is trained with cosine distance and G_2 is trained the Euclidean distance. (b) Colour variation loss term for both respective generators. (c) Total loss combining both terms for both respective generators.

3.6 Discussion

The results for all of the training experiments using the colour variance loss term demonstrate similar visual and aesthetic properties. It is my suspicion here that it is the colour variance loss term is the component for the loss that is most instrumental in defining the direction of optimisation and in the resulting visual results. While the experiments using distance metrics as alternatives to the adversarial loss may have a slight impact on the differences between the visual characteristics of each result, it is clear that this impact does not have a significant impact on the overall visual aesthetic. The difference in visual appearance could be accounted for by the randomness of the initial starting parameters, or the random sequence of sampling latent variables throughout the course of training, rather than the difference in loss terms for the generator networks. What was more important in my observations when training these models was the dynamics of the models over the course of training (albeit a very limited training run of 1500 iterations), and I settled on this arrangement because of the striking abstract visual qualities and diversity of colours across the whole generative space, which made them suitable for representing as artworks for the series of works *(un)stable equilibrium*.

Many people are surprised when I tell them that these works were made by training generative neural networks without data. I have had many encounters where people are in disbelief that this is possible. It is commonly assumed that I must have trained these models on a dataset of abstract paintings[1], or a synthetic dataset of colour gradients. The resemblance here to paintings by Mark Rothko is not lost on me in these experiments, and indeed that was the first thing that sprung to mind when I performed the first training run with the colour variance loss term.[2] This however, was a happy accident, or at least, my

[1] When presenting this work at the NeurIPS Workshop on Machine Learning for Creativity and Design, one woman was adamant that I must have used trained these networks on a dataset of painting from the Color field movement of abstract painting, and would not believe me when I told her I did not use any training data.

[2] I even labelled the folder ‘Rothko-esque’ on my computer after performing the initial training run with the colour variance loss term.

aesthetic preferences were the ‘meta-heuristic’ guiding me in my development of the code over the course of several weeks which finally led to creating a training ensemble of networks that created works that share this resemblance.

These models can be trained very quickly, especially when compared to the time required for training standard generative models such as GANs. The length of training used in all these experiments is 1500 iterations, and the visual formation begins to stabilise after 1000 iterations. This could be due to a convergence towards some kind of stable equilibrium, but could also be in part determined by the progressive growing training arrangement in the StyleGAN models. By the time the models are scaled up to the resolution 256x256, the overall shape of the generations starts to stabilise, and this could be related to the number of parameters being trained in addition to the diminishing gradients that reach the lower layers of the network, where the structural formation of the images takes place. My experiments in Chapter 5 go into more detail exploring the functions of different layers within the generators of GANs.

3.7 Conclusion

The work in this chapter was significant for a number of reasons. This was the first breakthrough in the aim of developing a data-divergent way of training generative neural networks such that they create something completely novel. The way this was achieved was by leaning on my significant experience of coding and training machine learning systems, which I had grown to be quite comfortable with; to the degree that I could playfully experiment with the code and build neural network ensembles where gradient-based learning was able to be performed successfully. In addition, the results from the original training experiments (Fig. 3.1), which became the series of artworks *(un)stable equilibrium*, have had a significant artistic reception, which has been more significant than the technical contribution. This artistic reception is detailed in Section 7.2.

The following chapter builds on this work, though instead of focusing on training models from scratch, My next experiments explored fine-tuning models that had already been trained in a data divergent fashion.

Chapter 4

Optimising Towards Unlikelihood: Data-Divergent Fine-Tuning of Generative Neural Networks

This chapter details experiments in the divergent fine-tuning of pre-trained models with the goal of diverging from the likelihood-driven data modelling approach, towards the generation of novel, unseen data distributions. The work in this chapter was first published in the paper *'Amplifying The Uncanny'* was published at the 8th Conference on Computation, Communication, Aesthetics & X (xCoAx) [Broad et al., 2020a]. These experiments were the first peer-reviewed and published attempt at divergent fine-tuning that does not rely on imitation-based learning. Other approaches to divergent fine-tuning are detailed in Section 6.3.4. It should be noted that the results presented here are not the exact results first shared in Broad et al. [2020a]. The experiments were re-run, so that more data could be logged, and more variations of parameter settings and loss functions could be compared.

4.1 Motivation

Following the work presented in Chapter 3, I was motivated to explore further the possibility of training generative neural networks without data. However, given the rather esoteric nature of the arrangements of neural networks and loss functions presented in the last chapter, I wanted to pursue an approach that was more deliberate, with experiments that could be repeated by others more easily.

Instead of training neural networks completely from random initialisation, I wanted to find new ways to fine-tune pre-trained models using novel loss functions and auxiliary networks. The reasoning for this was that it was clear, based on experiments such as *deepdream* [Mordvintsev et al., 2015] and Tom White's perception engines [White, 2018, 2019] show that CNN-based computer vision models had powerful representations baked into them and that these representations had the potential to be utilised in the context of fine-tuning models to reveal otherwise hidden aspects of machine vision. My intuition was that if pre-trained image recognition models could be used for generating individual images, then they could also be used for fine-tuning existing generative models, in a data-divergent fashion.

The computational resources required to train the then state-of-the-art models such as BigGAN [Brock et al., 2019] and StyleGAN [Karras et al., 2019] were prohibitive in the context of this research. However, transfer learning and fine-tuning were something that I could experiment with much more rapidly in divergent ways. This has become increasingly common amongst creative practitioners working with high-fidelity models [Berns and Colton, 2020]. Instead of training from scratch, fine-tuning pre-trained models to be fine-tuned in divergent ways was something that I could experiment much more rapidly.

4.2 Method

These experiments were performed using pre-trained checkpoints from the original StyleGAN, which had been trained on the FFHQ dataset [Karras et al., 2019]. Three different checkpoints were used: one at 256x256 resolution, one at 512x512 resolution, and one at 1024x1024 resolution. These different checkpoints were available because the original StyleGAN implemented the progressive growing training method in [Karras et al., 2017]. The checkpoints used were not the official StyleGAN models released by NVIDIA, but instead, a PyTorch reimplementation [Rosinality, 2019]. This alternative implementation was used because the checkpoints contained both the weights of the generator and the discriminator.[1]

The standard GAN training objective is:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[1 - \log D(G(z))] \tag{4.1}$$

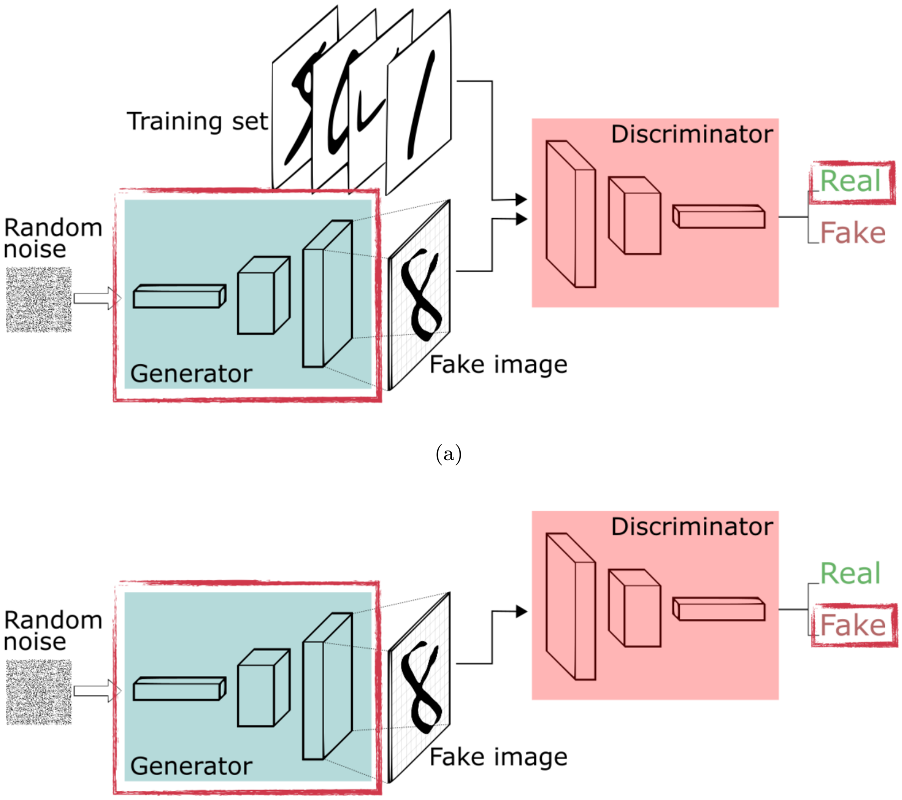
Where the generator network G is trying to minimise the likelihood of its generated samples being classified as fake, whereas the discriminator is optimised to maximise classification accuracy between real data samples and fake (generated) samples. The modification in this training regime is to invert this loss function following training, and then instead optimise the generator toward maximising the likelihood that its generated samples are correctly classified as being fake:

$$\max_G \mathbb{E}_{z \sim p_z(z)}[1 - \log D_f(G(z))] \tag{4.2}$$

In this procedure, there is no training objective for the discriminator. The weights of the discriminator D are kept frozen (D_f), and the network is simply

[1] The NVIDIA official releases of StyleGAN weights only have the checkpoints of the generator, not the discriminator.

used to calculate the loss function for the generator. Figure 4.1 shows visually the difference between the standard GAN training regime and this modified fine-tuning procedure.



(a)

(b)

Figure 4.1: Diagrams showing the standard GAN training regime with the loss that the generator is optimised towards (a), and the inverted adversarial loss fine-tuning procedure with the alternative loss for the generator network (b).

In a second set of experiments, this loss is modified to take the natural logarithm of the modified GAN loss:

$$\max_{G} \log(\mathbb{E}_{z\sim p_z(z)}[1 - \log D_f(G(z))]) \tag{4.3}$$

The results from this operation can be seen in Section 4.3.2.

For each model checkpoint (256, 512 & 1024) and loss function (inverted adversarial (Eq. 4.2) & log inverted adversarial (Eq. 4.3)), three fine-tuning runs were performed with different parameters for batch size during training: 2, 4 & 8. Experimenting with different conditions for batch size was undertaken in order to determine if this would have an impact on the generated results, as it is widely assumed that increasing the batch size improves visual fidelity in the standard GAN training regime. My intuition was that adjusting the batch size may also impact the diversity of generated outputs.

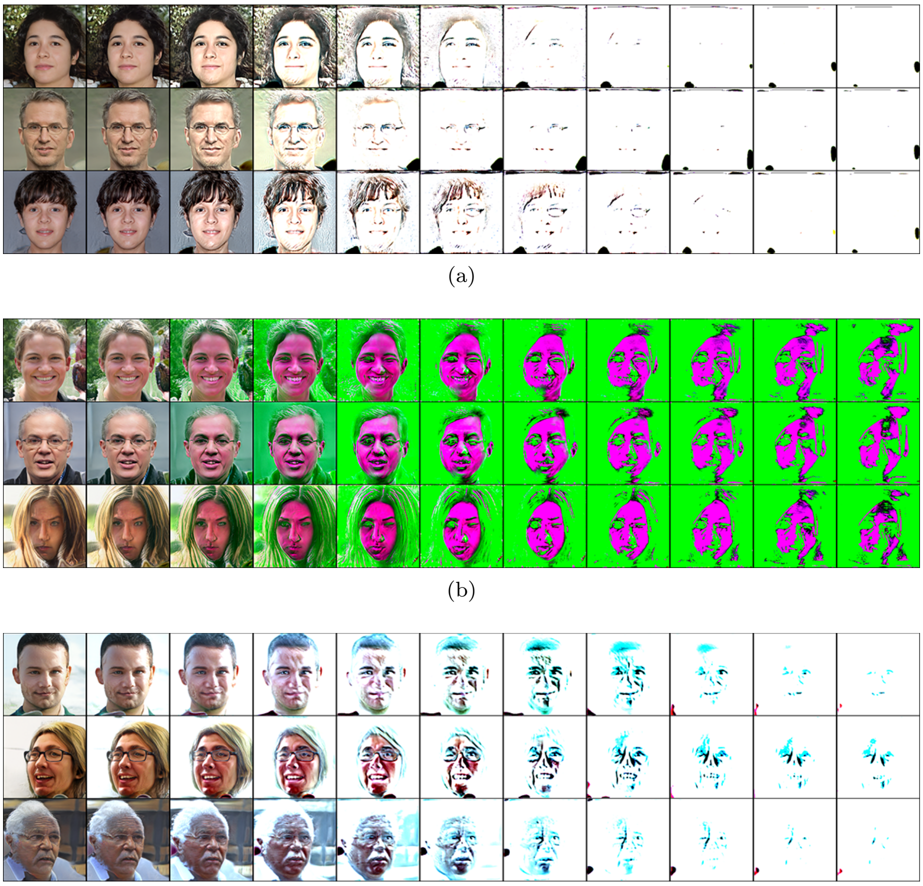
For the standard inverted adversarial loss function (Eq. 4.2), the fine-tuning procedure is run for 1000 iterations. For the natural logarithm of the inverted adversarial loss function (Eq. 4.3), fine-tuning is run for 10000 iterations. The reason for the difference in training duration is that the log-loss is less aggressive and therefore the fine-tuning procedure does not collapse as rapidly as the standard loss 4.2.

4.3 Results

This section shows the results of the fine-tuning training procedure. It is divided into two sub-sections. The first (§4.3.1) shows the results of the standard inverted adversarial loss (Eq. 4.2). The second (§4.3.2) shows the results for the natural logarithm of the inverted adversarial loss (Eq. 4.3).

4.3.1 Inverted Adversarial Loss

This section shows the results of the standard inverted adversarial loss (Eq. 4.2). For all of these experiments with this loss function the fine-tuning procedure was performed for 1000 iterations. Figures 4.2 & 4.3 show the generated samples and losses during fine-tuning for the 256x256 checkpoint. Figures 4.4 & 4.5 show the generated samples and losses during fine-tuning for the 512x512 checkpoint. Figures 4.6 & 4.7 show the generated samples and losses during fine-tuning for the 1024x1024 checkpoint.

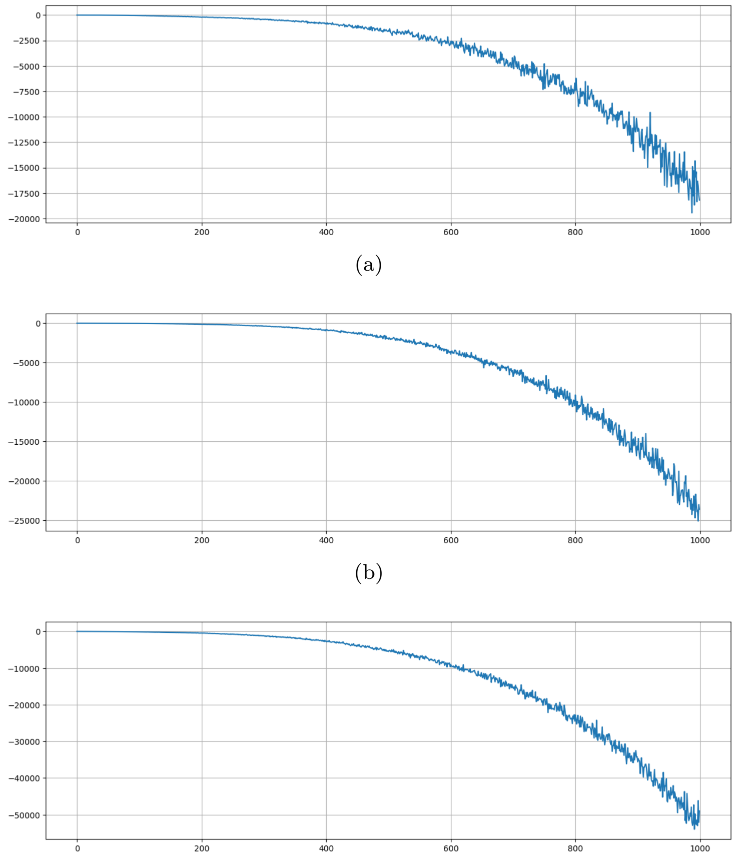


(a)

(b)

(c)

Figure 4.2: Samples of generations during the fine-tuning procedure for 256x256 StyleGAN model with the inverted adversarial loss function, at increments of 100, between training steps 0-10000. (a) Results with batch size 2. (b) Results with batch size 4. (c) Results with batch size 8.

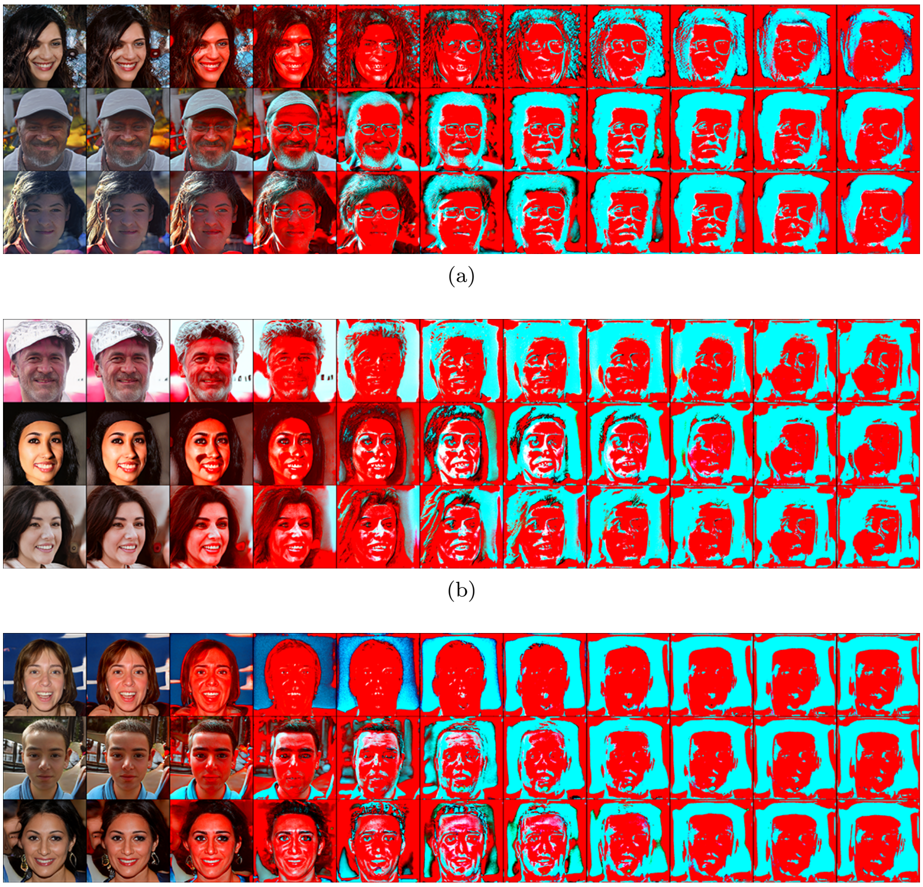


(a)

(b)

(c)

Figure 4.3: Loss plots for the fine-tuning procedure for 256x256 StyleGAN model with the inverted adversarial loss function. (a) Results with batch size 2. (b) Results with batch size 4. (c) Results with batch size 8.

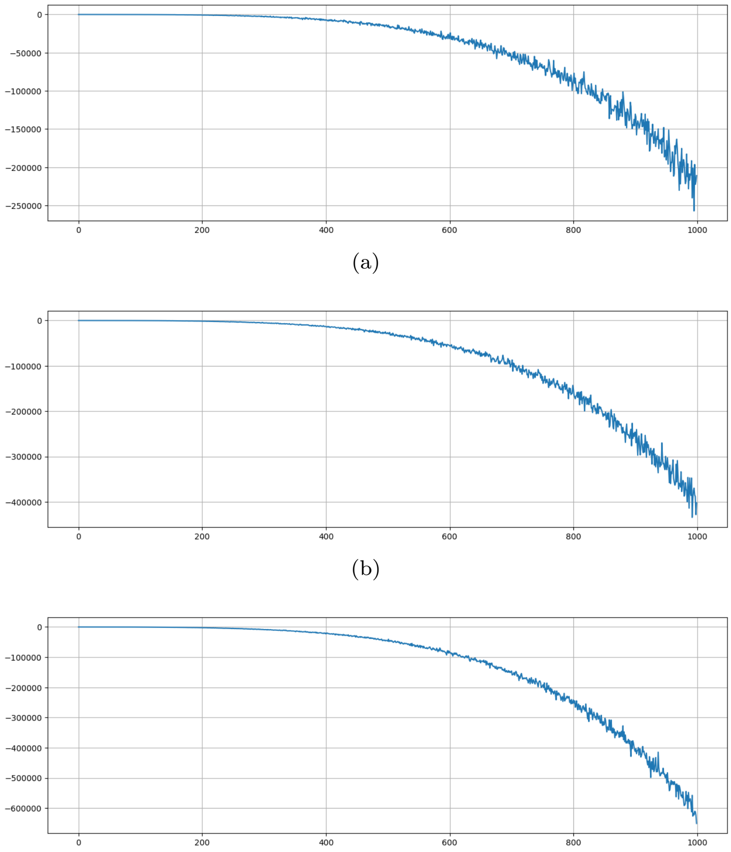


(a)

(b)

(c)

Figure 4.4: Samples of generations during the fine-tuning procedure for 512x512 StyleGAN model with the inverted adversarial loss function, at increments of 100, between training steps 0-1000. (a) Results with batch size 2. (b) Results with batch size 4. (c) Results with batch size 8.

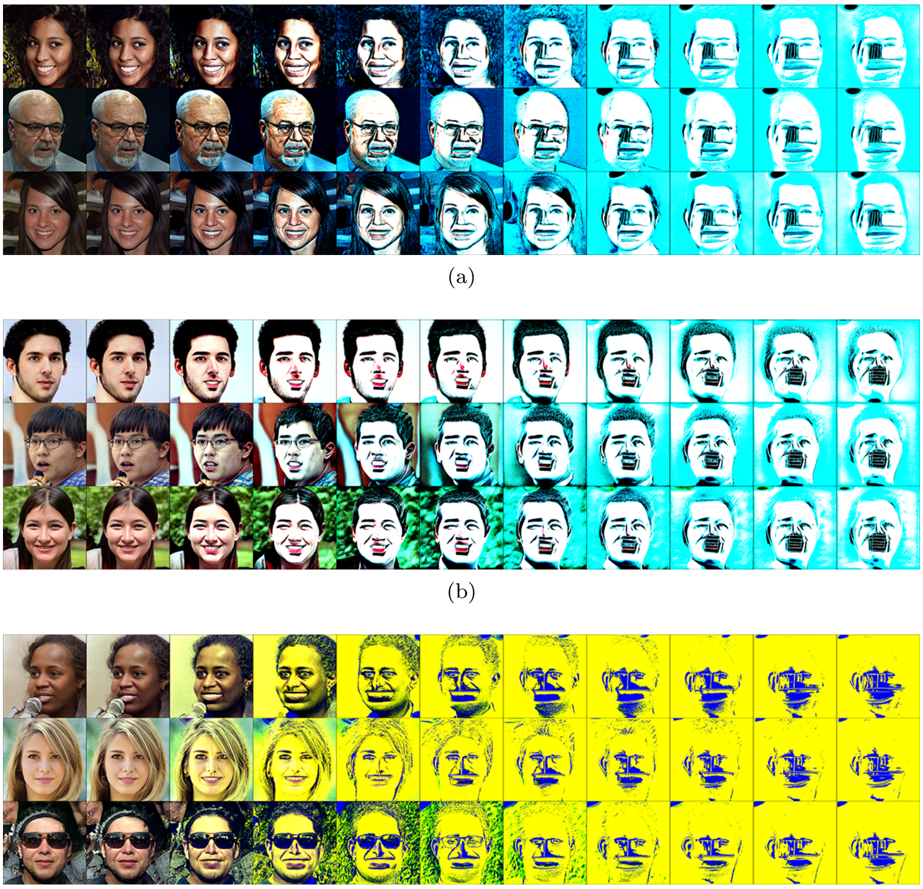


(a)

(b)

(c)

Figure 4.5: Loss plots for the fine-tuning procedure for 512x512 StyleGAN model with the inverted adversarial loss function. (a) Results with batch size 2. (b) Results with batch size 4. (c) Results with batch size 8.

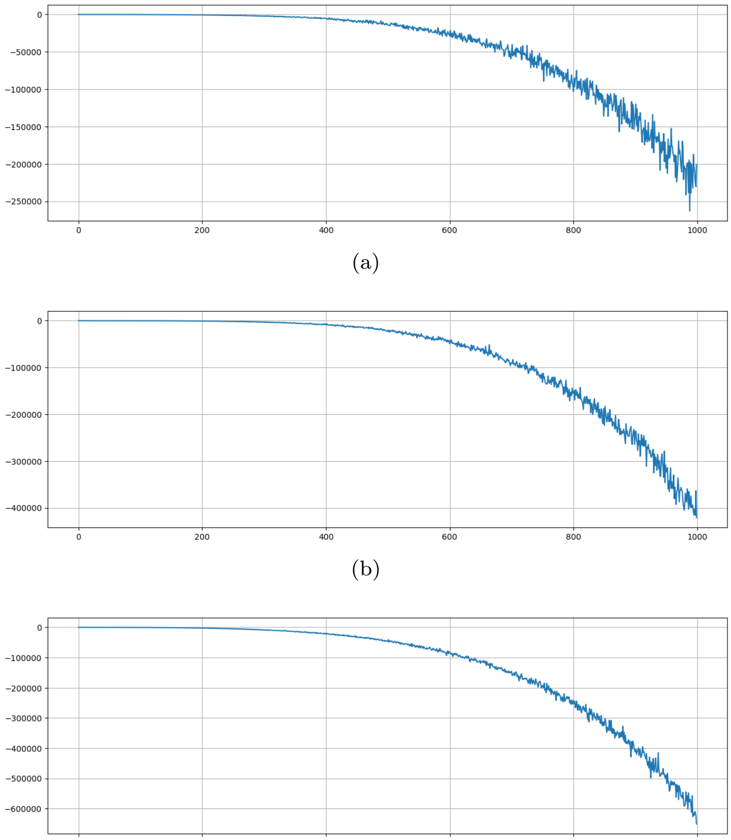


(a)

(b)

(c)

Figure 4.6: Samples of generations during the fine-tuning procedure for 1024x1024 StyleGAN model with the inverted adversarial loss function, at increments of 100, between training steps 0-1000. (a) Results with batch size 2. (b) Results with batch size 4. (c) Results with batch size 8.



(a)

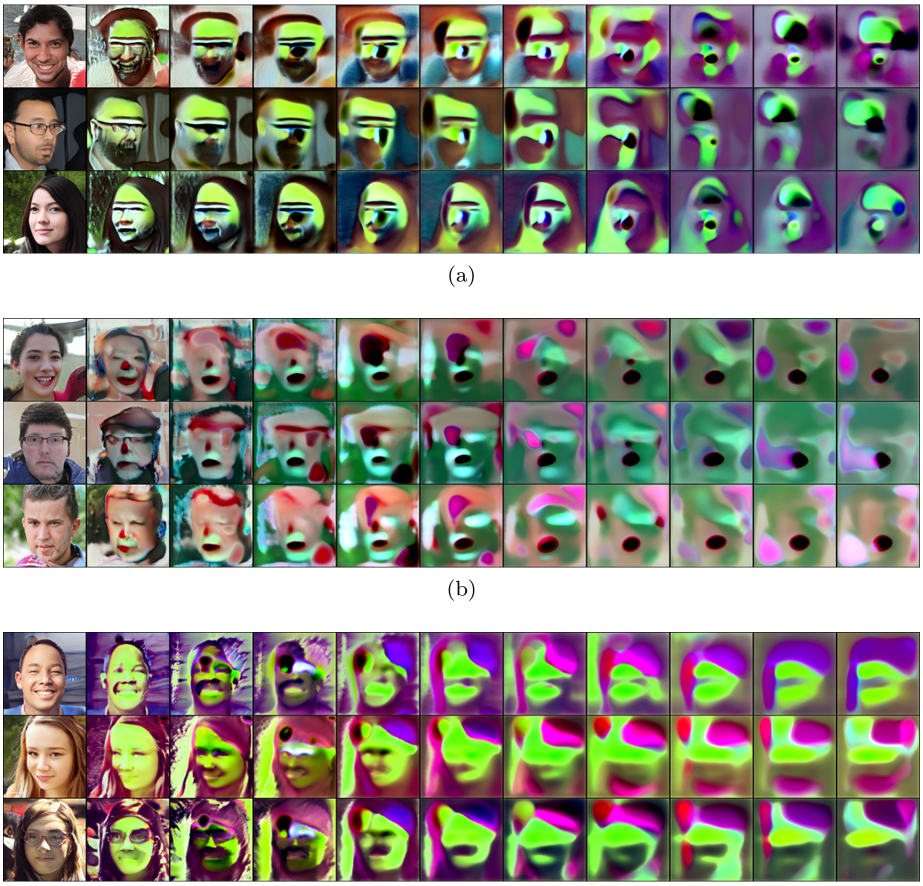
(b)

(c)

Figure 4.7: Loss plots for the fine-tuning procedure for 1024x1024 StyleGAN model with the inverted adversarial loss function. (a) Results with batch size 2. (b) Results with batch size 4. (c) Results with batch size 8.

4.3.2 Natural Logarithm of Inverted Adversarial Loss

This section shows the results of the standard inverted adversarial loss (Eq. 4.3). For all of these experiments with this loss function the fine-tuning procedure was performed for 10000 iterations. Figures 4.8 & 4.9 show the generated samples and losses during fine-tuning for the 256x256 checkpoint. Figures 4.10 & 4.11 show the generated samples and losses during fine-tuning for the 512x512 checkpoint. Figures 4.12 & 4.13 show the generated samples and losses during fine-tuning for the 1024x1024 checkpoint.

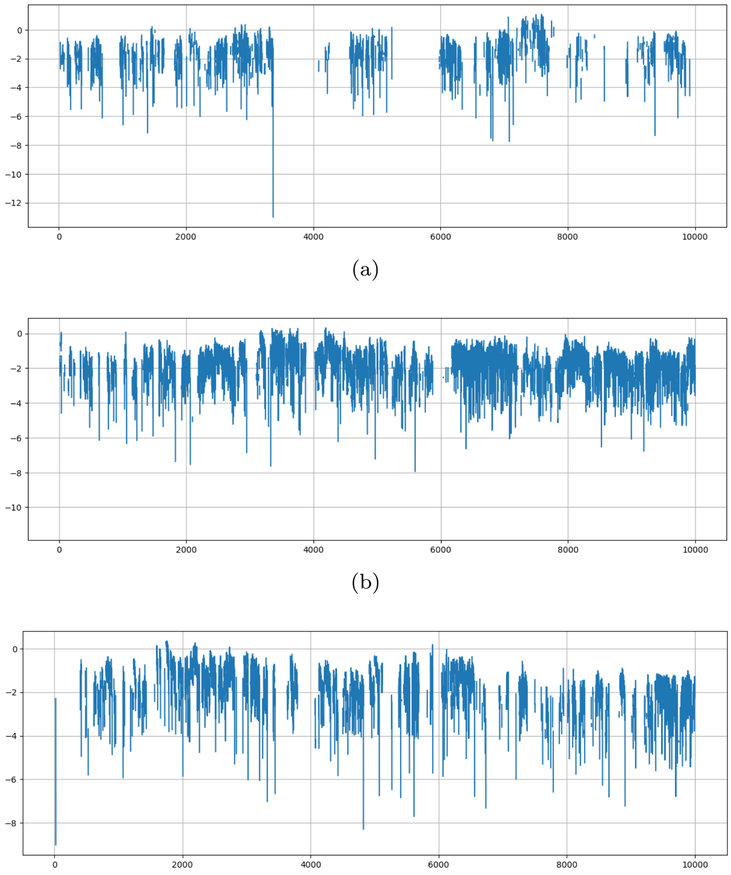


(a)

(b)

(c)

Figure 4.8: Samples of generations during the fine-tuning procedure for 256x256 StyleGAN model with the natural logarithm of the inverted adversarial loss function, at increments of 1000, between training steps 0-10000. (a) Results with batch size 2. (b) Results with batch size 4. (c) Results with batch size 8.

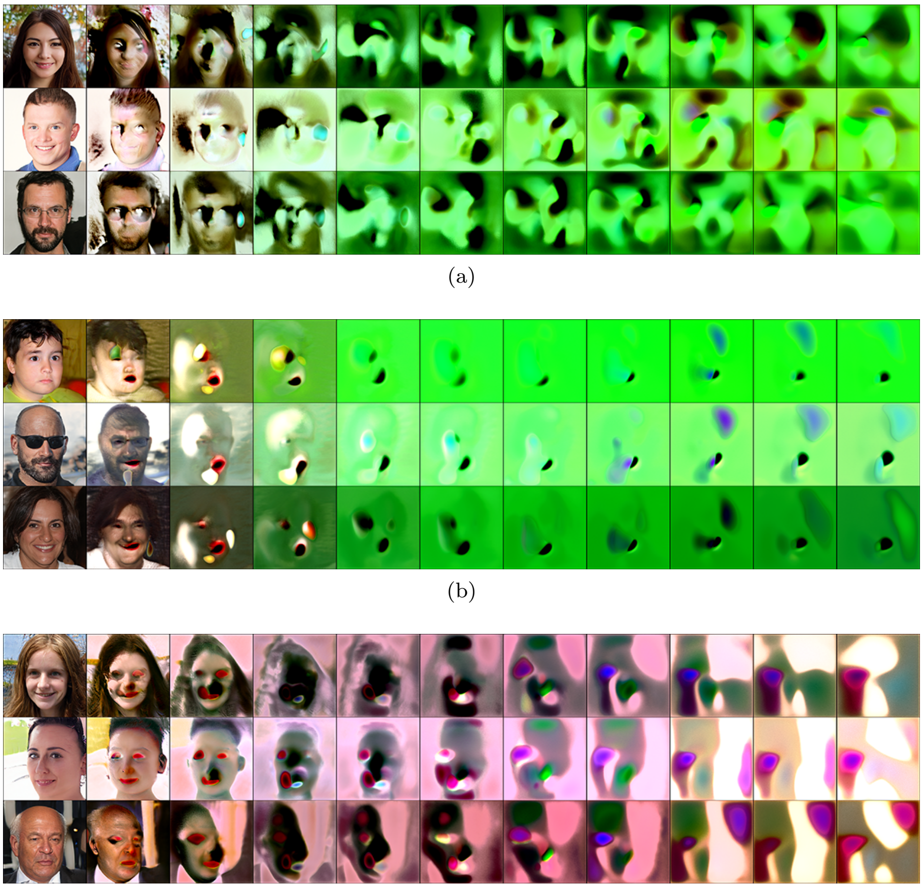


(a)

(b)

(c)

Figure 4.9: Loss plots for fine-tuning procedure for 256x256 StyleGAN model with the natural logarithm of the inverted adversarial loss function. (a) Results with batch size 2. (b) Results with batch size 4. (c) Results with batch size 8. Note: gaps in the plot are where the loss was undefined from taking the log of a negative number.



(a)

(b)

(c)

Figure 4.10: Samples of generations during the fine-tuning procedure for 512x512 StyleGAN model with the natural logarithm of the inverted adversarial loss function, at increments of 1000, between training steps 0-10000. (a) Results with batch size 2. (b) Results with batch size 4. (c) Results with batch size 8.

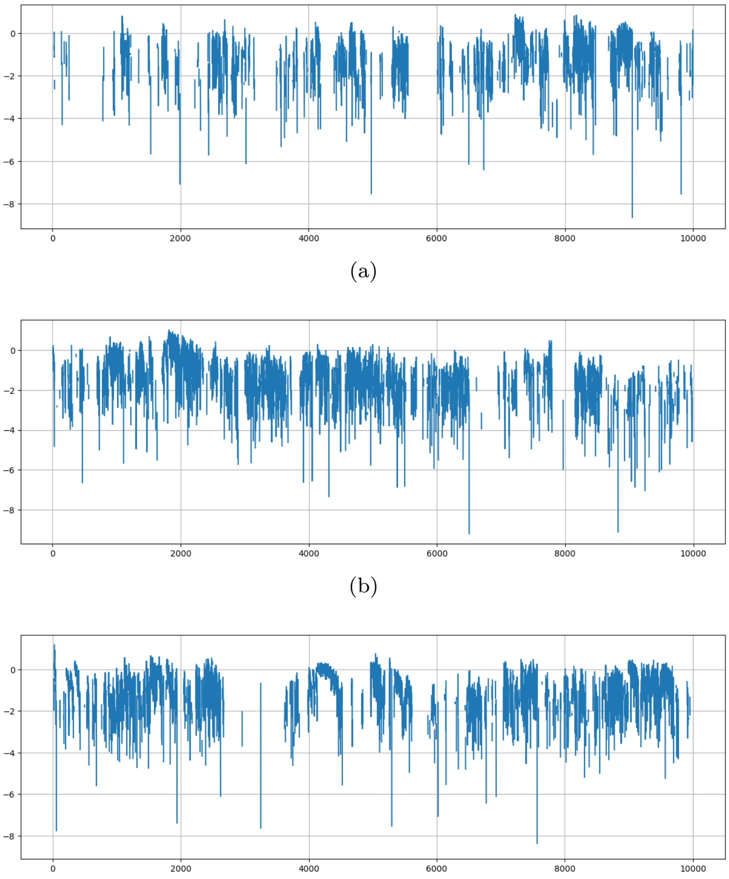
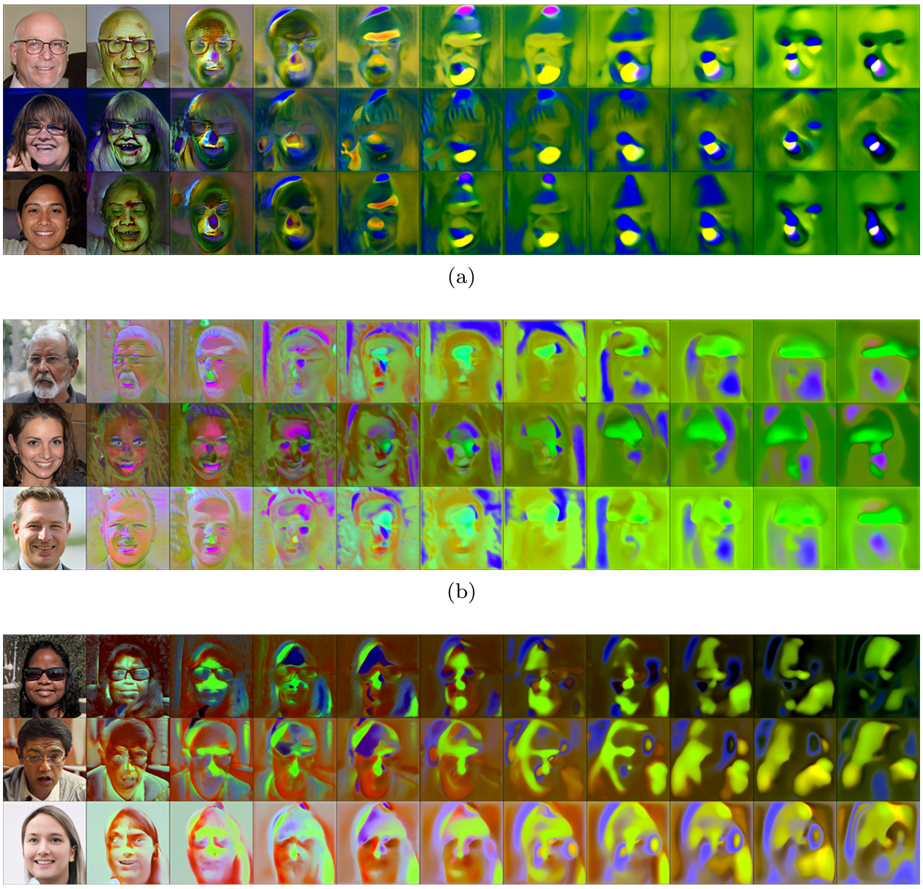


Figure 4.11: Loss plots for fine-tuning procedure for 512x512 StyleGAN model with the natural logarithm of the inverted adversarial loss function. (a) Results with batch size 2. (b) Results with batch size 4. (c) Results with batch size 8. Note: gaps in the plot are where the loss was undefined from taking the log of a negative number.

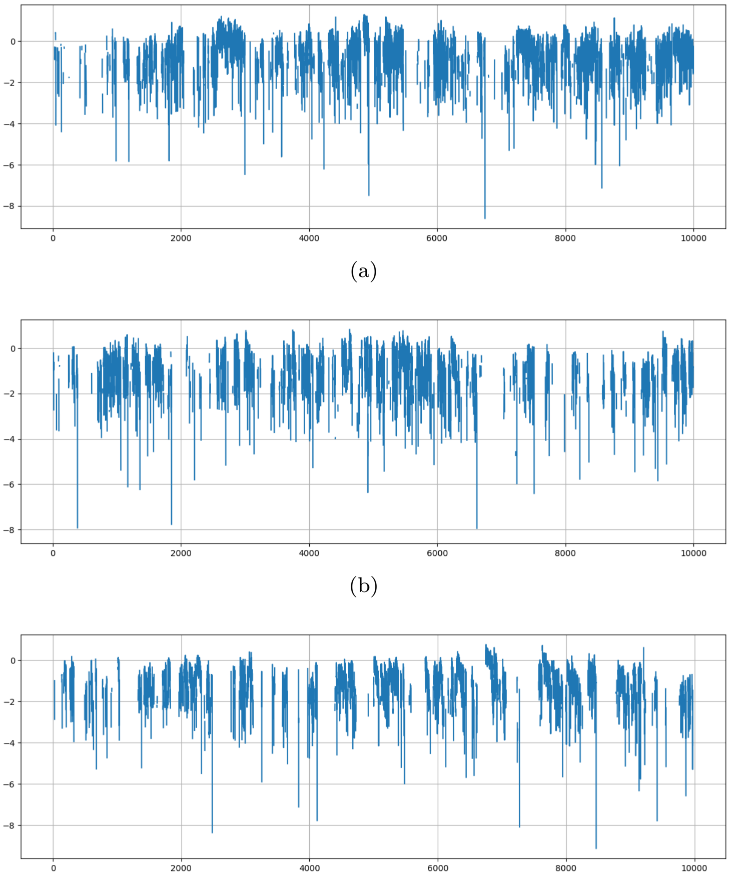


(a)

(b)

(c)

Figure 4.12: Samples of generations during the fine-tuning procedure for 1024x1024 StyleGAN model with the natural logarithm of the inverted adversarial loss function, at increments of 1000, between training steps 0-10000. (a) Results with batch size 2. (b) Results with batch size 4. (c) Results with batch size 8.



(a)

(b)

(c)

Figure 4.13: Loss plots for fine-tuning procedure for 1024x1024 StyleGAN model with the natural logarithm of the inverted adversarial loss function. (a) Results with batch size 2. (b) Results with batch size 4. (c) Results with batch size 8. Note: gaps in the plot are where the loss was undefined from taking the log of a negative number.

4.4 Discussion

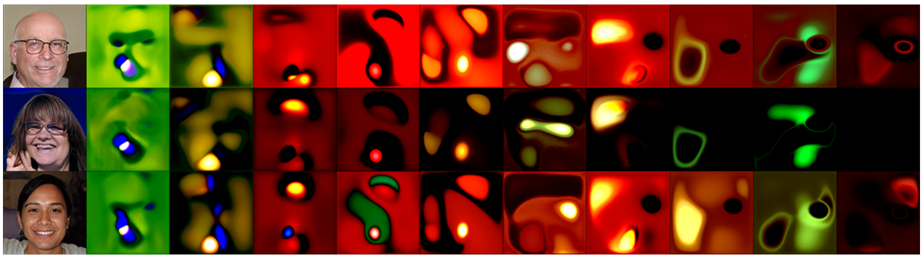
The results presented here demonstrate an efficient method for divergent fine-tuning that draws the generator away from the likelihood of the original training data. The results from the experiments with the standard inverted adversarial loss (Figs. 4.2, 4.4 & 4.6) show a method that quickly optimises towards a fixed representation of 'unlikelihood'. Of course, this representation of unlikelihood is contingent on the particular state of the two models (generator and discriminator) that is then fixed when the model checkpoints are saved during the training process. GANs, unlike many other training regimes for generative models, do not converge to a fixed point in the optimisation process. Instead, they act as a dynamic system, with no target end state. The optimisation problem in adversarial approaches is circular [Nagarajan and Kolter, 2017]. The generator and discriminator will forever be playing this game of forger/detective. The discriminator endlessly picks up on new minuscule flaws in the generator output, and the generator in turn responds in a potentially eternal adversarial game. Therefore, for each different saved checkpoint of the discriminator, different attributes that define unlikelihood are being optimised by the generator. It is clear in the visual results of the different models using the standard adversarial loss (Eq. 4.2; Figs. 4.2, 4.4 & 4.6) that there are specific aspects and features of the image that are being optimised towards.

Whilst the results for the standard loss show that there is a generally coherent fixed point of optimisation for each of the model snapshots, there are minor visual differences in each of the different training runs. Whilst it would be easy to attribute this solely to the change in the batch size parameter, I suspect that the sampling regime is also a contingent part of the visual differences between the training runs. Input latents for the generator are sampled randomly, and this random sampling and the sequence in which random latents are sampled may be just as important in determining the direction for optimisation and the final visual results, as much as the batch size used in each of these training runs.

For the second loss function 4.3, which uses the natural log of the inverse adversarial loss, there is clearly much less of a fixed point to which the optimiser is moving and in-fact the optimal state is mathematically undefined in this loss function (as the natural logarithm prevents the loss tending towards a negative number feedback loop, as is seen in the original inverted adversarial loss experiments). Whilst there are similarities in all of these final results (abstract shapes with generally consistent colours), there is clearly not one fixed point here that is being optimised for each of the model checkpoints. Taking the log of the loss appears to constrain the loss function better. The logs of the losses with the standard adversarial loss (Eq. 4.2) quickly explode into very high ranges (exceeding -6 x 10^5 in some cases). This is due to the feedback loop of reinforcing changes, where any change to the parameters of the generator further increases the loss. By taking the log of the loss this feedback loop is nullified. There are likely two things at play here. In log space the exponential feedback loop is far less aggressive. As these loss scores also tend towards negative numbers, when we take the log of this loss, these negative numbers become undefined (as taking the log of a negative number is always undefined). As can be seen in Figures 4.9, 4.11 & 4.13, there are large gaps where there is the loss in undefined.[2] Once again this may act as a constraint that prevents the feedback loop seen in Figures 4.3, 4.5 & 4.7 from taking hold.

The fine-tuning procedure using Equation 4.3 appears to produce a dynamic between models between models that may be endless. Figure 4.14 shows one of these training runs continuing to 100,000 iterations, where it appears this process is still evolving. Similarly to the work in Chapter 3 this is an endless and undefined goal, leading to a dynamic process that continually produces ever-evolving abstract outputs.

[2] Surprisingly, backpropagating undefined numbers does not seem to be an issue in PyTorch.



(a)

Figure 4.14: 100k iterations of 1024x1024 StyleGAN model with the natural logarithm of the inverted adversarial loss function with batch parameter 8, at increments of 10000, between training steps 0-100000.

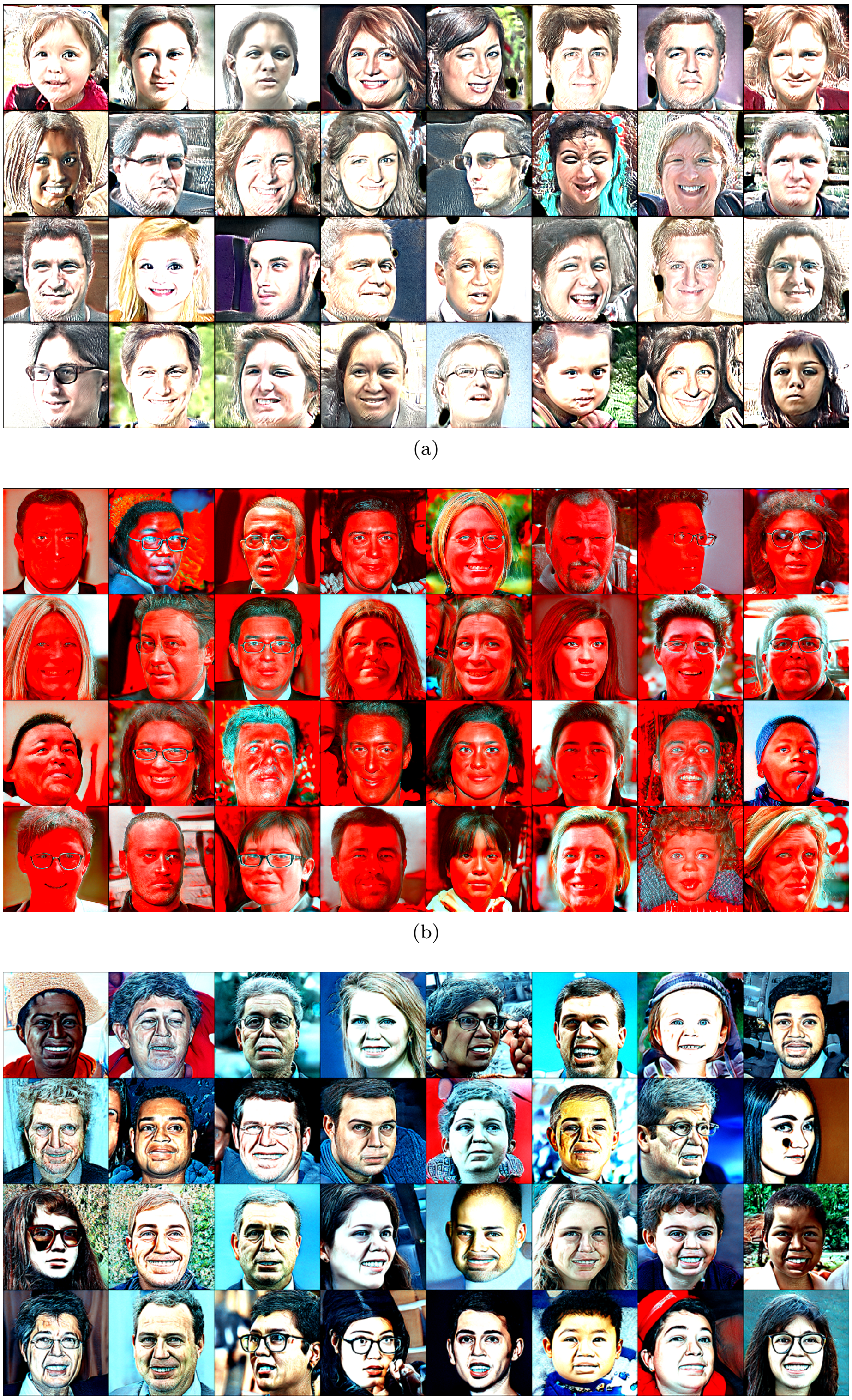
4.4.1 Relationship to The Uncanny

One of the observations that was made regarding the visual results of this process, especially in the transition stages of standard inverted loss, was the uncanny nature of the images.[3]

The uncanny is a psychological or aesthetic experience that can be characterised as observing something familiar that is encountered in an unsettling way. Jentsch defined the uncanny as an experience that stems from uncertainty, giving an example of it as being most pronounced when there is 'doubt as to whether an apparently living being is animate and, conversely, doubt as to whether a lifeless object may not in fact be animate' [Jentsch, 1906]. This definition was later refined to argue that the uncanny occurs when something familiar is alienated when the familiar is viewed in an unexpected or unfamiliar form [Freud, 1919].

The uncanny valley is a concept first introduced in 1970 by Masahiro Mori, a professor of robotics. It describes how in the field of robotics, an increase in the fidelity of human likeness increases feelings of familiarity up to a point (Fig. 4.17a), before suddenly decreasing. As representations of human or animal likeness approach a close resemblance to human or animal form, it provokes an unsettling feeling. Responses in likeness and familiarity rapidly become more

[3] Some of the early images I created during this process were quite disturbing (Fig. 7.4). The first person I showed them to was my partner at the time, whose response was to the effect of: 'Well that is horrifying, please never show me those pictures again'. I later showed the results to a PhD colleague of mine, Shringi, who had an equally negative reaction.

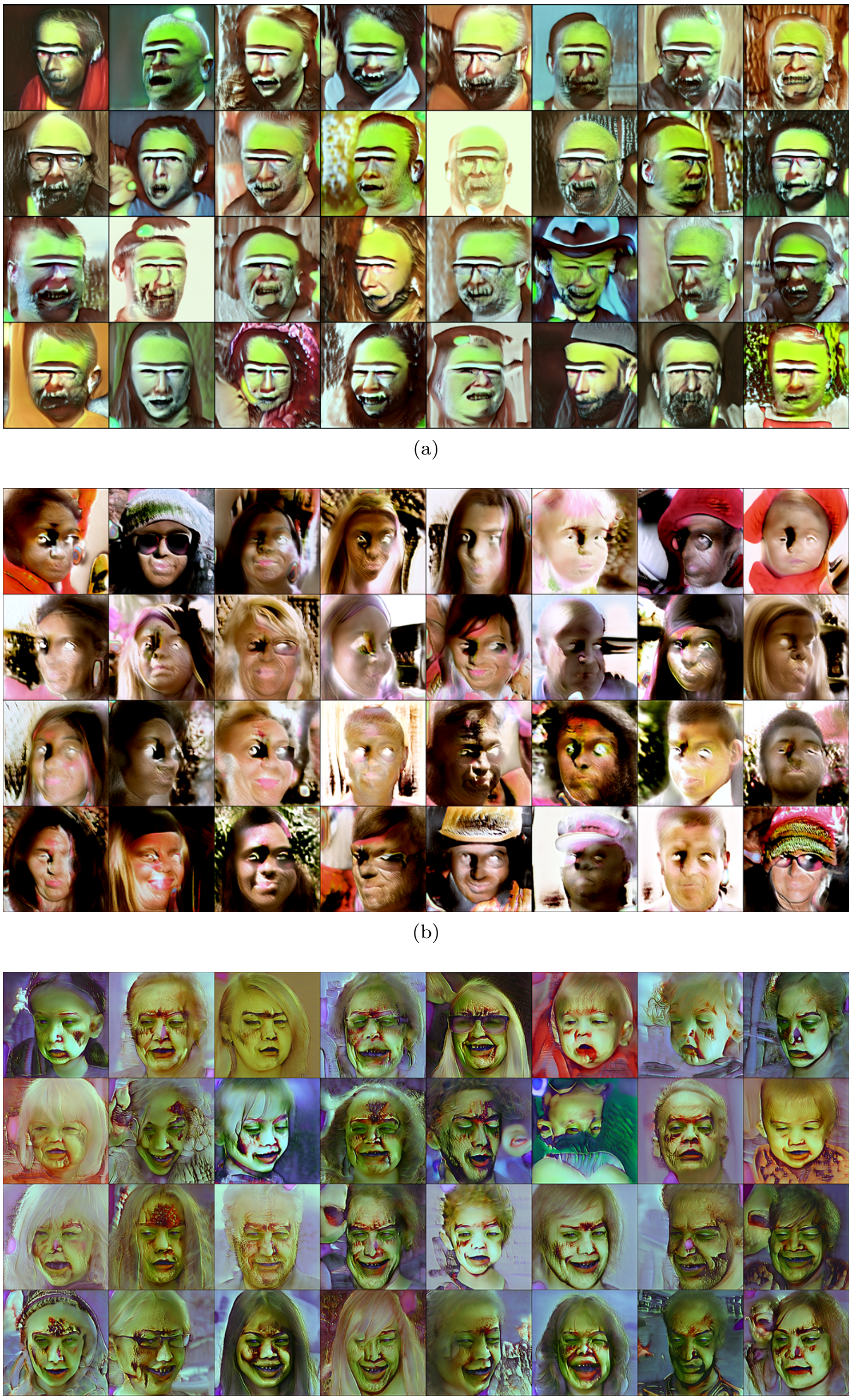


(a)

(b)

(c)

Figure 4.15: Uncanny images of samples from 300 iterations of fine-tuning with inverse loss, using a batch size of 2. (a) 256x256 model. (b) 512x512 model. (c) 1024x1024 model.



(a)

(b)

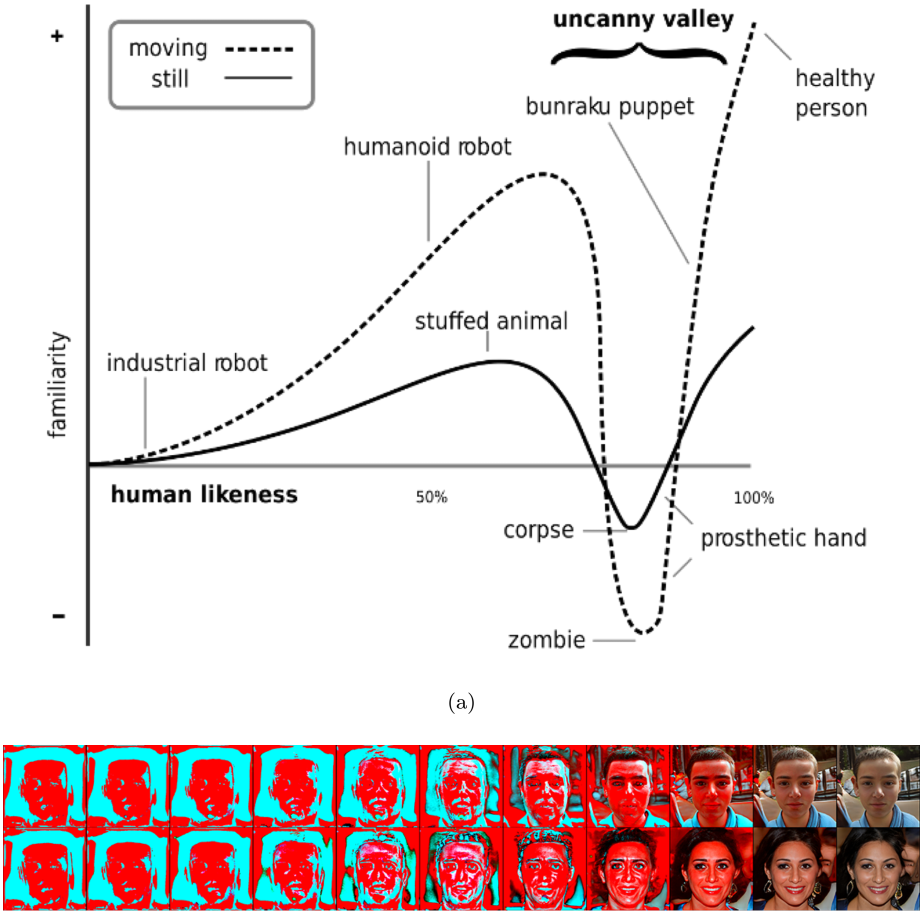
(c)

Figure 4.16: Uncanny images of samples from 1000 iterations of fine-tuning with natural logarithm of the inverse loss, using a batch size of 2. (a) 256x256 model. (b) 512x512 model. (c) 1024x1024 model.

negative than at any prior point. It is only when the robotic form is close to imperceptible with respect to human or animal likeness that the familiarity response becomes positive again [Mori, 1970]. As well as in robotics, this phenomenon has been observed in video games [Ratajczyk, 2019], visual effects [Schwind et al., 2018], and animation [Assaf et al., 2020].

In visual arts, the uncanny can be used deliberately to evoke unsettling feelings and explore boundaries between what is living and what is machine. This often reflects the anxieties and technologies of any given era, such as interactive robotic installations in the late 20th Century [Tronstad, 2008]. In work from the early 20th Century, such as Jacob Epstein's *Rock Drill* (circa 1913) which depicts the human form as transformed and amalgamated by industrial machinery [Grenville, 2001]. In the moving image, Czech animator Jan Svankmajer is well known for creating animated representations of the human form that deliberately confuse the viewer with respect to notions of life and lifelessness [Chryssouli, 2019].

The process of fine-tuning shown in Section 4.3 can be described as crossing the uncanny valley in reverse (Fig. 4.17). The original StyleGAN model trained on FFHQ was one of the first generative models to be able to generate images that were completely indistinguishable from real people to the untrained eye [Ajder et al., 2019], and a sophisticated understanding of the flaws of these models is needed in order to spot these deepfakes [McDonald, 2018]. Given the fact that images from these models have been used to make fake social media accounts [Satter, 2019] by spies trying to penetrate the American defence establishment, it is clear that StyleGAN-generated images, at least in some instances, have crossed the threshold of the uncanny valley towards producing completely plausible and convincing images of people. Starting from realism, and training towards abstraction, the process crosses the uncanny valley in reverse. As the generator starts to diverge from realism the images quickly become increasingly unsettling, before starting to plateau back to abstraction and returning to a more favourable likeness.



(a)

(b)

Figure 4.17: Uncanny valley diagram juxtaposed with fine-tuning samples in reverse order. (a) Diagram showing the uncanny valley [Mori, 1970]. (b) Samples from the fine-tuning procedure in Figure 4.4 (512x512 finetuned with loss 4.2 with batch parameter 8) in reverse order. Diagram (a) reproduced under a CC BY-SA 3.0 licence.

4.5 Conclusion

In this chapter, I have demonstrated an approach to fine-tuning pre-trained generative neural networks in a data-divergent fashion. This approach was the first peer-reviewed and published method for divergent finetuning that does not rely on imitation-based learning (to the best of my knowledge). A complete account of other known methods for divergent finetuning to date is given in Section 6.3.4.

While this work is novel, the results from all of the training runs described here are very idiosyncratic. The results are contingent on the unique state that the auxiliary models are in when their parameters are saved into checkpoints during training. In the case of the discriminator, this is completely unpredictable and not repeatable. While this can make for surprising outcomes, it also means that the experiments described would be impossible to reproduce without the exact model checkpoints and the random seed that is used for sampling the latent codes used for sampling the models during fine-tuning.

How would it be possible then to manipulate a generative network in a way that was more controllable and repeatable? This became a question that was playing on my mind after doing these experiments. The techniques described here use gradient descent to manipulate the weights of the model to produce novel outcomes. The process of gradient descent, however, is not something that we as humans can clearly understand, or easily control. I became preoccupied with finding a way of manipulating generative models, without relying on gradient descent. The next chapter is the third and final chapter that details a novel technical contribution of this thesis, one that centres humans in the creative process and allows them to manipulate generative neural networks without training or fine-tuning whatsoever.

Chapter 5

Network Bending: Direct and Expressive Manipulation of Generative Neural Networks

5.1 Introduction

This chapter details the development of the *network bending* framework. Which is a way to directly manipulate the internal features of generative neural networks during inference (Fig. 5.1 gives an overview of this process). This work was first published online as a pre-print on arxiv [Broad et al., 2020b] along with the source code on github[1] then later a revised manuscript was accepted as a conference paper at EvoMUSART [Broad et al., 2021b], and then as an extended journal paper in Entropy [Broad et al., 2022]. The experiments in the original paper were applied to the task of image generation with StyleGAN2 on

[1] The source for the original StyleGAN2 network bending experiments is available here: `https://github.com/terrybroad/network-bending`

models that were pre-trained on the Flickr-Faces High-Quality (FFHQ) [Karras et al., 2020] and Large-scale Scene UNderstanding (LSUN) churches dataset [Yu et al., 2015]. A follow-up study applying the same techniques to audio using a custom variational autoencoder (VAE) [Kingma and Welling, 2013, Rezende and Mohamed, 2015] trained on a dataset of varied musical genres (§5.6) was later completed for the extended Entropy paper. In this chapter, these experiments are documented chronologically.

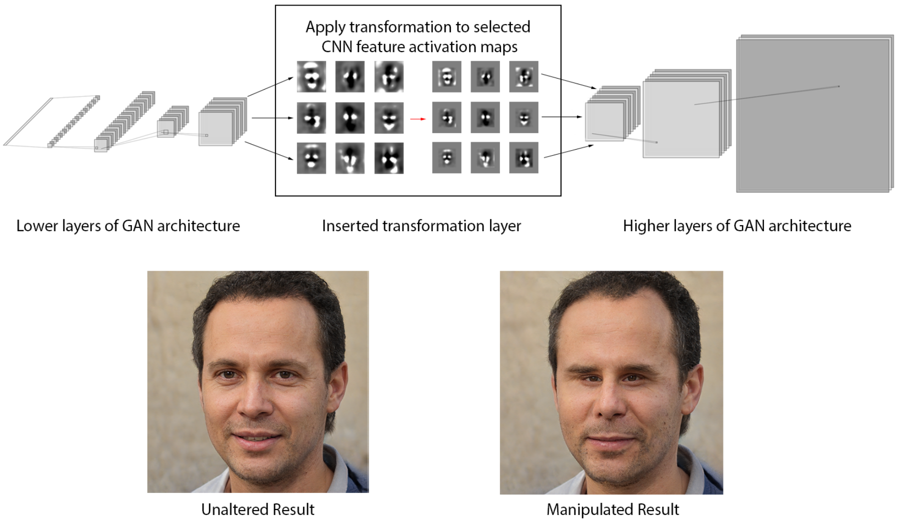

Apply transformation to selected CNN feature activation maps
Lower layers of GAN architecture
Inserted transformation layer
Higher layers of GAN architecture
Unaltered Result
Manipulated Result

Figure 5.1: Visual overview of the *network bending* framework, where deterministically controlled transformation layers can be inserted into a pre-trained network. As an example, a transformation layer that scales the activation maps by a factor of $k_x = k_y = 0.6$ is applied (§5.3.2) to a set of features in layer 5 responsible for the generation of eyes, which has been discovered in an unsupervised fashion using the clustering algorithm to cluster features based on the spatial similarity of their activation maps (§5.4.3). Bottom left shows the sample generated by StyleGAN2 [Karras et al., 2020] trained on the FFHQ dataset without modification, while the image on the right shows the same sample generated with the scaling transform applied to the selected features. NB: the GAN network architecture diagram shown on the top row is for illustrative purposes only.

5.2 Motivation

Following the experiments detailed in Chapters 3 & 4, I wanted to find an approach for actively diverging from data with generative neural networks, that was easier to control than methods that required the direct training or fine-tuning of the network itself. In addition, whilst producing novel outputs, the previous approaches did not necessarily expand the possibility space of what could be generated in a way that was arguably superior to traditional generative modelling. Both previous approaches focus on learning a set of weights that lead to reduced diversity in the generated outputs when compared with the successful training of a standard generative model such as a GAN or VAE. The goal of the work described in this chapter was to find an approach that would expand the generative space, not shrink it.

The inspiration for *network bending* came from a conversation with my supervisor Mick Grierson. After showing him the results detailed in the previous chapters, he said that though he liked the results, he was interested in methods that were more interactive and controllable, saying something along the lines of 'I just want to stick my hand in the model and squeeze it, and see what pops out the other side' [Grierson, 2019][2]. This statement stuck with me and eventually led to the development of the framework described here.

In some of the early experiments that led to this work, I hard-coded simple transformations into StyleGAN1 [Karras et al., 2019] models during inference. These early experiments (which later went on to become the series of artworks *Teratome* §7.4.1) sparked the intuition that eventually led to the implementation of many kinds of transformation layers (§5.3) and the clustering approach for grouping features together (§5.4.3). The motivation for developing the clustering algorithm was the observation that when transformations were applied to random subsets of convolutional filters in a layer, then in some instances, manipulation of groups of filters had apparently powerful semantic effects, that could not be captured by only manipulating individual filters, as was done in the approach presented by Bau et al. [2019].

[2] According to Mick Grierson, the idea for network bending was also being discussed in MIMIC (Musically Intelligent Machines Interacting Creatively) research team meetings around the same time. I was not in those meetings so I cannot give an exactly chronology. However, many people clearly had similar ideas around this time as the idea of applying transformations to the activation maps of GANs was developed independently and concurrently developed by two others [Pinkney, 2020b, Pouliot, 2020].

In creating this framework, I wanted to give as much control and agency to people to manipulate generative neural networks as possible. The flexibility of this framework was key in order to achieve this, allowing for the expansion of the generative space of generative neural networks in a data-divergent fashion.

5.3 Transformation Layers

A key goal in this framework was to give as much direct control and agency to artists and creative practitioners as possible. To maximise the amount of control people could have, I implemented a broad variety of deterministically controlled transformation layers that can be dynamically inserted into the computational graph of the generative model. The transformation layers are implemented natively in PyTorch [Paszke et al., 2019] for speed and efficiency. I treated the activation maps of each feature of the generative model as 1-channel images in the range -1 to 1. Each transformation is applied to the activation maps individually before they are passed to the next layer of the network.

The transformation layers can be applied to all the features in a layer, or a random selection, or by using pre-defined groups automatically determined based on spatial similarity of the activation maps (§5.4.3). Figure 5.2 shows a comparison of a selection of these transformations applied to all the features layer-wide in various layers of StyleGAN2.

5.3.1 Pointwise Transformations

I began with simple pointwise numerical transformations $f(x)$ that are applied to individual activation units x. I implemented four distinct numerical transformations: the first is *ablation*, which can be interpreted as $f(x) = x \cdot 0$. The second is *inversion*, which is implemented as $f(x) = 1 - x$. The third is *multiplication by a scalar p* implemented as $f(x) = x \cdot p$. The final transformation is *binary thresholding* (often referred to as posterisation) with threshold t, such

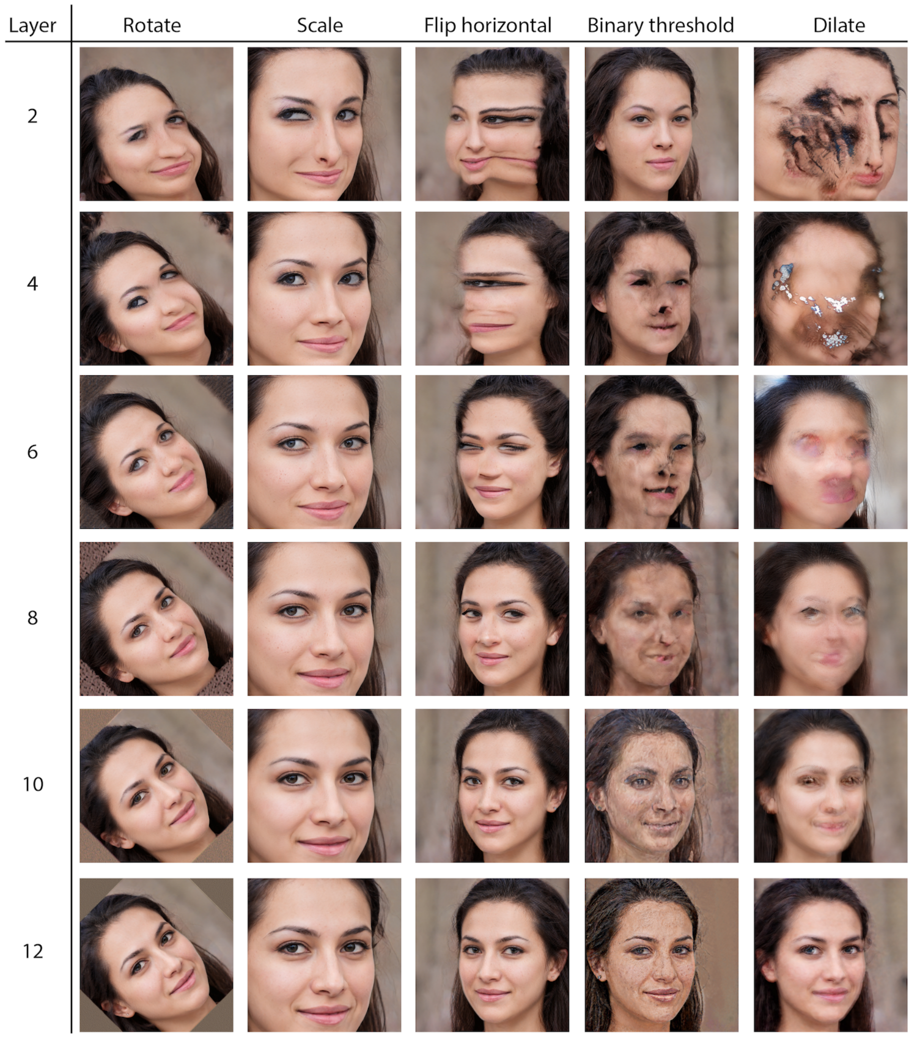


Figure 5.2: A comparison of various transformation layers inserted and applied to all of the features in different layers in the StyleGAN2 network trained on the FFHQ dataset, showing how applying the same filters in different layers can make wide-ranging changes the generated output. The rotation transformation is applied by an angle $\theta = 45$. The scale transformation is applied by a factor of $k_x = k_y = 0.6$. The binary threshold transformation is applied with a threshold of $t = 0.5$. The dilation transformation is applied with a structuring element with radius $r = 2$ pixels.

that:

$$f(x) = \begin{cases} 1, & \text{if } x \geq t \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

5.3.2 Affine Transformations

For this set of transformations, each activation map X for feature f is treated as an individual matrix that simple affine transformations can be applied to. The first two are horizontal and vertical *reflections* that are defined as:

$$X \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad X \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

The second is *translations* by parameters p_x and p_y such that:

$$X \begin{bmatrix} 1 & 0 & p_x \\ 0 & 1 & p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5.3}$$

The third is *scaling* by parameters k_x and k_y such that:

$$X \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.4}$$

Note that in this chapter, I only report on using uniform scalings, such that $k_x = k_y$. Finally, fourth is *rotation* by an angle θ such that:

$$X \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.5}$$

5.3.3 Morphological Transformations

I implemented two of the possible basic mathematical morphological transformation layers, performing *erosion* and *dilation* [Soille, 1999] when applied to the activation maps, which can be interpreted as 1-channel images (Fig. 5.3). These can be configured with the parameter r which is the radius for a circular kernel (aka structural element) used in the morphological transformations.

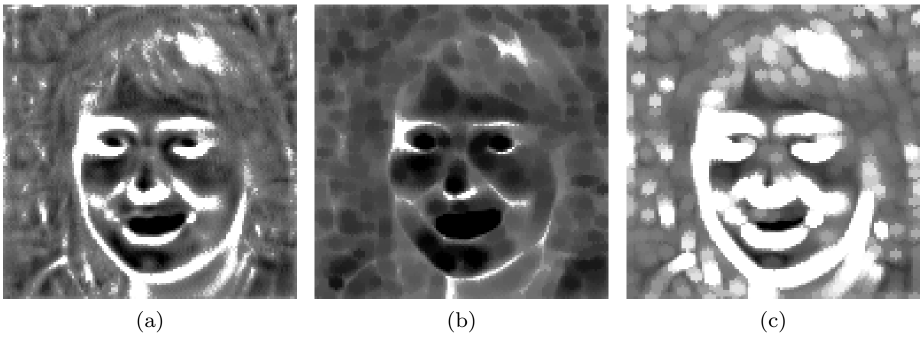

(a) (b) (c)

Figure 5.3: Examples of morphological transformations being applied to an individual activation map in Layer 10 of StyleGAN2. (a) Unmodified activation maps. (b) Activation map after erosion was applied ($r = 2$ pixels). (c) Activation maps after dilation were applied ($r = 2$ pixels).

5.4 Clustering Features

As most of the layers in the current state-of-the-art generative models, such as StyleGAN2, have very large numbers of convolutional features, controlling each one individually would be far too complicated to build a user interface around and control these in a meaningful way. In addition, because of the redundancy existing in these models, manipulating individual features does not normally produce any kind of meaningful outcome.[3] Therefore, it is necessary to find some way of grouping them into more manageable ensembles of sets of features. Ideally, such sets of features would correspond to the generation of distinct, semantically meaningful aspects of the image, and manipulating each set would

[3] I discovered this through my early hard-coded experiments with network bending that are discussed in Section 7.4.1.

correspond to the manipulation of specific semantic properties in the resulting generated sample. To achieve this, I developed a novel approach that combines metric learning and a clustering algorithm to group sets of features in each layer based on the spatial similarity of their activation maps. I trained a separate convolutional neural network (CNN) for each layer of StyleGAN2 to analyse the appearance of the activation maps. The CNN has a bottleneck architecture (first introduced by Grézl et al. [Grézl et al., 2007]) to learn a highly compressed feature representation; the latter is then used in a metric learning approach in combination with the k-means clustering algorithm [Lloyd, 1982, Celebi et al., 2013] to group sets of features in an unsupervised fashion.

5.4.1 Architecture

For each layer of StyleGAN2, I trained a separate CNN on the activation maps of all the convolutional features. The resolution of the activation maps and the number of convolutional features varies for the different layers of the model (a breakdown of which can be seen in Table 5.1). I employed an architecture that can dynamically be changed, by increasing the number of convolutional blocks, depending on what depth is required.

I employed the ShuffleNet architecture [Zhang et al., 2018] for the convolutional blocks in the network. For each convolutional block, I utilised a feature depth of 50 and had one residual block per layer. The motivating factor in many of the decisions made for the architecture design was not focused on achieving the best accuracy per se. Instead, I wanted a network that could learn a sufficiently good metric while also being reasonably quick to train (with 12-16 separate classifiers required to be trained per the StyleGAN2 model). I also wanted a lightweight enough network, such that it could be used in a real-time setting where clusters can quickly be calculated for an individual latent encoding, or when processing large batches of samples.

After the convolutional blocks, I flattened the final layer and used this to learn a mapping into a narrow bottleneck $\vec{v} \in \mathbb{R}^{10}$, before re-expanding the

Layer	Resolution	#features	CNN depth	#clusters	Batch size
1	8x8	512	1	5	500
2	8x8	512	1	5	500
3	16x16	512	2	5	500
4	16x16	512	2	5	500
5	32x32	512	3	5	500
6	32x32	512	3	5	500
7	64x64	512	4	5	200
8	64x64	512	4	5	200
9	128x128	256	5	4	80
10	128x128	256	5	4	80
11	256x256	128	6	4	50
12	256x256	128	6	4	50
13	512x512	64	7	3	20
14	512x512	64	7	3	20
15	1024x1024	32	8	3	10
16	1024x1024	32	8	3	10

Table 5.1: Table showing resolution, number of features of each layer, the number of ShuffleNet [Zhang et al., 2018] convolutional blocks for each CNN model used for metric learning, the number of clusters calculated for each layer using k-means and the batch size used for training the CNN classifiers for the StyleGAN2 models. Note: LSUN church and cat models have only 12 layers.

dimensionality of the final layer to the number of convolutional features present in the layer of the respective generative model. The goal of this bottleneck is to force the network to learn a highly compressed representation of the different convolutional features in the generative model. While this invariably loses some information, most likely negatively affecting classification performance during training, this is in fact the desired result. I wanted to force the CNN to combine features of the activation maps with similar spatial characteristics so that they can easily be grouped by the clustering algorithm. Another motivating factor is that the chosen clustering algorithm (k-means) does not scale well for feature spaces with high dimensionality.

5.4.2 Training

I generated a training set of the activations of every feature for every layer of 1000 randomly sampled images and a test set of 100 samples for the models trained on all of the datasets used in these experiments. I trained each CNN using the softmax feature learning approach [Dosovitskiy et al., 2014], a reliable method for distance metric learning. This method employs the standard softmax training regime [Bridle, 1990] for CNN classifiers. Each classifier has been initialised with random weights and then trained for 100 epochs using the Adam optimiser [Kingma and Ba, 2015] with a learning rate of 0.0001 and with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. All experiments were carried out on a single NVIDIA GTX 1080ti. The batch size used for training the classifiers for the various layers of StyleGAN2 can be seen in Table 5.1.

After training, the softmax layer is discarded and the embedding of the bottleneck layer is used as the discriminative feature vector where the distances between points in feature space permit gauging the degree of similarity of two samples. This approach differs from standard softmax feature learning as it uses the feature vector from the bottleneck, rather than the last layer prior to softmax classification, giving a more compressed feature representation than the standard softmax feature learning approach.

5.4.3 Clustering Algorithm

Once each of the CNNs for every layer has been trained, they can then be used to extract feature representations of the activation maps of the different convolutional features corresponding to each layer of the generative model. The approach is to perform clustering based on an average of features' embeddings drawn from many random samples, which can be used to find a general-purpose set of clusters for a trained model.

The activation map X_{df} for each layer d and feature f is fed into the CNN metric learning model for that layer C_d to get the feature vector $\vec{v}_{df}$. This process is repeated N times (1000 in these experiments) to find the mean feature vector $\bar{\vec{v}}_{df}$ for each convolutional filter. The mean feature vectors for each filter in each layer are then aggregated and fed to the k-means clustering algorithm — using Lloyd's method [Lloyd, 1982] with Forgy initialization [Forgy, 1965, Celebi et al., 2013].

The predetermined number of clusters for each layer in StyleGAN2 can be seen in Table 5.1. Examples from the clustering algorithm applied to the FFHQ StyleGAN2 model can be seen in Figure 5.4.

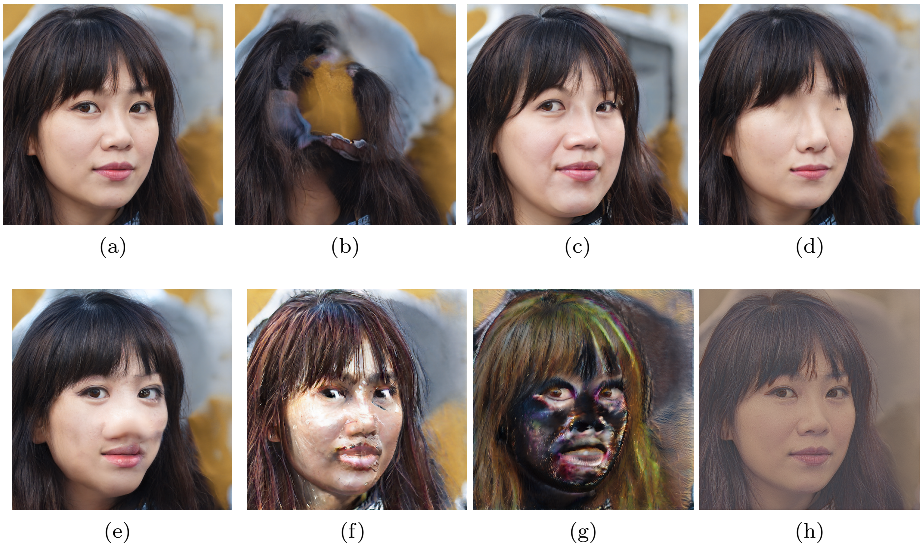


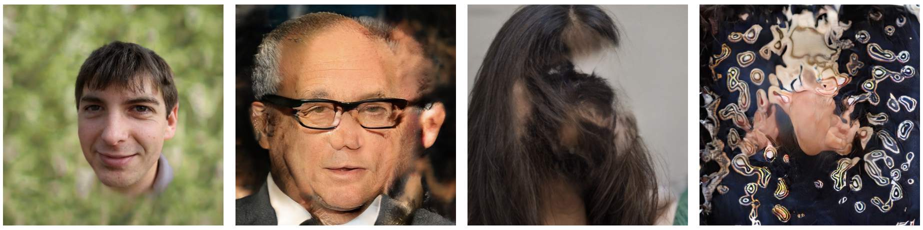
Figure 5.4: Examples from the clustering algorithm in the image domain. Clusters of features in different layers of the model are responsible for the formation of different image attributes. (a) The unmanipulated result. (b) A cluster in layer 1 has been multiplied by a factor of -1 to completely remove the facial features. (c) A cluster in layer 3 has been multiplied by a factor of 5 to deform the spatial formation of the face. (d) A cluster in layer 6 has been ablated to remove the eyes. (e) A cluster in layer 6 has been dilated with a structuring element with radius $r = 2$ pixels to enlarge the nose. (f) A cluster in layer 9 has been multiplied by a factor of 5 to distort the formation of textures and edges. (g) A cluster of features in layer 10 has been multiplied by a factor of -1 to invert the highlights on facial regions. (h) A cluster of features in layer 15 has been multiplied by a factor of 0.1 to desaturate the image. All transformations have been applied to sets of features discovered using the feature clustering algorithm (§5.4) in the StyleGAN2 model trained on the FFHQ dataset.

The main motivation of the clustering algorithm presented in this paper was to simplify the parameter space in a way that allows for more meaningful and controllable manipulations whilst also enhancing the expressive possibilities afforded by interacting with the system. These results show that the clustering algorithm is capable of discovering groups of features that correspond to the generation of different semantic aspects of the results, which can then be manipulated in tandem. These semantic properties are discovered in an unsupervised fashion and across the entire hierarchy of features present in the generative model. Figure 5.4 shows the manipulation of groups of features across a broad range of layers that control the generation of the entire face, the spatial formation of facial features, the eyes, the nose, textures, facial highlights and overall image contrast.

5.5 Manipulation Pipeline

Transforms are specified in YAML (YAML Ain't Markup Language) configuration files [Ben-Kiki et al., 2009] (Fig. 5.6 for an example of one of these configs), such that each transform is specified with 5 items: (i) the layer, (ii) the transform itself, (iii) the transform parameters, (iv) the layer type (i.e. how the features are selected in the layer: across all features in a layer, to pre-defined clusters, or to a random selection of features), and (v) the parameter associated with the layer type (either the cluster index, or the percentage of features the filter will randomly be applied to). Visual examples of how different layer types can be seen in Figure 5.5. There can be any number of transforms defined in such a configuration file and transforms can be chained together to produce more complex filtering effects in the generated output (Fig. 5.5d).

After loading the configuration, the software either looks up which features are in the cluster index or randomly applies indices based on the random threshold parameter. Then the latent is loaded, which can either be randomly generated, or be predefined in latent space z, or be calculated using a projection in latent space w [Abdal et al., 2019, Karras et al., 2020] (in the case of StyleGAN2). The latent code is provided to the generator network and inference is performed. As this implementation uses PyTorch [Paszke et al., 2019], a dynamic neural network library, these transformation layers can therefore be inserted dynamically during inference as and when they are required and applied only to the specified features as defined by the configuration. Once inference is unrolled, the generated output is returned. Figure 5.1 provides a visual overview of the pipeline, as well as a comparison between a modified and unmodified generated sample.



(a) (b) (c) (d)

Figure 5.5: Examples of transformation layers being applied to different configurations of features in StyleGAN2. (a) Transformation applied layer-wide. (b) Transformation is applied to a random selection of filters in one layer. (c) Transformation applied to a cluster in layer 2. (d) A combination of transformation layers applied across a network, the configuration of transformations used to generate this image can be seen in Figure 5.6.

```yaml
---
transforms:
- layer: 2
  transform: "invert"
  params: []
  features: "all"
  feature—param:
- layer: 6
  transform: "binary-thresh"
  params: [0.5]
  features: "random"
  feature—param: 0.5
- layer: 6
  transform: "scalar-multiply"
  params: [5]
  features: "cluster"
  feature—param: 2
---
```

Figure 5.6: Example of a YAML transformation config that is used in the network bending framework. This config combines randomly applied layers and layer-wide transformations. This config was used to generate the image Figure 5.5d.

5.6 Network Bending in the Audio Domain

As a follow-up study to the original network bending approach on images, I applied the same approach to the audio domain as an extension to the work for the journal paper in Entropy [Broad et al., 2022]. The motivation for this was to demonstrate that network bending was applicable to different types of media and to demonstrate the general-purpose nature of this framework.

For this study, I trained a custom VAE model on spectrograms of music[4] and applied the exact same algorithm for clustering and applying the same transformation layers as was applied in network bending for image generation. Network bending has also been applied to audio by other researchers and practitioners, this efforts are detailed in Sections 7.5.2.4 & 7.6.1.

5.6.1 Custom Audio Model

For this experiment, I trained a variational autoencoder (VAE) [Kingma and Welling, 2013, Rezende et al., 2014] on spectrograms extracted from a custom dataset of varied musical genres, totalling 3461 audio tracks. This approach is based on previous methods for learning generative models of spectrograms [Akten, 2018] and Mel spectrograms [Valenzuela, 2021] with VAEs. The tracks are randomly split up into short sequences and the Fourier transform is performed with a hop size of 256 and a window size of 1024 to produce spectrograms that have a bin size of 513. The spectrograms are then cut into shorter sequences of a window length of 128. These shortened spectrograms are then converted to decibels and then normalised for training with the VAE.

The VAE was built using a convolutional architecture with a latent vector with dimension $\vec{v} \in \mathbb{R}^{512}$. The encoder has 5 layers that use standard convolutions with a kernel size of 5x5, a stride of 2x2 and no padding for all of

[4] Whilst training a model from scratch on a large music dataset does take away somewhat from the 'data-free' stance of this thesis, this was done purely for the purposes of demonstrating the generalisability of network bending to a different data domain. The dataset was a personal and legally purchased music collection. Training on this dataset was permissible under the UK's Text and Data Mining copyright exemption for non-commercial research [UK Government, 1988]. Neither the dataset, nor the trained model were made publicly available after training.

Layer	Resolution	#features	kernel size	stride	padding
1	8x33	512	5x5	1x2	0x2
2	17x65	256	3x5	2x2	2x2
3	32x129	128	4x5	2x2	2x2
4	64x257	64	4x5	2x2	2x2
5	128x513	1	4x5	2x2	2x2

Table 5.2: Table showing resolution, number of features of each layer, convolutional kernel size, strides, and padding parameters for the decoder network in the spectrogram VAE.

the layers. The decoder uses transposed convolutions, and Table 5.2 lists the output resolution, kernel size, stride, and padding parameters for each of the 5 convolutional layers. A fully connected layer is used in both the encoder and decoder to interface between the convolutional layers and the latent vector. The model was trained for 50 epochs on the dataset with batch normalisation using a batch size of 64. The model was trained using the Adam optimiser [Kingma and Ba, 2014] with a learning rate of 0.0003 and with $\beta_1 = 0$ and $\beta_2 = 0.99$.

After training it is possible to sample randomly in the latent space and then sample directly from the decoder. It is also possible to input audio sequences, both from the training set and outside of it, and produce reconstructions of the audio track mediated through the VAE model, in a method that I have previously referred to as *autoencoding* [Broad and Grierson, 2017]. Performing this autoencoding procedure in combination with network bending, provides a new way of transforming and filtering audio.

5.6.2 Clustering

the approach to clustering for this audio model was identical to what was demonstrated in Section 5.4. As the VAE model did not have as many layers as StyleGAN2, clusters were only calculated for four layers, the details of which can be seen in Table 5.3.

Layer	CNN Depth	#clusters
1	1	5
2	2	5
3	3	4
4	4	4

Table 5.3: Table showing the number of ShuffleNet [Zhang et al., 2018] convolutional blocks for each CNN model used for metric learning and the number of clusters calculated for each layer using k-means.

5.6.3 Results

The clustering approach applied to the audio model appears to work well when visualising the spectrograms, and it is clear that this approach can capture and manipulate some semantically meaningful components in the audio signal[5] (Fig. 5.7). Not all of the transformations that can be applied to images work as well in audio, such as scaling and rotation. This is not a surprise given that the location of each pixel is essential information used to represent frequency and time information in the audio signal, and can completely transform the information represented when manipulated. However, the morphological transformations do at least preserve locality in the signal, and using these filters in generative models of spectrograms offers a completely new way to transform audio signals.

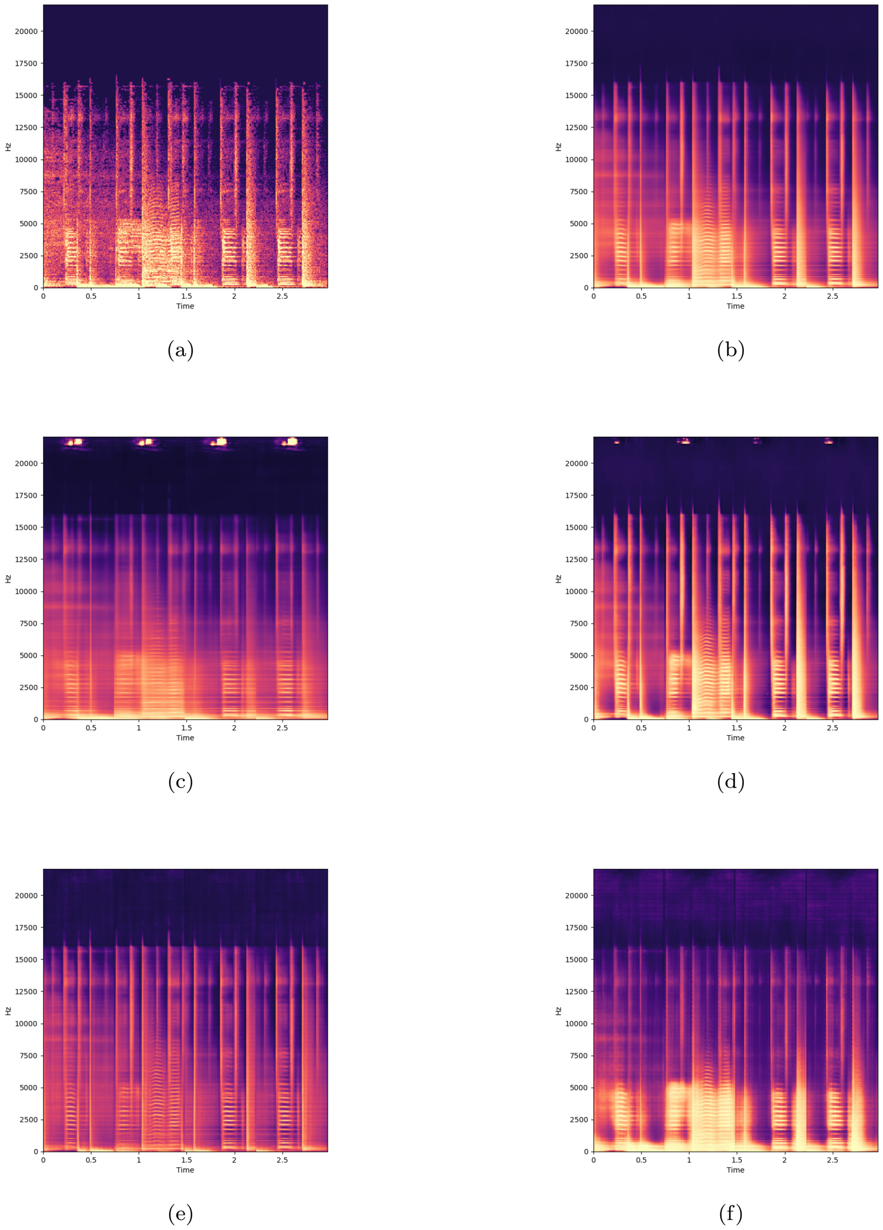
5.7 Discussion

In this section, I discuss several different perspectives on the outcomes presented here: expressive manipulation, active divergence, comparisons of the results between the image and audio domains, and comparisons with other methods.

5.7.1 Expressive Manipulation

The main motivation of the clustering algorithm presented in this chapter was to simplify the parameter space in a way that allows for more meaningful and

[5] Unfortunately there was a bug in the decoding of the spectrograms back into audio which meant that the audio quality in the generated samples was very noisy – something that I have not had the time to fix.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.7: Examples from the clustering approach in the audio domain. (a) Spectrogram of an original source track not in the training set. (b) Reconstruction of source track using VAE without manipulation. (c) Reconstruction of the same signal where a cluster in layer 1 responsible for the generation of the transients of the signal has been ablated. (d) Reconstruction of the same signal where the same cluster in layer 1 responsible for the transients has been multiplied by a factor of 2, increasing the intensity of the transients in the resulting signal. (e) Reconstruction of the signal where a cluster in layer 3 responsible for the low and mid-range frequencies has been eroded with a structuring element with radius $r = 2$ pixels, diminishing the intensity of these frequency components. (f) Reconstruction of the signal where the same cluster in layer 3 responsible for the low and mid-range frequencies has been dilated with a structuring element with radius $r = 2$ pixels, increasing the intensity of these frequency components. The audio sample used is a clip from *Saulsalita Soul* by Mr.RuiZ, reproduced and transformed with permission granted under the CC BY-NC 4.0 licence. All the audio samples shown can be listened to here: `https://drive.google.com/drive/folders/1KjuG2MOU9ngO1a3yMsAA32eVslBzFVW7?usp=sharing`.

controllable manipulations whilst also enhancing the expressive possibilities afforded by interacting with the system. These results show that the clustering algorithm is capable of discovering groups of features that correspond to the generation of different semantic aspects of the results, which can then be manipulated in tandem. These semantic properties are discovered in an unsupervised fashion and across the entire hierarchy of features present in the generative model. For example, Figure 5.4 shows the manipulation of groups of features across a broad range of layers that control the generation of the entire face, the spatial formation of facial features, the eyes, the nose, textures, facial highlights and overall image contrast. Figure 5.7 shows the clustering algorithm performed in the audio domain, to demonstrate how aspects of the audio signal such as the transients and frequency components can be manipulated with various kinds of transformations.

Grouping and manipulating features in a semantically meaningful fashion is an important component of allowing expressive manipulation. However, artists are often also ready to consider surprising, unexpected results, to allow for the creation of new aesthetic styles, which can become uniquely associated with an individual or group of creators. Therefore the tool needs to allow for unpredictable as well as predictable possibilities, which can be used in an exploratory fashion and can be mastered through dedicated and prolonged use [Dobrian and Koppelman, 2006]. There is usually a balance between the utility and expressiveness of a system [Jacobs et al., 2017].

Section 7.4 shows the various different ways this framework has been used to make artworks. Whilst I did not make a user interface for network bending myself, many other researchers have gone on to do so. Their efforts are detailed in Section 7.5.2.

5.7.2 Comparison Between Audio and Image Domains

In this chapter, I have described the network bending framework in both the image and audio domains. For the image domain, I have used StyleGAN2 [Karras et al., 2020], the state of the art generative model for unconditional image generation. In the audio domain, I have built a custom generative model to demonstrate how the same principles of clustering features and applying transformations to clustered features can be applied indirectly to another domain. The generative model for audio I have presented is building on a much smaller body of research and has more room for improvement in terms of the fidelity of the generated outputs, however, it is still adequate and demonstrates that the clustering algorithm is capable of discovering semantically meaningful components of the signal (Fig. 5.7). Some of the transformation layers that were designed for image-based models such as rotation and scaling do not transfer meaningfully into the audio domain. However, numerical and morphological transformations do work effectively in the audio domain, representing a completely new approach for manipulating audio signals. In addition to my efforts, other researchers have also gone on to successfully implement network bending in audio models (§7.5.2.4 & §7.6.1).

5.7.3 Comparison with Other Methods

With respect to the semantic analysis and manipulation of a generative model, this approach of clustering features and using a broad array of transformation layers is a significant advance over previous works [Bau et al., 2017, 2018, 2019, Brink, 2019]. This recent thread of techniques only interrogates the function of individual features, and as such is unlikely to be capable of capturing a full account of how a deep network generates results since such networks tend to be robust to the transformation of individual features.

The results in this chapter show that sets of features, which may not be particularly responsive to certain transformations, are very responsive to others. Figure 5.8 shows that in the model trained on the LSUN church dataset, a cluster of features, when ablated, have little noticeable effect on the result. However, significant changes are visible when using the pointwise scalar multiplication transformation on the same cluster, here removing the trees and revealing the church building that was obscured by the foliage in the original result. The clustering approach described in this paper suggests that the functionality of features, or sets of features, cannot be understood only through ablation, because of the high levels of redundancy present in the learned network parameters. In addition, the research here shows that their functionality can be better understood by applying a wide range of deterministic transformations, of which different transformations, some of which are better suited to revealing the utility of different sets of features (Figs. 5.4 & 5.8). An approach that has since been developed further by Oldfield et al. [2023, 2024].

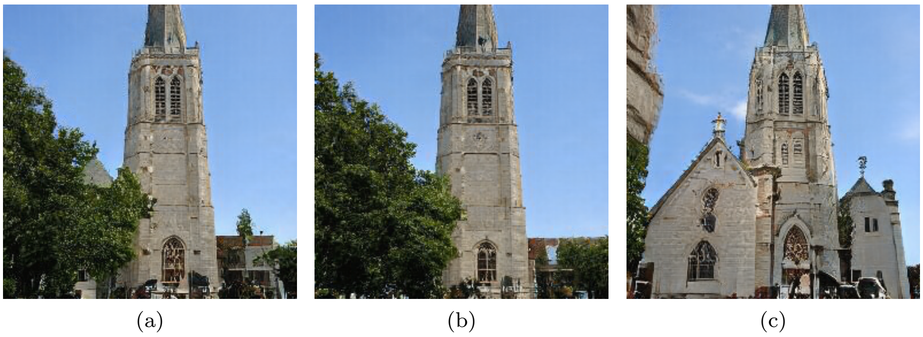


Figure 5.8: Groups of features that are not particularly sensitive to ablation may be more sensitive to other kinds of transformation. (a) Original unmodified input. (b) A cluster of features in layer 3 that has been ablated. (c) The same cluster of features that has been multiplied by a scalar of 5. As can be seen, ablation had a negligible effect, only removing a small roof structure that was behind the foliage. On the other hand, multiplying by a factor of 5 removes the trees whilst altering the building structure to have gable roof sections on both the left and right sides of the church - which are now more prominent and take precedence in the generative process. Samples are taken from the StyleGAN2 model trained on the LSUN church dataset.

This method of analysis is completely *unsupervised* and does not rely on auxiliary models trained on large labelled datasets (such as in [Bau et al., 2018, Isola et al., 2017, Park et al., 2019]) or other kinds of domain-specific knowledge. This approach therefore can be applied to any CNN-based generative model architecture which has been trained on any dataset, as I have demonstrated by using the exact same clustering method for both image and audio domains. This is of particular relevance to artists who create their own datasets and would want to apply these techniques to models they have trained on their own data. Labelled datasets are prohibitively time-consuming (and expensive) to produce for all but a few individuals or organisations. Having a method of analysis that is completely unsupervised and can be applied to unconditional generative models is important in opening up the possibility for such techniques to become adopted more broadly. Section 7.4 details a number of artworks made by myself and others, applied to a range of datasets, both pre-existing and custom. The limitation of this approach is the time and computational resources needed to train a separate model for each layer of the network. This limitation is discussed further in Section 9.2, and ways to improve upon this are further presented in Section 9.3.

5.8 Conclusion

In this chapter, I have introduced a novel approach for the interaction with and manipulation of generative neural networks, which has been demonstrated in both the image and audio domains. By inserting deterministic filters inside pre-trained neural networks, this framework allows for manipulation to be performed inside the networks' 'black-box', generating samples that have no resemblance to the training data, or anything that could be created easily using conventional media editing software. This chapter also presents a novel clustering algorithm that can group sets of features in an unsupervised fashion, based on the spatial similarity of their activation maps. I have demonstrated that this method is capable of finding sets of features that correspond to the generation of a broad array of semantically significant aspects of the generated results in both image and audio domains.

The goal of the work in this thesis was to find a way to expand the possibility space of what can be generated with neural networks. *Network bending* is an approach that expands the generative space of existing pre-trained models in a way that gives direct control and agency to artists and creative practitioners, and in a way that *actively diverges* from data. This now concludes the documentation of new algorithms and approaches to *active divergence* presented in this thesis. The next chapter will a broader perspective, contextualising the work in this thesis with other related efforts that occurred during its development. Chapter 6 gives a detailed survey and taxonomy of active divergence methods, placing the work presented in this thesis into a larger context and delineating and outlining the landscape of active divergence methods to date.

Chapter 6

Surveying The Active Divergence Landscape

6.1 Introduction

This chapter is an updated version of the survey paper and taxonomy of active divergence methods that I presented at the International Conference on Computational Creativity (ICCC) in 2021 [Broad et al., 2021a] in collaboration with Sebastian Berns and Simon Colton from Queen Mary, University of London. The concept of active divergence was introduced by Berns and Colton [2020] at ICCC the year before, and this survey is a follow-up to that first introduction of active divergence, giving a comprehensive account of all active divergence methods that were published and disseminated in 2021. This survey and taxonomy was developed primarily by myself. Berns and Colton were brought on as collaborators later in the process, to get their insights and perspectives on the survey from the viewpoint of computational creativity research, which is the primary community this survey was first disseminated within.

Much like the Berns and Colton [2020] paper, this survey bridges together two related but distinct fields of AI research and creative practice: CreativeAI research and computational creativity [Cook and Colton, 2018]. Many of the novel advances in this survey were developed by creative practitioners working in the CreativeAI communities, where work is primarily shared in the NeurIPS Workshop on Machine Learning for Creativity and Design, as well as developments being shared on social media and open-source code channels like Twitter, GitHub, and Google Colab notebooks. A great deal of effort was made on my part to find and document the original contributions, regardless of where they were disseminated, so as not to simply rely on peer-reviewed machine learning and computational creativity papers, which would have given only a partial view of the developments in this space when the CreativeAI community was flourishing between 2017-2021.

All of the experimental work in this thesis from Chapters 3, 4 & 5 are presented in this survey and taxonomy. It is unconventional to present a survey at the end of the thesis, and include the work done by the author in the survey, but to exclude my own contributions in this survey would give an incomplete picture of the active divergence landscape, given that the three chapters of experimental work in this thesis are each three different categorical contributions to active divergence methods. I also decided to put this survey at the end of the thesis to better reflect the timeline of events. Much of the work by others in active divergence happened concurrently with my own work, so would have given a misleading chronology if this were to have all been documented in the background chapter of this thesis (Ch. 2). Any works that precede this thesis are also documented in the background chapter (§2.8), but presented again here in the context of the wider developments in this survey and taxonomy.

This survey starts with a statistical view of standard generative model training and then proceeds with the many different ways that active divergence can be achieved in relation to this statistical view. The definition of divergence in active divergence refers to divergence in the statistical sense, which is the distance between two data distributions, and should not be confused with other definitions of divergence, such as divergent thinking from psychology and creativity research [Guilford, 1957] (§2.2.1.1).

6.2 Generative Models: A Statistical View

Given a data distribution P, a generative model will model an approximate distribution P'. The parameters for the approximate distribution can be learned by an artificial neural network. This learning task is tackled differently by different architectures and training schemes. E.g. autoencoders [Rumelhart et al., 1985] and variational autoencoders (VAE) [Kingma and Welling, 2013, Rezende et al., 2014] learn to approximate the data through reconstruction via an encoding and a decoding network, while generative adversarial networks (GAN) [Goodfellow et al., 2014] consists of a generator that is guided by a discriminating network. In most cases, the network learns a mapping from a lower-dimensional latent distribution X to the complex high-dimensional feature space of a domain. The model, thus, generates a sample p' given an input vector x which should resemble samples drawn from the target distribution P. In the simplest case of a one-layer network the generated sample p' is generated using the function: $p' = \sigma(Wx + b)$ where x is the input vector from the latent distribution $x \in X$, σ is a non-linear activation function, W and b are the learned association matrix and bias vector for generating samples in the approximate distribution $p' \in P'$. The model parameters W and b, are typically learned through a gradient-based optimisation process. In this process, a loss function will require the model to maximise the likelihood of the data either: (i) explicitly, as in the case of autoencoders, and autoregressive models [Frey et al., 1996]; (ii) approximately, as is the case in VAEs; (iii) or implicitly, as in the case of GANs. Generative models can also be conditioned on labelled data. In the conditional case, the generative model takes two inputs x and y, where y represents the class label vector. Another form of conditional generative models is translation models, such as pix2pix [Isola et al., 2017], that takes a (high dimensional) data distribution as input Q and learns a mapping to P' which is an approximation of the true target function $f : Q \to P$.

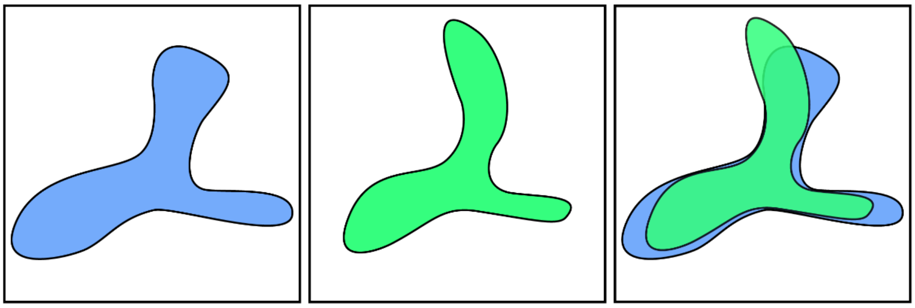


Figure 6.1: Diagram illustrating the parameter view of training a generative model. **Left:** The true distribution P. **Middle:** The approximate distribution P'. **Right:** The approximate distribution P' overlayed on the true distribution P.

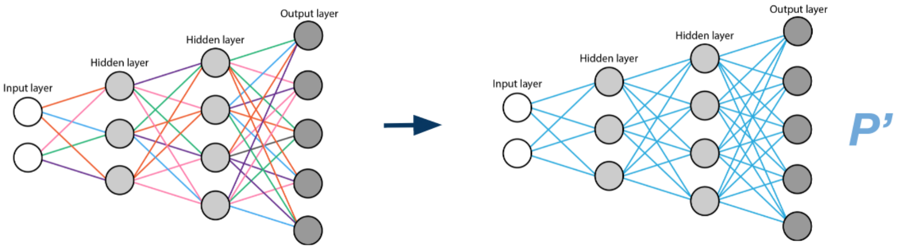


Figure 6.2: Diagram illustrating the parameter view of training a generative model. A network with randomly initialised parameters is trained to model the true distribution P and produces the approximate distribution P'

.

All deep generative models, and in particular ones that generate high dimensional data domains like images, audio and natural language, will have some level of divergence $D(P||P') \geq 0$ between the target distribution P and the approximate distribution P', because of the complexity and stochasticity inherent in high dimensional data. The goal of all generative models is to minimise that level of divergence, by maximising the likelihood of generating the given data domain. Active divergence methods, however, intentionally seek to create a new distribution U that does not directly approximate a given distribution P, or resemble any other known data distribution. This is either done by seeking to find model parameters W^* and b^* (in the single layer case) that generate novel samples $u = \sigma(W^*x + b^*)$ or by making other kinds of interventions to the chain of computations.

6.3 Taxonomy of Active Divergence Methods

This section presents the taxonomy and survey of active divergence methods. For three of these categories of active divergence methods, I have made major contributions, being the first to publish examples of all of these methods, (detailed in Chs. 3, 4 & 5). This section will reiterate these contributions, for the purposes of defining, formally explaining and delineating them from other approaches.

6.3.1 Novelty Search Over Learned Representations

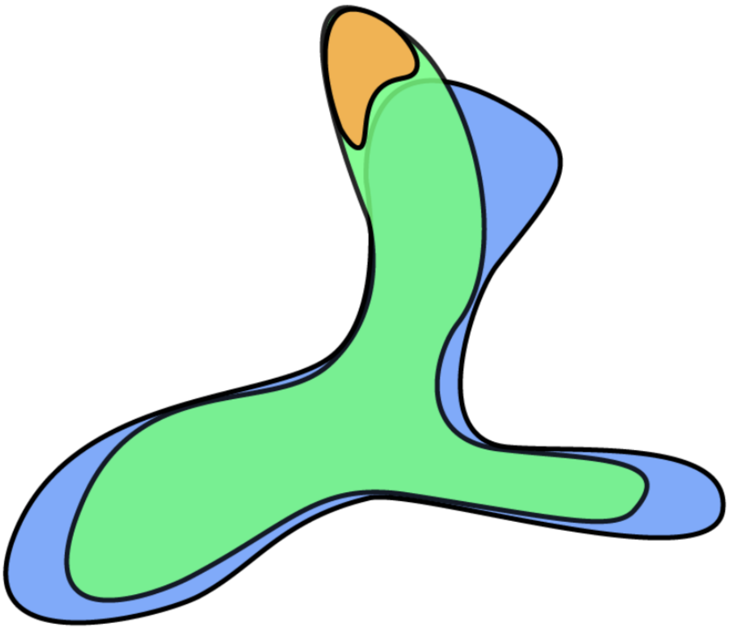


Figure 6.3: Diagram illustrating the distribution view of novelty search over learned representations which finds the subset U of the approximate distribution P' that is not present in true distribution P.

Methods in this category take existing generative models trained using standard maximum likelihood regimes and then specifically search for the subset of learned representations that do not resemble the training data by systematically sampling from the model. Taking account of the fact that any approximate distribution P' will be somewhat divergent from the true distribution P, these methods seek to find the subset U of the approximate distribution which is not contained in the true distribution $U \subset P' \wedge U \not\subset P$. Kazakçı et al. [2016] present an algorithm for searching for novelty in the latent space of a sparse autoencoder trained on the MNIST dataset [LeCun et al., 1998]. They start by creating a sample of random noise and by using a Markov chain Monte Carlo (MCMC) method of iteratively re-encoding the sample through the encoder, then refining the sample until it produces a stable representation. They use this approach to map out all the representations the model can generate, then perform k-means clustering on the latent space encoding of these representations. By disregarding clusters that correspond to real digits, they are left with clusters of representations of digits that do not exist in the original data distribution. It has been argued that these 'spurious samples' are the inevitable outcome of generative models that learn to generalise from given data distributions [Kégl et al., 2018] and that there is a trade-off between the ability to generalise to every mode in the dataset and the ratio of spurious samples in the resulting distribution.

6.3.2 Novelty Generation from an Inspiring Set

The methods in this section train a model from scratch using a training dataset but do not attempt to model the data directly, rather using it as reference material to draw inspiration from. We, therefore, refer to this training set (the given distribution P) as the inspiring set [Ritchie, 2007].

An approach for novel glyph generation utilises a class-conditional generative model trained on the MNIST dataset [LeCun et al., 1998], but in this case, they train the model with 'hold-out classes' [Cherti et al., 2017], additional classes that do not exist in the training data distribution. These hold-out classes can then be sampled during inference, which encapsulates the subset U

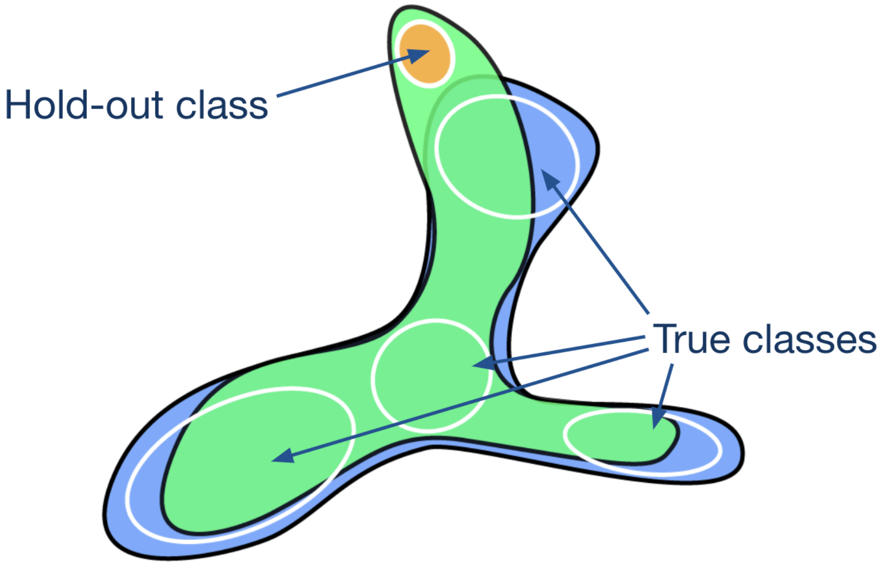


Figure 6.4: Diagram illustrating the distribution view of using hold-out classes to encapsulate the subset U of the approximate distribution P' that is not present in true distribution P.

of the approximate distribution P' that is not included in the target distribution $U \subset P' \wedge U \not\subset P$. These divergent samples can then be generated directly by conditioning the generator with the hold-out class label, without the need for searching the latent space.

An approach that directly generates a new distribution U from an inspiring set P is the Creative Adversarial Networks (CAN) algorithm [Elgammal et al., 2017]. The algorithm uses the WikiArt dataset [Saleh and Elgammal, 2016], a labelled dataset of paintings classified by ‘style’ (historical art movement). This algorithm draws inspiration from the GAN training procedure [Goodfellow et al., 2014], but adapts it such that the discriminator has to classify real and generated samples by style, and the generator is then optimised to maximise the likelihood of the generated results being classified as ‘artworks’ (samples that fit the training distribution of existing artworks), maximising their deviation from existing styles in order to produce the novel distribution U. A similar approach is also taken by Chemla–Romeu-Santos and Esling [2022] with their

Bounded Adversarial Divergence (BAD) algorithm for training generative neural networks to diverge from existing labelled classes, but to maintain generated samples within the overall training data distribution.

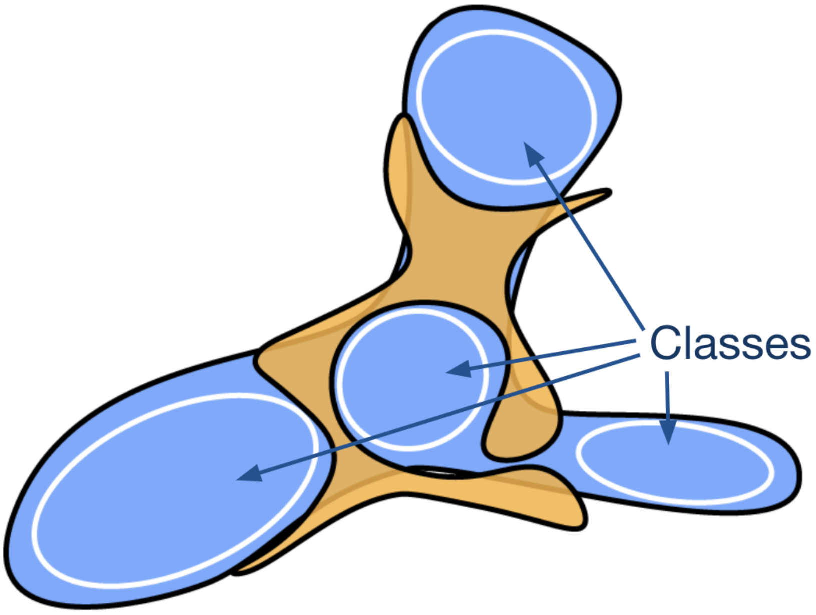


Figure 6.5: Diagram illustrating novelty generation from an inspiring set, from the distribution view in the creative adversarial networks framework [Elgammal et al., 2017] that learns to generate the distribution U by fitting the true distribution P which divergence from the existing classes within the distribution

6.3.3 Training Without Data

Training a model from a random initial starting point without any training data almost certainly guarantees novelty in the resulting generated distribution. Existing approaches to doing this all rely on the dynamics between multiple models to produce emergent behaviours through which novel data distributions can be generated.

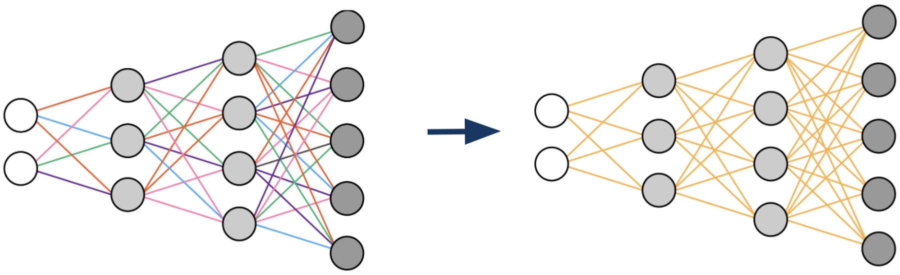


Figure 6.6: Diagram illustrating the network parameter view of training without data. A randomly initialised network is trained without data to learn a novel distribution U.

6.3.3.1 Multi-Generator Dynamics

The work in Chapter 3 (originally disseminated in [Broad and Grierson, 2019a]) is an approach to training generative deep learning models without any training data, by using two generator networks and relying on the dynamics between them for an open-ended optimisation process. In order to have some level of diversity in the final results, the two generators are simultaneously trying to produce more colours in the generated output than the other generator network, leading to the generation of two novel, yet closely related distributions U and V.

6.3.3.2 Generation via Communication

An alternative approach to generating without data uses a single generator network and uses the generated distribution U as a channel for communication between two networks, which together learn to generate and classify images that represent numerical and textual information from a range of existing datasets [Simon, 2019].[1] In subsequent work, by constraining the generator with a strong inductive bias for generating line drawings, this approach can be utilised for novel glyph generation [Park, 2020].

[1] As far as I am aware, this work was done simultaneously and independently of the work presented in Chapter 3.

6.3.4 Divergent Fine-Tuning

Divergent fine-tuning methods take pre-trained models that generate an approximate distribution P' and fine-tune the model away from the original training data. This can either be done by optimising based on new training data, or by using auxiliary models and custom loss functions, the goal being to find a new set of model parameters that generate a novel distribution U, that is significantly divergent from the approximate distribution P' and the original distribution P.

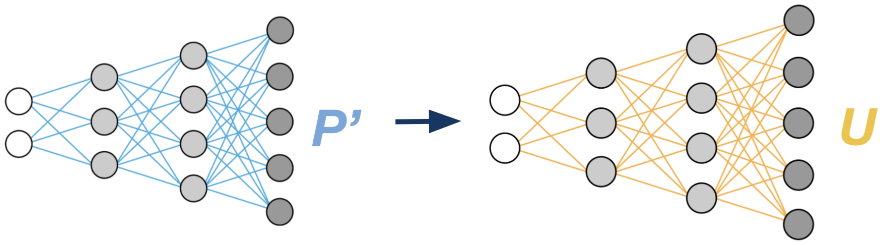


Figure 6.7: Diagram illustrating the network parameter view of divergent fine-tuning. A network pre-trained on the distribution P and can produce the approximate distribution P' is fine-tuned in a divergent fashion to create a novel distribution U.

6.3.4.1 Cross-Domain Training

In cross-domain training, transfer learning is performed to a pre-trained model that generates the approximate distribution P' and is then trained to approximate the new data distribution Q. This transfer learning procedure will eventually lead to the model learning a set of parameters that generate the approximate distribution Q'. However, by picking an iteration of the model mid-way through this process, a set of parameters can be found that produced a blend between the two approximate distributions P' and Q', resulting in the producing the novel distribution U [Schultz, 2020a]. This method was discovered by many artists and practitioners independently, who were performing transfer learning with GAN models for training efficiency, but noted that the iterations of the model part-way through produced the most interesting, surprising and sometimes horrifying results [Adler, 2020, Black, 2020, Mariansky, 2020, Shane, 2020].

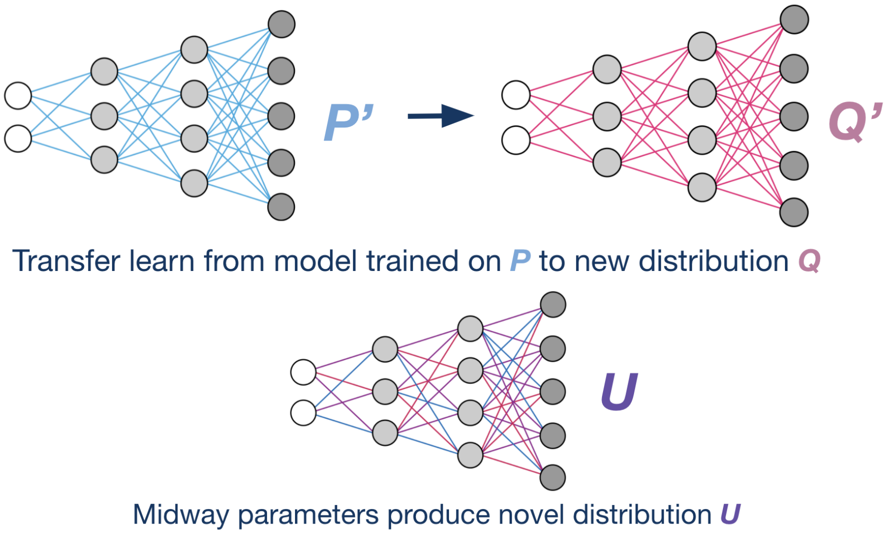

P'
Q'
Transfer learn from model trained on P to new distribution Q
U
Midway parameters produce novel distribution U

Figure 6.8: Diagram illustrating the network parameter view of cross-domain training. A network pre-trained on the distribution P that produces the approximate distribution P' is used as the starting point for transfer learning to a new distribution Q that will eventually learn to produce the approximate distribution Q'. If early stopping is performed through the transfer learning process, a set of parameters for the network that produces the novel hybrid distribution U can be ascertained.

6.3.4.2 Continual Domain Shift

Going beyond simply mixing two domains, one approach that gives more opportunity to steer the resulting distribution in the fine-tuning procedure, is to optimise on a domain that is continually shifting. In creating the artworks *Strange Fruit* [Som, 2020], the artist Mal Som 'iterate[s] on the dataset with augmenting, duplicating and looping in generated images from previous ticks' to steer the training of the generator model [Som, 2021]. In this process, the target distribution Q_t at step t may contain samples q'_{t-n} generated from earlier iterations of the model at any previous time step $t-n$ where $0 < n < t$. Additionally, the target distribution Q_t, may no longer include samples or may have duplicates of samples q_{t-n} from previous iterations of the target distribution. Using this process, the target distribution can be continually shaped and guided.

This process of modelling a continually shifting domain often leads to the —generally unwanted— phenomenon of mode collapse [Thanh-Tung and Tran, 2020]. However, in Som's practice, this is induced deliberately. After a model has collapsed, Som explores its previous iterations to find the last usable instance right before collapse. Som likens this practice to the artistic technique of defamiliarisation, where common things are presented in unfamiliar ways so audiences can gain new perspectives and see the world differently [Som, 2021].

Som's artistic experiments are a precursor to subsequent research studies that demonstrate that generative models that are subsequently trained on their own synthetic outputs, regularly lead to mode (or model) collapse [Alemohammad et al., 2023, Martínez et al., 2023, Shumailov et al., 2023, 2024]. An issue that is becoming so widespread, with the outputs of generative AI polluting the internet with 'low-quality' data, that the issue has even made its way into the popular and business press [Peel, 2024, Bhatia, 2024].

6.3.4.3 Loss Hacking

An alternative strategy is to fine-tune a model without any training data. Instead, a loss function is used that directly transforms the approximate distribution P' into a novel distribution U without requiring any other target distribution. Chapter 4 uses the frozen weights of the discriminator to directly optimise based on the inverse likelihood of the data, by using the inverse of the adversarial loss function. This process reverses the normal objective of the generator to generate 'real' data and instead generates samples that the discriminator deems to be 'fake'. By applying this process to a GAN that can produce photorealistic images of faces, this fine-tuning procedure crosses the uncanny valley in reverse, taking images indistinguishable from real images, and amplifying the uncanniness of the images before eventually leading to mode collapse. In a similar fashion to Som's practice (§6.3.4.2), one instance of the model before mode collapse was hand-selected and a selection of its outputs turned into the series of artworks *Being Foiled* [Broad, 2020a] (§7.3).

6.3.4.4 Infusing External Knowledge

By harnessing the learned knowledge of externally trained models, it is possible to fine-tune models to infuse that knowledge to transform the original domain data with characteristics defined using the auxiliary model. In [Broad and Grierson, 2019b], I utilised a classifier model $C_{classifier}$ trained to differentiate between datasets, in conjunction with the frozen weights of the discriminator D_{frozen} to fine-tune a pre-trained GAN generator model G away from the original distribution and towards a new local minimum defined by the loss function L.[2] L is defined as the weighted sum of the two auxiliary models $L = \alpha C_{classifier}(G(x)) + \beta D_{frozen}(G(x))$ given the random latent vector x, and α and β being the hyper-parameters defining the weightings for the two components of the loss function.

The StyleGAN-NADA framework [Gal, 2021] takes advantage of the external knowledge of a contrastive language–image pre-training model (CLIP) [Radford et al., 2021]. CLIP has been trained on billions of text and image pairs from the internet and provides a joint-embedding space of both images and text, allowing for similarity estimation of images and text prompts. In StyleGAN-NADA, pretrained StyleGAN2 models [Karras et al., 2020] can be fine-tuned using user-specified text prompts, the CLIP model C_{clip} is then used to encode the text prompts and the generated samples in order to provide a loss function where the cosine similarity S between the clip encodings of the text string t and the generated image embedding $G(x)$ given random latent x, can be minimised using the loss $L = S(C_{clip}(t), C_{clip}(G(x))$. This training procedure guides the

[2] Whilst this work was done during the course of my PhD research, it has been left out of the final writeup of this thesis for two reasons: the work was not peer-reviewed in a published conference (only being disseminated as an arxiv pre-print), and the work is not strictly data-free, the common thread between the three experimental contributions presented in Chapters 3, 4 & 5.

generator towards infusing characteristics from an unseen domain defined by the user as text prompts.

6.3.4.5 Reinforcement Learning from Human Feedback

Reinforcement learning from human feedback (RLHF) is an approach to fine-tuning models not using new training data, but by using human feedback to both label and correct the outputs of generative models [Ziegler et al., 2019]. This approach is most commonly used in the alignment of Large Language Models (LLMs) (§6.5.1), where unwanted outputs derived from the original training data are corrected, and the outputs of the generative neural networks are more closely 'aligned' with specified human values, which is a common issue when models are trained on mass-scraped data from the internet which contains many unwanted outputs, such as racist, misogynistic or things that could be considered dangerous. Whilst RLHF is a means by which models are fine-tuned to diverge from their original training data distributions, it is still primarily an imitation-based form of learning. Whilst it could be used to fine-tune models in a truly divergent fashion (discussed in the future research directions §6.7.5), for now, it is primarily used as a form of imitation learning to correct and align the outputs generative models.

6.3.5 Chaining models

An approach that is widely used by artists who incorporate generative models into their practice, but not well documented in academic literature, is the practice of chaining multiple custom models trained on datasets curated by the artists. The ensembles used will often utilise standard unconditional generative models, such as GANs, in combination with other conditional generative models such as image-to-image translation networks, such as pix2pix [Isola et al., 2017] and CycleGAN [Zhu et al., 2017], along with other approaches for altering the aesthetic outcomes of results such as style transfer [Gatys et al., 2016]. Artists will often train many models on small custom datasets and test out many combinations of different models, with the aim of finding a configuration that produces unique and expressive results. The artist Helena Sarin will often chain multiple CycleGAN models into one ensemble, and will reuse training data during inference, as the goal of this practice 'is not generalization, my goal is to create appealing art' [Sarin, 2018]. The artist Derrick Schultz draws parallels between the practice of chaining models and Robin Sloan's concept of 'flip-flopping' [Schultz, 2021], where creative outcomes can be achieved by 'pushing a work of art or craft from the physical world to the digital world and back, often more than once' [Sloan, 2012]. For Schultz, shifting data from one generative distribution to another is the way that novel and creative outcomes can be achieved. An example of Schultz's work, where network bending is used as part of chaining models, is detailed in Section 7.4.6.

6.3.6 Network Bending

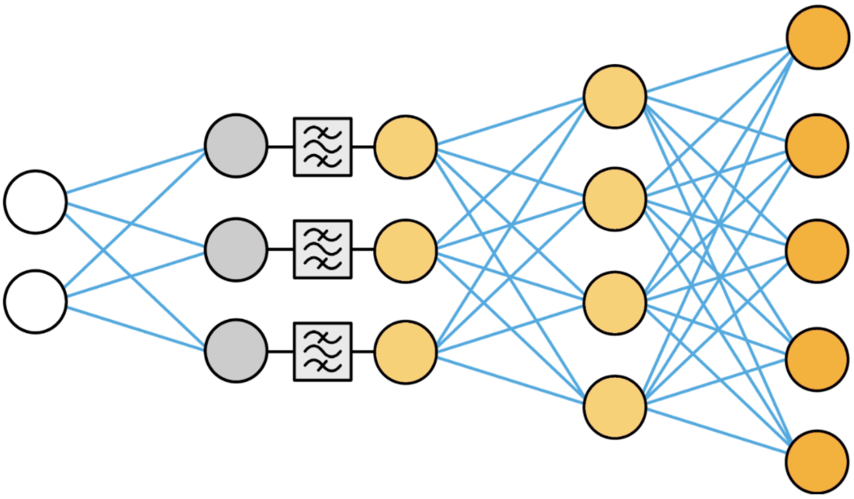


Figure 6.9: Diagram illustrating the network parameter view of network bending. A network pre-trained on the distribution P that produces the approximate distribution P' has additional deterministic transform layers inserted into it which when activated are used to produce the novel distribution U.

Network bending [Broad et al., 2021b, 2022] (presented in Ch. 5) is a framework that allows for active divergence using individual pre-trained models without making any changes to the weights or topology of the model. Instead, additional layers that implement standard image filters are inserted into the computational graph of a model and applied during inference to the activation maps of the convolutional features[3]. As the computational graph of the model has been altered, the model which previously generated samples from the approximate distribution P', now produces novel samples from the new distribution U, without any changes being made to the parameters of the model. In the simplest case of a two-layer model an association weight matrix W_l and bias b_l vector for each layer l. Which generates sample $p' = \sigma(W_2(\sigma(W_1 x + b_1)) + b_2)$ from input vector x and using a non-linear activation function. In the network bending framework, a deterministic function f (controlled by the parameter y) is inserted into the computational graph of the model and applied to the internal activations of the model $u = \sigma(W_2(f(\sigma(W_1 x + b_1), y)) + b_2)$, allowing the model to produce new samples u from the new distribution $u \in U$. Beyond the simplest case of a transformation being applied to all features in a layer, the transformation layer can also be applied to a random subset of features, or to a pre-selected set of features (§5.4.3).

Network bending has been further developed by others into applications in other forms of generative models and domains, as well as building user interfaces for network bending. A full account of the technical impact of the network bending work is given in Section 7.5.

6.3.7 Network Blending

Blending multiple models trained on different datasets allows for more control over the combination of learned features from different domains. This can either be done by blending the predictions of the models, or by blending the parameters of the models themselves.

[3] Inserting filters into GANs was also developed independently in the Matlab StyleGAN playground [Pinkney, 2020b] and in a blog post entitled *GAN Bending* [Pouliot, 2020]

6.3.7.1 Blending Model Predictions

Akten and Grierson [2016] present an interactive tool for text generation allowing for the real-time blending of the predicted outputs of an ensemble of Long-Short Term Memory network (LSTM) models [Hochreiter and Schmidhuber, 1997] trained to perform next character prediction from different text sources. A graphical user interface allows the user to dynamically shift the mixture weights for the weighted sum for the predictions of all of the models in the ensemble, prior to the one hot vector encoding which is used to determine the final predicted character value.

6.3.7.2 Blending Model Parameters

A number of approaches, all demonstrated with StyleGAN2 [Karras et al., 2020], take advantage of the large number of pre-trained models that have been shared on the internet [Pinkney, 2020a]. Of these, almost all have been transfer-learned from the official model weights trained on the Flickr-Faces High Quality (FFHQ) dataset. It has been shown that the parameters of models transfer-learned $p_{transfer}$ from the same original source p_{base} share commonalities in the way their weights are structured. This makes it possible to meaningfully interpolate between the parameters of the models directly [Aydao, 2020]. By using an interpolation weighting α, it is possible to control the interpolation for the creation of a set of parameters $p_{interp} = (1 - \alpha)p_{base} + \alpha p_{transfer}$.

Layers can also be swapped from one model to another [Pinkney and Adler, 2020], allowing the combination of higher-level features of one model with lower-level features of another. This layer-swapping technique was used to make the popular 'toonification' method, which can be used to find the corresponding sample to a real photograph of a person in a Disney-Pixar-esque 'toonified' model, simply by sampling from the same latent vector that has been found as the closest match to the person in FFHQ latent space [Abdal et al., 2019]. A generalised approach that combines both weight interpolation and layer-swapping methods for multiple models, uses a cascade of different weightings of interpo-

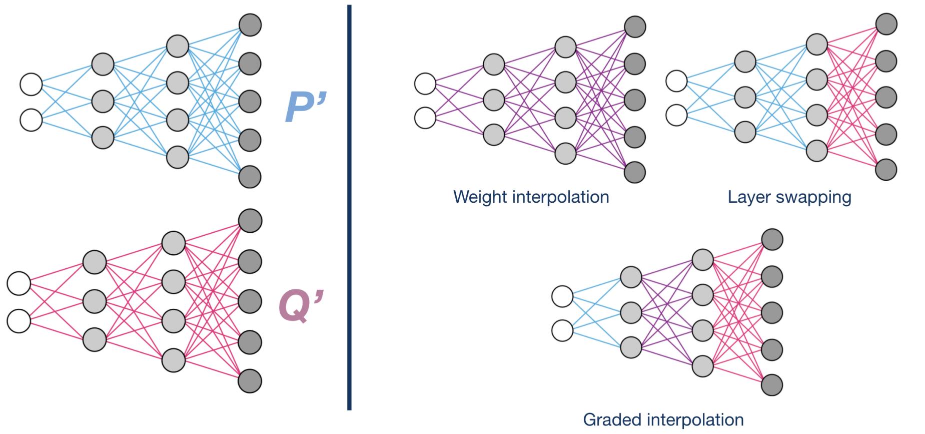


Figure 6.10: Diagram illustrating the network parameter view of network blending. Two networks, pre-trained on the distributions P and P that then produce the approximate distributions P' and Q' can have their parameters blended by either interpolating on the weights, swapping the layers between models or performing graded interpolation across the model hierarchy.

lation for the various layers of the model [Arfafax, 2020].

Going beyond StyleGAN, Colton [2021] presents an evolutionary approach for exploring and finding effective and customisable neural style transfer blends. Upwards of 1000 neural style transfer models trained on 1-10 style images each can be blended through model interpolation, using an interface that is controlled by the user. MAP-Elites [Mouret and Clune, 2015] in combination with a fitness function calculated using the output from a ResNet model [He et al., 2016] were used in evolutionary searches for optimal neural style transfer blends.

6.3.7.3 Stitching Model Parameters

In contrast to model-wide or layer-wide blending of two networks, more sophisticated approaches will use algorithms to perform more complex stitching and blending of parameters and features from different models. The CombiNets framework [Guzdial and Riedl, 2018a], informed by prior research in combinational creativity [Boden, 2004], can be utilised to create a new model by combining parameters from a number of pre-trained models in a targeted fashion.[4] The parameters of existing models are recombined to take into account a new mode of generation that was not present in the training data (an example given would be a unicorn for a model trained on photographs of non-mythical beings). In this framework, a small number of new samples is provided (not enough to train a model directly) and then heuristic search is used to recombine parameters from existing models to account for this new mode of generation.

More recent approaches to blending and stitching networks fall under the label *model fusion* [Li et al., 2023]). Approaches have been developed in the contexts of LLMs and reverse diffusion models, with algorithms like DARE (Drop delta & REscale) [Yu et al., 2024] and TIES-Merging (Trim, Elect, Sign & Merge (TIES-Merging)) becoming the state of the art approaches for network blending.

6.3.8 Model Rewriting

Model rewriting encompasses approaches where either the weights or network topology are altered in a targeted way, through manual intervention or by using some form of heuristic-based optimisation algorithm.

6.3.8.1 Stochastic Rewriting

To create the series of artworks *Neural Glitch* the artist Mario Klingemann randomly altered, deleted or exchanged the trained weights of pre-trained GANs [Klingemann, 2018]. In a similar fashion, the convolutional layer reconnection technique [Růžička, 2020] randomly swaps convolutional features within layers of pre-trained GANs. This technique is applied in the *Remixing AIs* audiovisual synthesis framework [Collins et al., 2020].

[4] In the original published survey paper [Broad et al., 2021a] this was categorised under model rewriting (§6.3.8) in the taxonomy, but has been moved into network blending for this thesis writeup, as on reflection this work better fits this latter category.

6.3.8.2 Targeted Rewriting

Bau et al. [2020] presents a targeted approach to model rewriting. Here, a sample is taken from the model and manipulated using standard image editing techniques (referred to as a 'copy-paste' interface). Once the sample has been altered corresponding to the desired goal (such as removing watermarks from the image or getting horses to wear hats), a process of constrained optimisation is performed. All of the layers but one are frozen, and the weights of that layer are updated using gradient descent optimisation until the generated sample matches the new target. After this optimisation process is complete, the weights of the model are modified such that the targeted change becomes present in all the samples that the model generates.

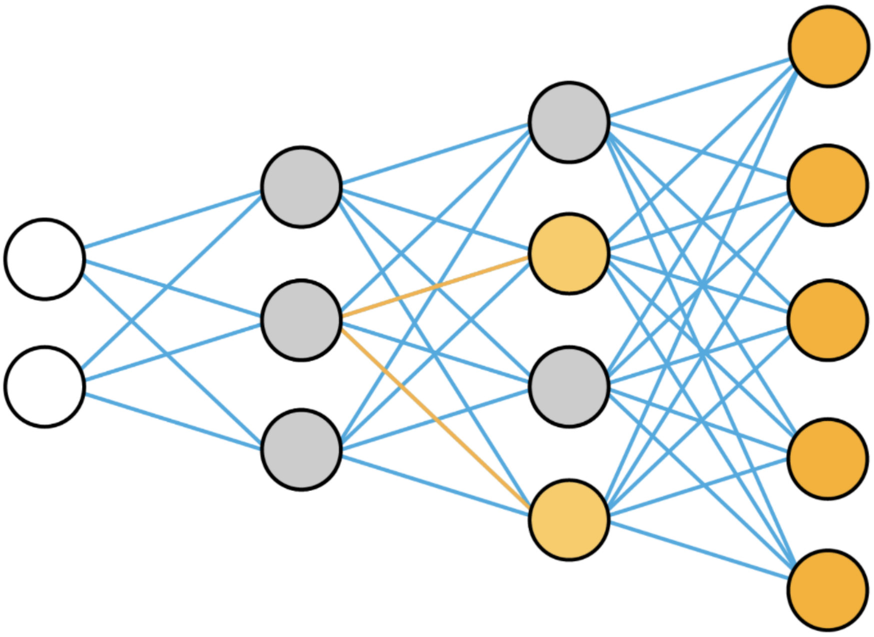


Figure 6.11: Diagram illustrating the network parameter view of model rewriting. A network pre-trained on the distribution P that produces the approximate distribution P' has selected changes made to a small number of parameters resulting in the new distribution U.

In recent years *machine unlearning* [Bourtoule et al., 2021], the goal of removing learned information or representations from trained machine learning models, has become a topic of active research. This usually falling under the umbrella of research into *alignment* (§6.5.1). The work presented by Bau et al. [2020] could be considered an approach to machine unlearning (before the term was coined), though was demonstrated as applicable to many tasks beyond simply erasing concepts. In subsequent years, many approaches to machine unlearning have been developed for generative models generative models [Liu et al., 2024], including notable examples for diffusion models [Gandikota et al., 2023] and LLMs [Zhao et al., 2023].

6.4 Further Demarcations of Active Divergence Methods

6.4.1 Training from Scratch vs. Using Pretrained Models

Finding stable, effective ways of training generative models, in particular GANs, is difficult and, depending on the training scheme, there are only a handful of methods that have been found to work successfully. Few methods for active divergence train a model completely from scratch. Instead, most take pre-trained models as their starting point for interventions. This way, training from scratch can be avoided, but fine-tuning may still be required.

6.4.2 Utilising Data vs. Data-Free Approaches

Most of the approaches described utilise data in some way, whether as an inspiring set for novelty generation or for combining features from different datasets (divergent fine-tuning, network blending and chaining models). Even methods for model rewriting use very small amounts of example data to guide optimisation algorithms that alter the model weights. However, methods like network bending show how models can be analysed in ways that don't rely on any data, and are used for intelligent manipulation of the models — an approach which could be applied to other methods like model rewriting. Methods that train and fine-tune models without data also show how auxiliary networks and the dynamics between models can be utilised for achieving active divergence. The common thread of the novel technical contributions in this thesis (presented in Chs. 3, 4 & 5) is that they are all data-free approaches to active divergence.

6.4.3 Human Direction vs. Creative Autonomy

Very few of the approaches described have been developed with the expressed intention of handing over creative agency to the systems themselves. Most of the methods have been developed by artists or researchers in order to allow people to manipulate, experiment with and explore the unintended uses of these models for creative expression. However, the methods described that are currently designed for, or rely on a high degree of human curation and intervention, could easily be adapted and used in co-creative or autonomous creative systems in the future [Berns et al., 2021].

6.5 Related Research Areas

Since the publication of the original survey paper [Broad et al., 2021a], there have been two areas of related research in generative neural networks that have emerged to be significant sub-fields of AI research, that are considered related fields or even subfields of active divergence research.

6.5.1 Alignment

AI alignment first emerged from the philosophical study of AI, which focuses on AI safety, and the goal of ensuring that the behaviour of AI 'aligns' with human values [Yudkowsky, 2016, Gabriel, 2020]. The first attempts at codifying rules for AI agents were made much earlier than the current discourse around AI alignment, and examples from science fiction literature such as Asimov's Three Laws of Robotics [Asimov, 1942] can be considered constituting the first proposed ways that the behaviour of AI is governed, or aligned with human values.

How ‘human-values’ are defined is not always universally agreed upon [Turchin, 2019]. Often, when done by large organisations, alignment is driven as much by legal compliance, liability, and the avoidance of public relations controversies, as it is defined by a universally agreed-upon set of human values.

Ji et al. [2023] give a comprehensive survey of methods for achieving AI alignment, which is most commonly undertaken in the space of LLMs based on transformer architectures [Vaswani et al., 2017]. The most commonly used method for alignment is Reinforcement Learning from Human Feedback (RLHF) [Ziegler et al., 2019], which is used by many companies now that provide LLMs as API services. RLHF could be considered a form of *divergent fine-tuning* (§6.3.4), though this is often done to imitate human preferences and to prevent LLMs from repeating racist, misogynistic or generating other dangerous outputs that would have been learnt from original training data (usually on text mass-scraped from the world wide web). So while this is diverging from the original training data, this is usually still done as a form of imitation-based learning in the fine-tuning stage, and not applied in creative contexts.

Notably, there are two other emerging research areas of techniques used in AI alignment: *machine unlearning* and *model fusion*. Both of these areas of alignment research are preceded by experiments from creative practitioners and researchers exploring the creative potential of generative neural networks, as is shown in Section 6.3.8 (model rewriting) and Section 6.3.8 (network blending) of this survey.

6.5.2 Data Poisoning

Data poisoning, first performed by Biggio et al. [2012], is the task of injecting data into training datasets that interfere with the training and optimisation process for malicious effect [Fan et al., 2022]. Data poisoning has become popular in the generative space with widely used algorithms such as *glaze* [Shan et al., 2023] *nightshade* [Shan et al., 2024], which are designed to protect the intellectual property of artists against their work being used without consent in training text-to-image diffusion models [Rombach et al., 2022]. In this context, data poisoning for generative models can be seen as an approach to active divergence where the data is deliberately corrupted in order to prevent the faithful imitation of the training data and disrupt training such that the generative model no longer accurately models the training data, but instead, diverges from it in unwanted ways (from the perspective of the actors who are training the generative model).

6.6 Applications of Active Divergence

In this section, I outline some of the applications for active divergence methods outside of the two related areas of research detailed in the previous section (§6.5).

6.6.1 Novelty generation

Generative deep learning techniques are capable of generalisation, such that they can produce new artefacts of high typicality and value, but are rarely capable of producing novel outputs that do not resemble the training data. Active divergence techniques play an important role in getting generative deep learning systems to generate truly novel artefacts, especially when there may be limited or even no data to draw from.

6.6.2 Creativity Support and Co-Creation

Some of the frameworks presented are already explicitly designed as creativity support tools, such as the network bending framework, designed to allow for expressive manipulation of deep generative models. The *Style Done Quick* [Colton, 2021] application where many style transfer models have been evolved, was built as a casual creator application [Compton and Mateas, 2015]. Though many of the other methods described are still preliminary artistic and research experiments, there is a lot of potential for these methods to become better understood and eventually adapted and applied in more easily accessible creativity support tools and co-creation frameworks.

6.6.3 Knowledge Recombination

Reusing and recombining knowledge in efficient ways is an important use-case of active divergence methods. While impressive generalisation can be ascertained from extremely large models trained on corpora extracted from large portions of the internet [Ramesh et al., 2021], this is outside of the capabilities of all but a handful of large tech companies. Instead of relying on ever-expanding computational resources, active divergence methods allow for the recombination of styles, aesthetic characteristics and higher-level concepts in a much more efficient fashion. Methods like chaining models, network blending and model rewriting offer alternatives routes to achieving flexible knowledge recombination and generalisation to unseen domains without the need for extremely large models or data sources.

6.6.4 Unseen Domain Adaptation

Active divergence methods allow for the possibility of adapting to and exploring unseen domains, for which there is little to no data available. The network blending approach presented by Pinkney and Adler [2020] can be used for the translation of faces while maintaining a recognisable identity into a completely synthesised data domain, something which would not be possible with standard techniques for image translation [Zhu et al., 2017].

The model rewriting and network bending approaches offer the possibility of reusing and manipulating existing knowledge in a controlled fashion to create new data from a small number of given examples, or theoretically without any prior examples if external knowledge sources are integrated, as discussed further below. This approach could also be utilised by agents looking to explore hypothetical situations, by reorganising learned knowledge from world models

[Ha and Schmidhuber, 2018] to explore hypothetical situations or relations.

6.7 Future Research Directions

In this section, I discuss possible future research directions and applications for developing, evaluating and utilising methods for active divergence.

6.7.1 Metrics for Quantitative Evaluation

For the advancement of research on active divergence, methods for quantitative evaluation will be critical in order to keep track of progress, to compare techniques and for benchmarking. Metrics for active divergence will have to go beyond measuring the similarity or dissimilarity between distributions, as is usually done in the evaluation of generative models [Gretton et al., 2019]. Active divergence metrics should contribute to a better understanding of *how* the distributions diverge. Therefore, various changes to the modelled distribution should be taken into consideration when looking to measure divergence between distributions in creative contexts. These include increases or decreases in diversity, the consistency and concurrency of change across the whole distribution and whether changes primarily affect low or high-level features.

6.7.2 Automating Qualitative Evaluation

In addition to quantitative evaluation, other metrics are needed for evaluating active divergence metrics. These could ideally rely less on qualitative evaluation for guiding decisions in creating new models, and do this in a computational fashion so that these aspects of the process could be automated. For instance, a recently developed metric for measuring visual indeterminacy [Wang et al., 2020b], which is argued as being one of the key drivers for what people find interesting in GAN-generated art [Hertzmann, 2020], could be used for replacing the qualitative evaluation and curation step completed by humans. Other metrics that could be used are novelty metrics [Grace and Maher, 2019], bayesian surprise [Itti and Baldi, 2009], aesthetic evaluation [Galanter, 2012], or measurements for optimal blends between data domains and evaluating the novelty of changes made to semantic relationships.

6.7.3 Multi-Agent Systems

It has been argued that the GAN framework is the simplest example of a multi-agent system [Agüera y Arcas, 2019], and frameworks such as neural cellular automata [Mordvintsev et al., 2020] offer new possibilities for multi-agent approaches in generative deep learning. The active divergence methods for training without data described in this paper all rely on the dynamics of multiple agents to produce interesting results, but this could be taken much further. It has been argued that art is fundamentally social [Hertzmann, 2021] and exploring more complex social dynamics between agents [Saunders, 2019] could be a fruitful avenue for exploration in the development of these approaches. There is a large body of work in emergent languages from cooperative multi-agent systems [Lazaridou et al., 2017] that could be drawn from in furthering the work in generative multi-agent systems.

6.7.4 Open-Ended Reinforcement Learning

Open-ended reinforcement learning, where there is no set goal [Wang et al., 2020a], offers possibilities for new more autonomous approaches to achieving active divergence. Reinforcement learning has not been discussed in this survey but has been used in generative settings [Luo, 2020] in nascent research. Reinforcement learning approaches offer many opportunities for frameworks of creativity to be explored that are not available to standard generative deep learning methods, as they take actions in response to their environment, rather than just fitting functions. Paradigms like intrinsic motivation [Shaker, 2016], cooperating or competing with other agents, and formulating and acting on intentions are all concepts that conventional generative deep learning systems alone cannot explore, but these paradigms could be investigated in open-ended systems utilising reinforcement learning.

6.7.5 Divergent RLHF

Reinforcement Learning from Human Feedback (RLHF) is most commonly used as a form of imitation-based learning in order to correct and align the outputs of large models, such as LLMs and text-to-image models. However, this is not the only possible use for this. For instance, RLHF could be used for more personalised divergent fine-tuning of models, to actively fine-tune towards novel data distributions, and more closely align to an individual's aesthetic, political, social, or cultural preferences. Using RL to fine-tune generative neural networks could also be used in conjunction with open-ended RL (§6.7.4) to fine-tune models in truly novel and divergent directions.

6.8 Conclusion

This chapter has presented a comprehensive survey of active divergence methods, as well as related areas of research and potential future research directions for active divergence research. Active divergence has been the primary goal of all of the research in this thesis, even if it was done primarily before the term active divergence was conceived in 2020. The research in this thesis constitutes three categorical contributions to active divergence methods: *training without data* (Ch. 3; §6.3.3), *divergent fine-tuning* (Ch. 4; §6.3.4), and *network bending* (Ch. 5; §6.3.6). All of the methods I have presented in this thesis as original research contributions are data-free (§6.4.2), which makes this a largely unique and novel approach to achieving active divergence. These contributions, along with the formal technical delineation and survey of methods presented in this chapter each comprise distinct contributions of research in this thesis. The following chapter (Ch. 7) details the artistic and technical impact of my work outside of the formal active divergence perspective presented in this chapter.

Chapter 7

Impact

7.1 Introduction

This chapter details the impact such as the work described in this thesis has had, including outcomes like artworks that were made by myself and others; and recognition, such as exhibitions and awards. This also includes an overview of work and research that follows and has been influenced by this research, applications of the research into other domains, and examples of ideas from this thesis being used and put into technologies and practical interfaces.

7.2 *(un)stable equilibrium*

As a direct outcome of the original set of experiments detailed in Chapter 3, a series of six video artworks titled *(un)stable equilibrium 1:1, 1:2, . . . 1:6* (Fig. 3.1) were made by sampling from the paired generative models, in parallel, using the same latent code (Fig. 7.1). A looping (spherical) latent space interpolation [White, 2016] of the two videos was produced, which lasted approximately one hour. The interpolations were deliberately designed to be slow to provide a meditative loop seamlessly so that it could be played in a gallery setting without any interruption.

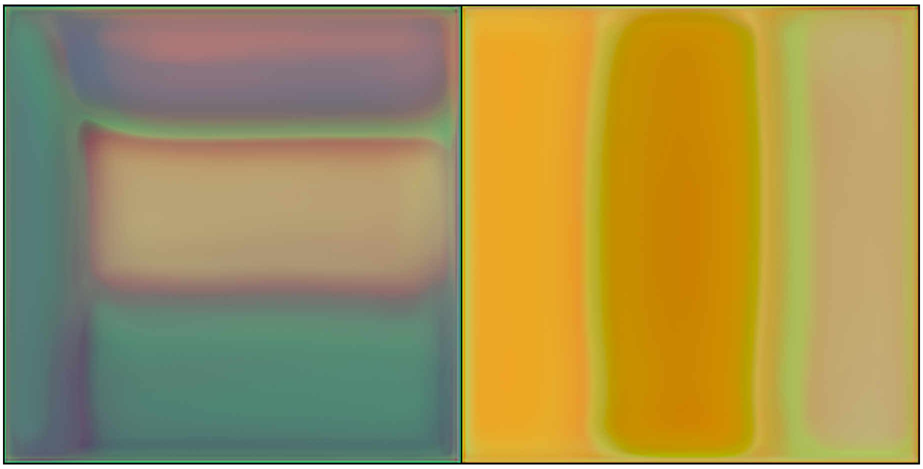


Figure 7.1: Still from *(un)stable equilibrium 1:1*.

The works were first shown in the exhibitions for the respective conferences ICCV (International Conference on Computer Vision) and NeurIPS (Conference on Neural Information Processing Systems) in 2019. At NeurIPS the works were shown in the AI Art Gallery, where the work was also presented as a workshop presentation at the NeurIPS Workshop for Creativity and Design. At ICCV the work was shown in the Computer Vision Art Gallery, where it won the Grand Prize in Computer Vision Art, an honour that is only shared between myself [Broad, 2019], Anna Ridler [2016] and Nouf Aljowaysir [2021].

The work was shown in a gallery setting in March 2020 in Geneva, Switzerland at One Gee in Fog, though unfortunately, that exhibition had to be cut short after 2 days because of the imposition of the COVID lockdown in Switzerland. In lockdown, I began producing prints of the works onto metal aluminium plates, where the glossy finish was a good match for the highly saturated colours in many of the prints. Initially, I was selling these prints online through my own website. Physical works in this series were later exhibited and sold in the commercial London gallery *the depot_*, in their debut show titled *the depot_ digs* [depot_, 2021] (Fig. 7.2), where I was also invited to give an artists talk in 2021.

Following the COVID-19 pandemic, the original video work *(un)stable equilibrium 1:1* was shown in the exhibition *SUPERCREATIVITY* at the Fiesp

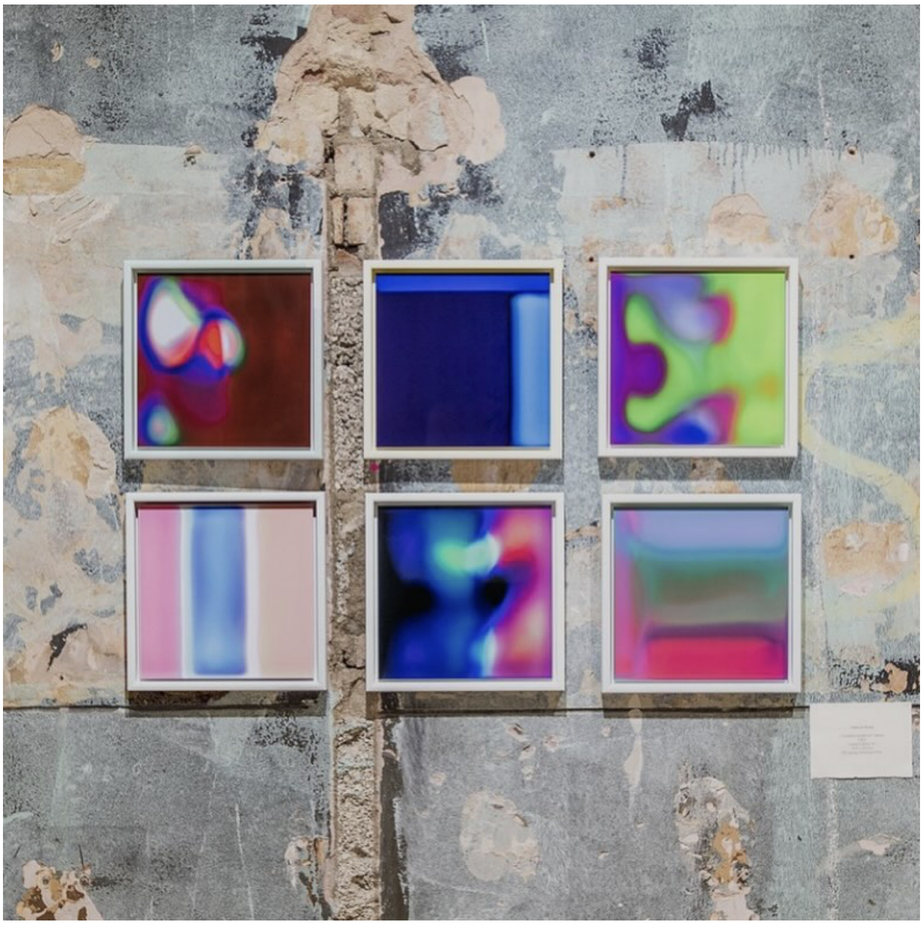


Figure 7.2: Installation view of prints from the *(un)stable equilibrium* series at *the depot_ digs* (the depot_, London, August 12th to August 29th, 2021). Image courtesy of the depot_.

Cultural Centre in São Paolo as part of FILE Festival 2022. FILE (Electronic Language International Festival) is the premier digital arts festival in South America. Here, the work was presented in its originally intended form as a lopping video piece in a gallery installation setting (Fig. 7.3).

7.3 Divergent fine-tuning

Directly from the latter set of experiments described in Chapter 4, inverting the objective function, the series of artworks *Being Foiled* (Fig. 7.4), were produced using the model checkpoints after 500 iterations from the 512x512 StyleGAN

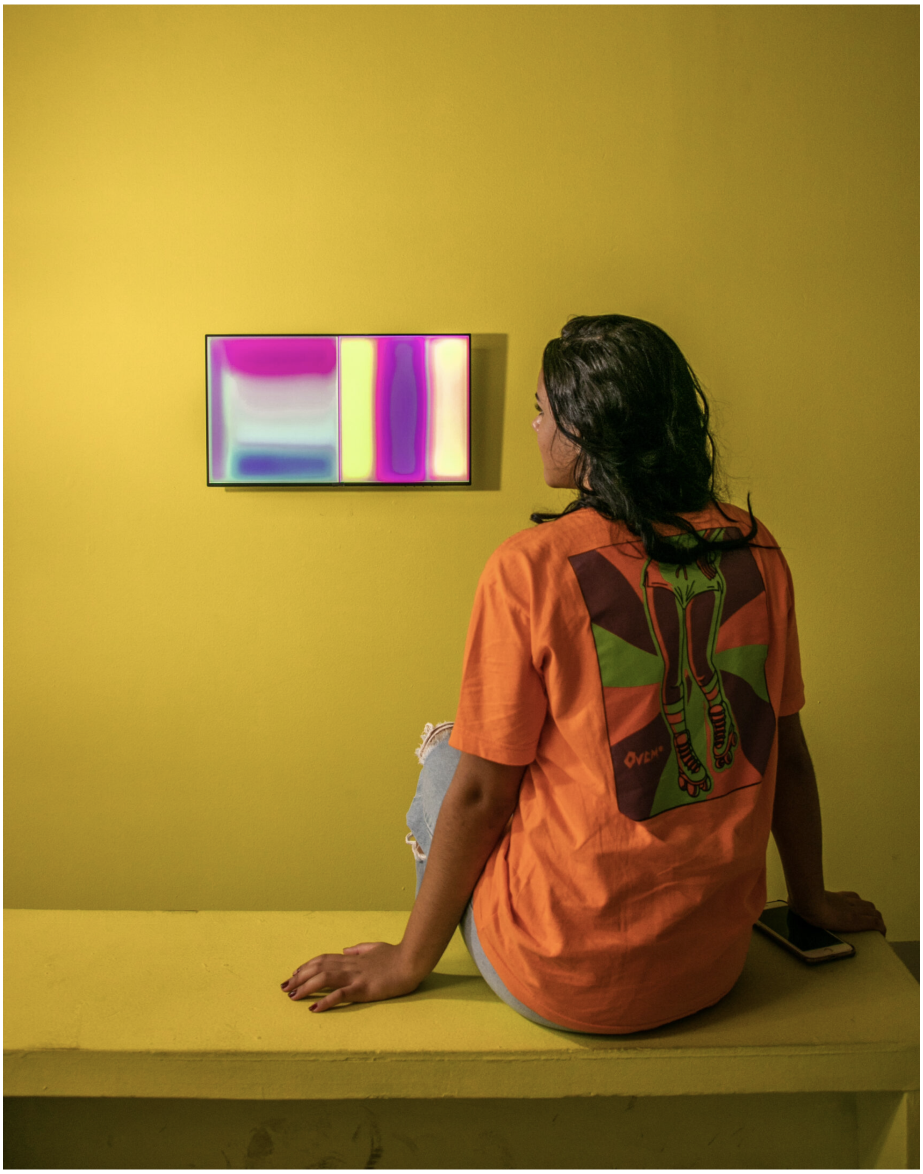


Figure 7.3: Installation view of *(un)stable equilibrium 1:1* at *FILE 2022 São Paolo - SUPERCREATIVITY* (Centro Cultural Fiesp, São Paolo, July 13th to August 28th, 2022). Photograph by Camila Picolo. Image courtesy of FILE - Electronic Language International Festival.

FFHQ model.

The paper *'Amplifying the Uncanny'*, which described the second set of experiments in Chapter 4, after being published in xCoAx was cited by Berns and Colton [2020] in their paper *'Bridging generative deep learning and com-*

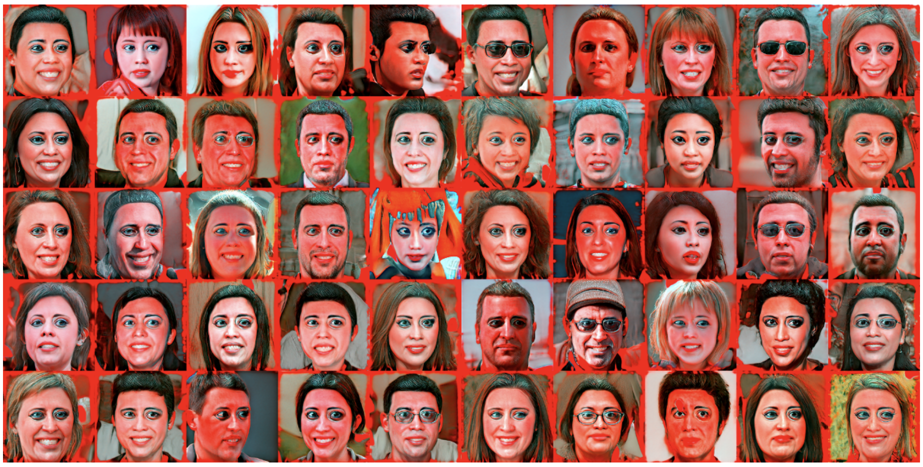


Figure 7.4: *Being Foiled* (2020).

putational creativity'. It was in this paper that they coined the term active divergence, in an original taxonomy with four categories: *latent space search*, *cross-domain training*, *early stopping and rollbacks* and *loss hacking*. Where the last category, loss hacking describes the work described in Chapter 4. This paper inspired the expanded survey and taxonomy of active divergence methods that I wrote in collaboration with Sebastian Berns and Simon Colton [Broad et al., 2021a], detailed in the previous chapter.

The idea of freezing the weights of the discriminator and using them for fine-tuning, was used in both experiments in Chapter 4, and was used independently in the *freezing the discriminator* method [Mo et al., 2020]. In this work, only the lower layers of the discriminator model were frozen, which was then used to aid and assist in the fine-tuning step. Further investigations of the representations of the frozen discriminator network after training was performed by Porres [2021]. This work uses the gradient methods popularised in the *deepdream* algorithm to visualise internal feature activations of the discriminator network.

The experiments described in Chapter 4 were the first published descriptions of methods for performing divergent fine-tuning without relying on imitation-based learning. Subsequent approaches are detailed in Section 6.3.4.

7.4 Artworks made with network bending

A number of artworks have been made with the network bending framework, by myself, and by others. This section will detail them in a mostly chronological order.

7.4.1 *Teratome*

Early on in the experimental development of the network bending framework, I was hand-coding modifications to the neural network code and seeing the respective changes. These were simple point-wise mathematical operations, like ablation $x * 0$ and inversion $x - 1$ on the feature maps. The most significant effect of these manipulations occurred in the first few layers of the generator. In my initial experiments, I hard-coded these transformations into the model. Initially, I would perform this layer-wide, and later on, I was performing these to a random selection of the feature maps in a single layer.

One of the things that struck me when examining the randomly selected manipulations of feature maps within a layer, was that in 1 in 50 to 100 images, recognisable characteristics would be preserved or altered in ways not seen in the other samples. For instance, eyes, and mouth, would be intact in the generated results. This exploratory stage of work is what led to the intuition that sets of features, rather than the approach of examining individual features taken by the GAN Dissection approach [Bau et al., 2019], would be important to allow for my semantically meaningful control and manipulation of the generated results (§5.4.3).

These early experimental images (developed in 2019) were not publicly disseminated until I had published the first pre-print of the network bending paper. I later revisited them and hand-picked some of the most striking results as a series of artworks named *Teratome* [2020c] (Fig. 7.5). The name was inspired by their resemblance to teratomas, which are tumours that can contain hair, teeth and bone.

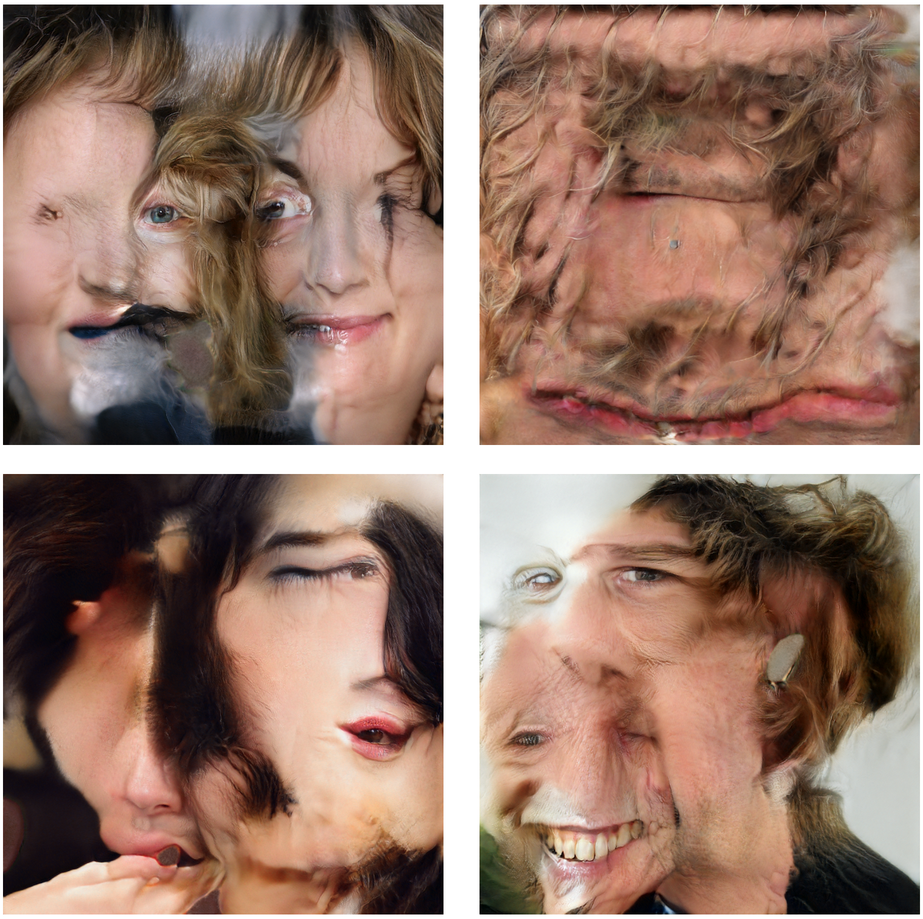
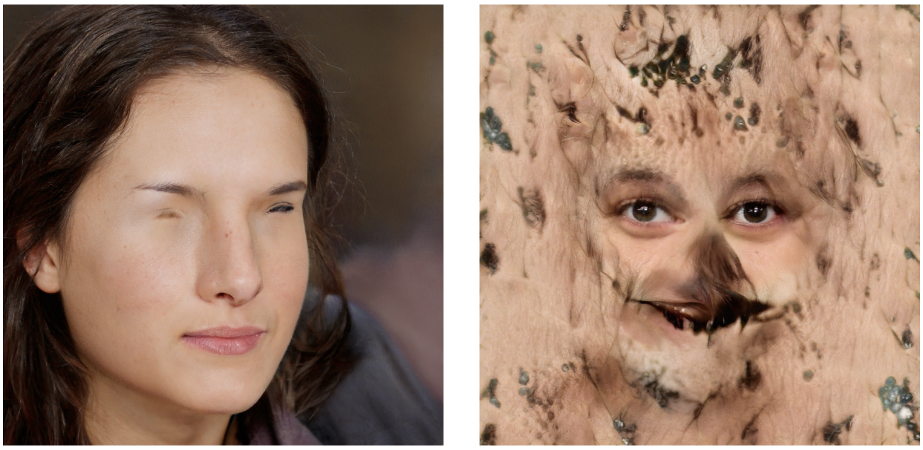


Figure 7.5: *Teratome* (2020).

The works from the series *Teratome* were one of the jurors selected works in the NeurIPS AI Art Gallery in 2020 [Broad, 2020c] and the HCI-Art gallery at the CHI conference in 2022 [Perry et al., 2022]. These were later included in the subsequent book publication *'The State of the (CHI)Art'* [Sturdee et al., 2023].

7.4.2 *Disembodied gaze*

Another artwork that was made during the development of the network bending framework was the work *Disembodied gaze*. This was made shortly after I completed the work on the clustering algorithm and investigated the results. One of the clusters from the algorithm that had the clearest effect was the cluster in layer 5 that determined the generation of eyes (§5.4.3). When the cluster is ablated, the eyes disappear and the model fills in the gaps with skin (Fig. 7.6). This alone was quite a surprising result. But when all the features but the eyes are ablated, things get a lot more surprising.


(a)

(b)

Figure 7.6: (a) Image generated using network bending with cluster controlling the generation of eyes ablated. (b) Image generated with network bending where all convolutional features apart from those that generate eyes are ablated.

I was struck by the bizarre textural regions that were filled in the background, and the ghostly smile that emerges from this absence. I experimented with making a latent interpolation video with this model. GAN latent space interpolations were very popular back in 2016-2020, and I personally was not that keen on them. I found the constant morphing of them quite nauseating and I felt like they were quite a cheap trick to do with any newly trained GAN model, relying on a tendency in AI art that Zylinska describes as ‘dazzling viewers with the mathematical sublime of big data sets, rapid image flows and an intermittent flicker of light, sound and movement’ that ‘ends up serving as a PR [public relations] campaign for corporate interests’ [2020]. However, with the network bending intervention, I did not get the nauseating effect from the constant shape-shifting in the same way. Though the identities were changing, so many of the recognisable characteristics were gone, leaving only the eyes as a fixed point in the video, contrasted with the stochastic nature of the constantly changing textural background.

After making an initial video at the standard square aspect ratio (1024x1024), ubiquitous for latent space interpolation videos at the time, I set about making a work that was bigger and at a more cinematic aspect ratio. I developed a new network bending transformation layer that mirrored and extended the width of the activation maps by padding the sides of them with zeros.[1] As all of these regions that were bordering the edges of the image were already ablated in the network, the effect of this padding appears seamless in the generated result.

After some cropping and formatting of the video into a commonly used aspect ratio, and creating a seamlessly looping video 13 minutes in length, I created the video work *Disembodied gaze* [Broad, 2020b]. This work was never exhibited as such, but I revisit it a lot in artist talks as it is a good illustration of what is possible with network bending, and is a good demonstration of something that is unique to the approach, and would be near impossible to make any other way. The padding layer was also something I used later on in the *Fragments of Self* (§7.4.4).

[1] I did not make this transformation layer publicly available in the open source network bending GitHub repository, as invalid parameters could quickly lead to the software crashing due to a segmentation fault or GPU memory error.

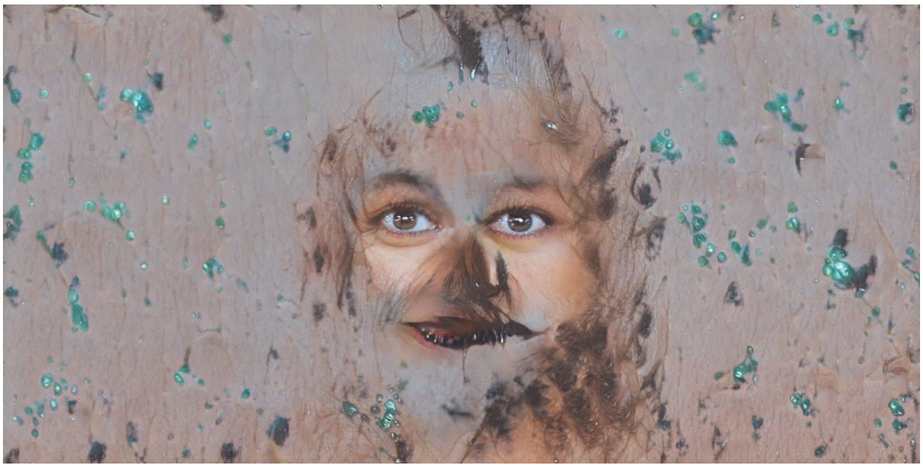


Figure 7.7: Still from *Disembodied Gaze* (2020).

7.4.3 Single and EP Artworks for *0171*

In early 2020, I was approached by musicians from the band 0171, who were releasing a series of singles, followed by an EP that autumn. They were keen to use images of themselves, and have them manipulated using some of the techniques that I had been developing in this PhD. As I was using StyleGAN2, there was already existing code online that performed projection of photographs of people into the GAN latent space [Abdal et al., 2019]. This meant that I could take portrait photographs of the two band members and project them into StyleGAN2 latent space, before then manipulating the models while generating these latent codes to make artwork for the commission.

They provided me with one of their press shots (Fig. 7.8) that was to be used for their forthcoming marketing campaign for the new singles and EP. I cropped the respective faces of the two band members, and projected them into StyleGAN2 latent space, using the gradient method [Abdal et al., 2019] (Fig. 7.9).

I took an exploratory approach to find combinations of stochastic and layer-wide transformations to the models, using this as an exercise to understand how transformations could be combined to produce more divergent and original images. I would keep the latent the same, alternating between the latent codes

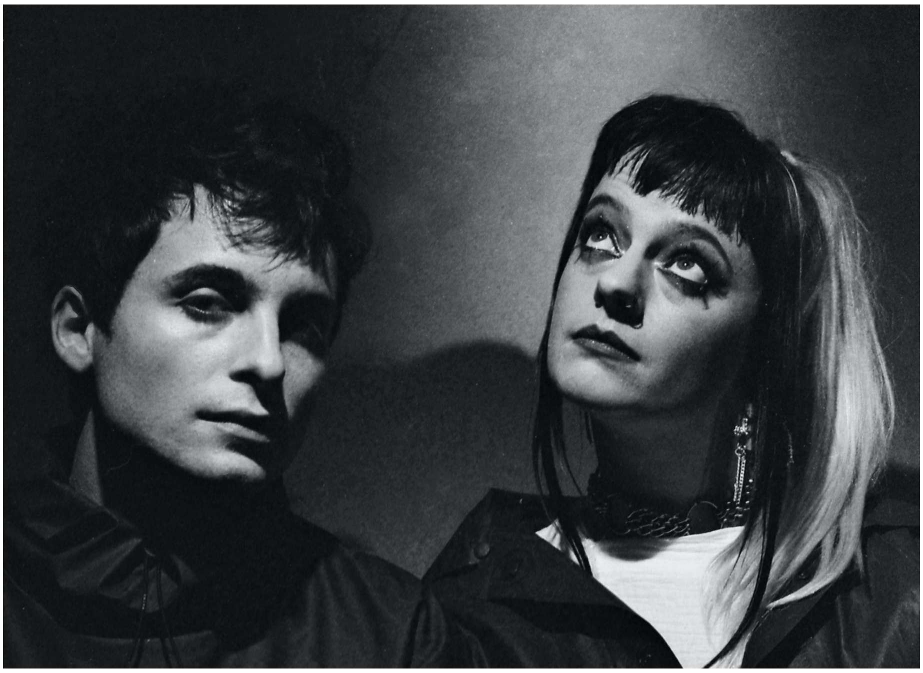
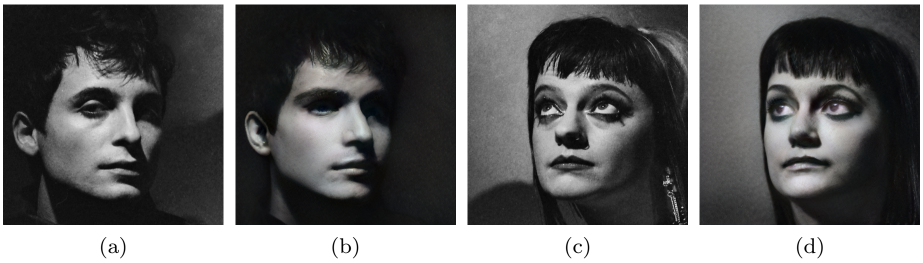


Figure 7.8: 0171 Press shot. Image courtesy of Georgie Hoare and Joe Bedell-Brill.



(a) (b) (c) (d)

Figure 7.9: Images of the individual band members cropped from the press shot (a,c) and their respective StyleGAN2 projections (b,d).

for the two respective band members, testing out different configurations of transformation parameters. I would generate 20 images using one set, and there would always be variation in the images because of the stochastic layer transformations used. There would be more significant variation if these were used in the earlier layers of the GAN, or if the percentage of random features distorted with a transformation was increased.

I would experiment intuitively with different configurations of transformations. If the set of results did not have much interesting variation I would boost the random threshold for features applied, and if it had too much I would tone these down. If one configuration produced a particularly fruitful set of results, I would generate more using the same parameters – i.e. 100 or 1000. After experimenting like this for several days I selected my favourite samples and shared those with the band.

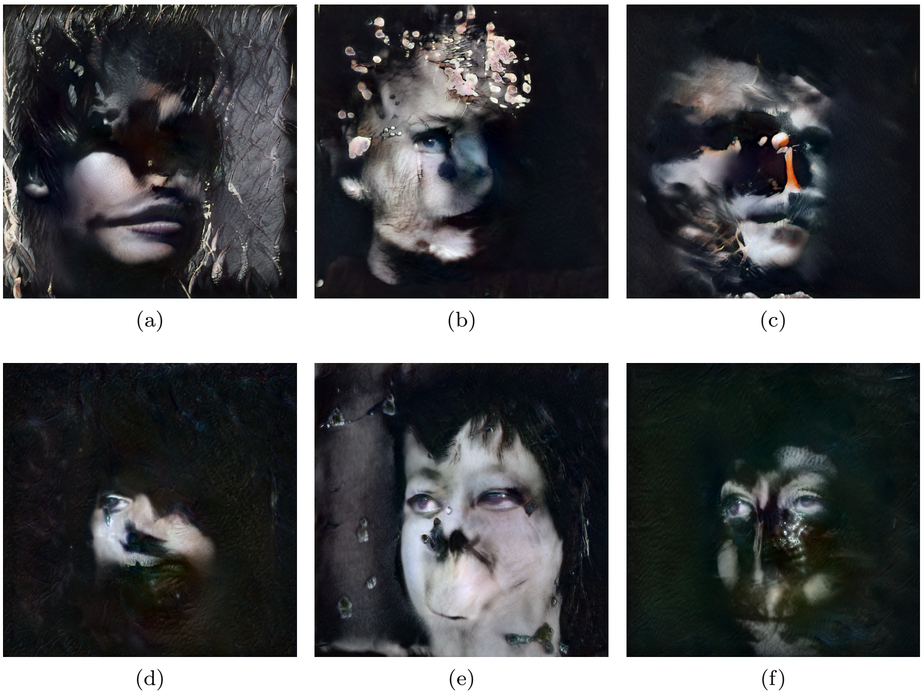
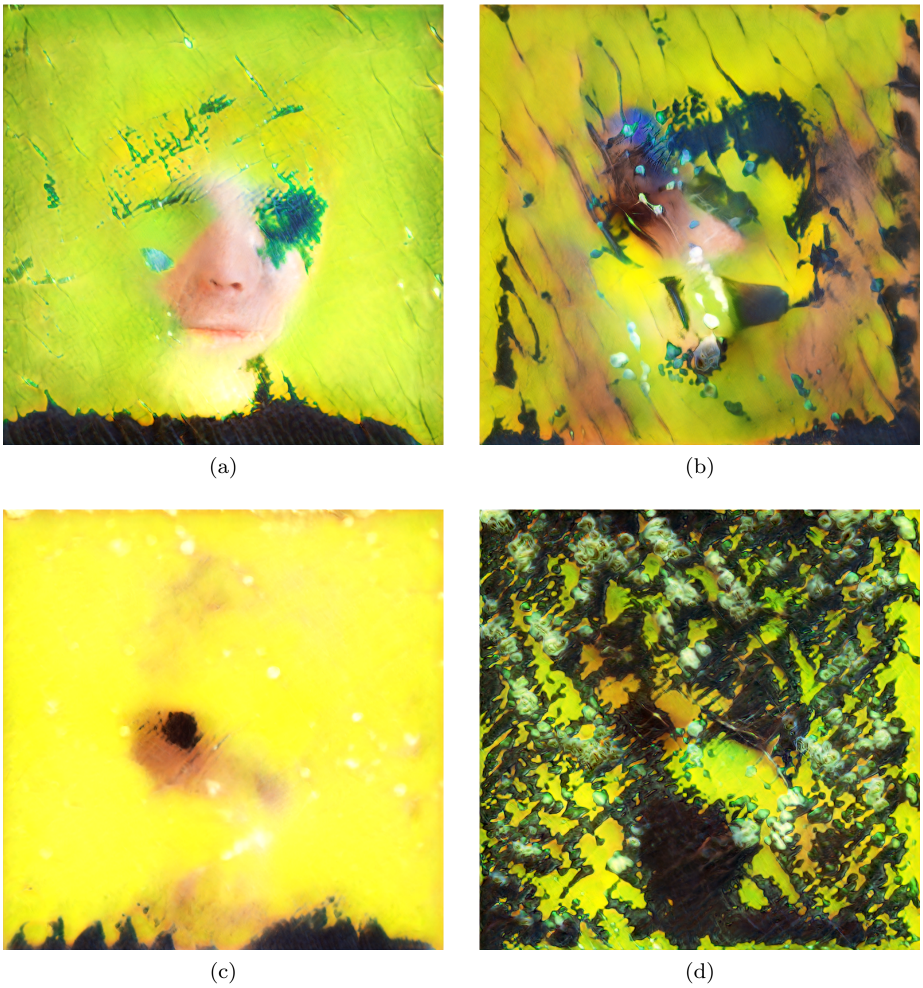


Figure 7.10: (a-c) images generated using network bending techniques applied to the latent 7.9b, (d-f) images generated using network bending techniques applied to the latent 7.9d.

I shared these original works with the band, and while they were very impressed with the result, they were not quite in line with the desired aesthetic for a synth-pop band. The original images were sourced from a dark, black-and-white film photograph, which had been deliberately distorted with scratches and other physical interventions made to the film (Fig. 7.8). As the latent codes were conditioned on this image, the dark, gothic look was persistent in the results and accentuated by the facial distortions present (Fig. 7.10). I advised them that if they wanted something more colourful and pop-friendly, then using a different more colourful image of themselves as the starting point would work better. They provided me with headshots taken in front of a colourful painting, which we then used for the project.

I began repeating the process described earlier with the previous latent codes of the band members. Trying out the same transformation parameters that produced interesting results with the previous latent codes, and adapting them to work better with the new ones. Not all transform configuration settings that worked with the previous latent codes worked well with these, without some tweaking. Showing that there was a clear contingency between how well different latent codes and transform parameter configurations would work well together. In addition, based on feedback from the band members that the distorted but recognisable faces were also not aligned with the appearance the band were trying to give off, I increased the level of distortion so that the generated images appeared more abstract than with the initial attempt (Fig. 7.10). This time, the band members themselves were more involved in the process. I would experiment with transformation parameters, select some of my personal favourites, and show these to the band who would tell me what they liked and disliked and that would inform further experimentation. We ended up with 10 pictures, 5 for each band member, and they selected their favourite of these for the 4 EP and single releases (Fig. 7.11).

Working on this series of artworks as the network bending framework was in development was fortunate in its timing as it served as a useful case study early on in the development of this framework in a real-world application, and later detailed in the original EvoMUSART paper [Broad et al., 2021b] The single and EP artworks are available to see on all good music streaming services. The earlier images (in Figure 7.10) were later used (with permission from the band) to produce the NFT artworks *Haunted Variations* [Broad, 2021b,c].



(a)

(b)

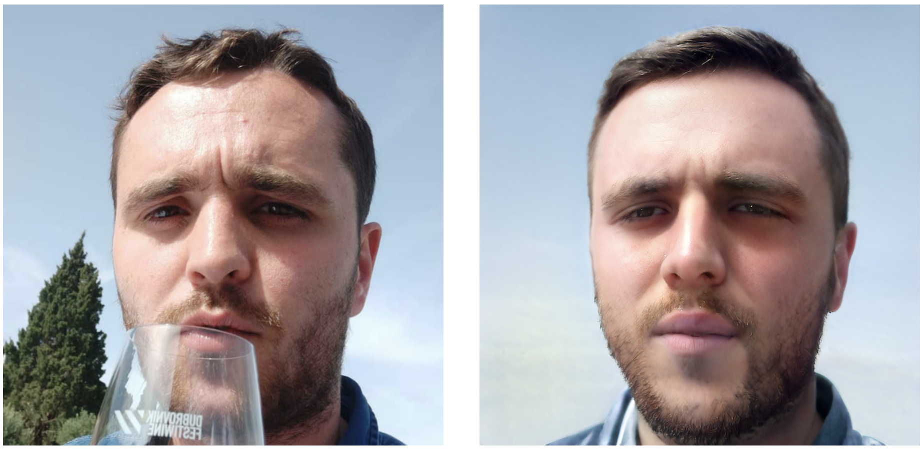
(c)

(d)

Figure 7.11: EP and single artworks created for the band 0171. (a) Artwork for single *Automatic*. (b) Artwork for the single *Follow*. (c) Artwork for the single *Photograph*. (d) Artwork for the EP (extended play) compilation *Change Nothing*.

7.4.4 *Fragments of self*

After producing some striking visuals with the transformations of the band members, I would have been remiss if I were not to have performed the same transformations myself. I took a self-portrait photograph of myself from a holiday in Croatia and projected that into StyleGAN2 latent space.



(a) (b)

Figure 7.12: (a) Original selfie photograph of myself. (b) Closest match to (a) in StyleGAN2 latent space.

I took some of the random transformation configurations that were used for the final 0171 EP artworks and tried them on myself, but these were rendering images that were largely unrecognisable from myself. Therefore, I reduced the number of convolutional filters that were being affected by the random layer filters and reduced some of the other parameters such that the results were more clearly recognisable (Fig. 7.13).

None of these images, by themselves, were particularly close to my own recognition, as the level of distortion was still quite high. As I was navigating through them in the image viewer, I noticed that if I held these keys and scrolled quickly through the images, as if they were in motion, the resemblance to myself was much stronger. I ended up stitching together 1000 of these randomly generated images into a looping video approximately 40 seconds long and making

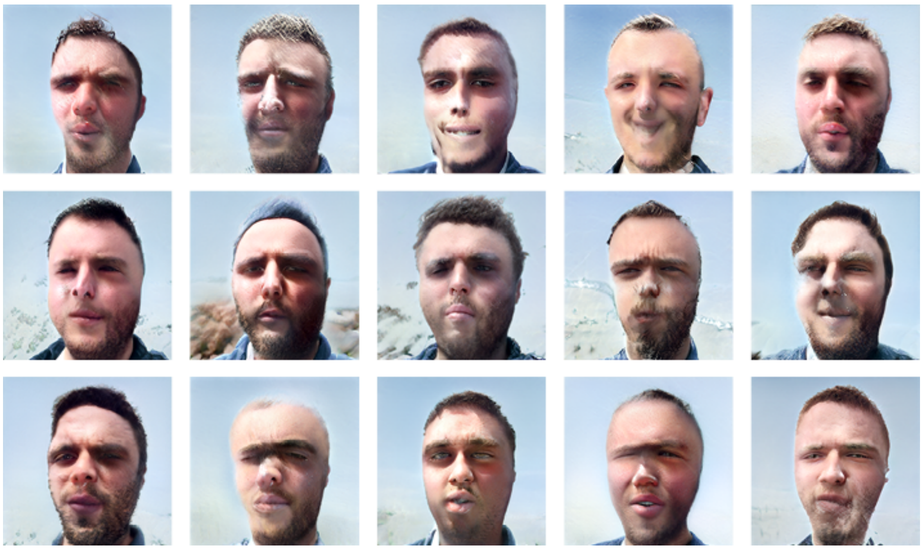


Figure 7.13: Samples of selfies distorted with random network bending parameters based on the latent 7.12b.

a short video piece that was originally shared on social media.[2] Like the work *Disembodied gaze* (§7.4.2), this was not exhibited anywhere but is something I share regularly in artist talks as it is an instructive illustration of the possibilities offered by the network bending framework. It was not until I was invited to participate in the upcoming exhibition for the digital art gallery platform Feral File (created by Casey Raes), that this line of enquiry had any major impact.

I was unsure of what I was going to exhibit in this show, but I was given the curator notes from Luba Elliot several months in advance for the exhibition opening, where she had decided on the title *Reflections in the water*:

> ‘Working with AI art sometimes feels like gazing into a pond of water — we are not sure what we will get as a reflection. [...] Looking into a still pond, we see a clear, gently blurred version of ourselves staring back at us, while turbulent waters return mere rippled echoes of our shape. These changing reflections of ourselves are similar to images generated from data, which can be hyper-realistic depictions of the original, or images that are surreal and barely recognizable, as flaws

[2] This video can be viewed here: `https://www.youtube.com/shorts/N4FIbfvViE8`

> and errors creep in. [...]
>
> GAN technologies have improved much over the years [...] present[ing] a surprising challenge to AI art practitioners—what to do now that perfect realism is within reach? [... W]orking with AI has the potential to change too, as the technology becomes more predictable and controllable, rendering blurry reflections, distorted forms and uncertain outcomes a thing of the past.' [Elliot, 2021].

I was inspired by some of the passages in these exhibition notes and wanted to make an artwork that best fit with the theme. The description of seeing a distorted representation of ourselves reflected in the results of AI-generated images rang particularly true, reflecting on the experiment with the selfies that I detailed here.

Using headshot photographs of myself, I began projecting them into the StyleGAN2 latent space and experimenting with network bending on them. In the latent for one of these headshots, the background became oversaturated off-white and was quite uniform. When ablation was applied to this, the face disappeared into the uniform background. Applying random ablation to filters in layer 5 of StyleGAN2 gave a strong resemblance to gazing at my own reflection in distorted waters. I set about creating an animated sequence that was as closely aligned to that visual metaphor as I could.

Sequencing frames where transformations are applied at random between each frame, as I had done with the images from Figure 7.13 made for very chaotic viewing, which did not really correspond to the imagery of looking at a reflection in the water. Therefore, I set about creating a more coherent temporal way of interpolating between random selections of features in a layer to manipulate. I opted to use Perlin noise [Perlin, 1985], where I could render a tensor of 3 dimensions, that would give a smooth transition between states in the tensor. I used one of the dimensions of the tensor to represent time, and the other dimension was used to map to each filter in a convolutional layer in the StyleGAN2 network. I would then use a threshold to determine whether a convolutional filter would be ablated or not in the GAN model. This threshold could be adjusted manually to get the right configuration of filters being ablated at any one time in order to get the desired visual effect, where a small fragment of my face could only ever be seen in a single frame, but when watched in motion, my overall resemblance would be seen in constructed in the viewers mind.

The final work was titled *Fragments of Self* [2021a] (Fig. 7.14). During the creation of this work, I was drawn to these images of fragmented versions of myself. On reflection, I can see that I was making this work during my recovery from Long-COVID, where during this period of chronic illness, I never felt like a whole person.[3] This illness was a constant barrage of changing ailments and symptoms, which left me with the feeling of being a fragment of my former self for a very long time.

7.4.5 Jen Sykes' *Field of View* and *The Offing*

Jennifer Sykes is an artist, designer and lecturer based between Glasgow, Scotland and London, England, where she teaches at the Creative Computing Institute, University of the Arts London. She has used network bending in the production of several artworks. Building on prior work, *Places You've Never Been* [Sykes, 2018], which used an archive of digitised film slides, captured from her family's migration from Canada to England, which was later used that to train a generative model.

In *Fields of View*, uses the clustering algorithm of network bending to 'change our interpretation to isolate "semantic grouping" that include only the sky or only the mountains of a specific narrative?' [Sykes, 2021]. Exploring the personal archive of family images of migration, network bending is used to

[3] I was one of the unfortunate people who contracted COVID-19 in the first wave in the UK, just before the March lockdown of 2020. Shortly after recovering, I became seriously ill with Post-COVID Syndrome, aka Long-COVID. For close to twelve months, an ever-changing set of physical and cognitive impairments made it nearly impossible for me to work on my PhD, and I did not fully recover until two years after I first became ill.

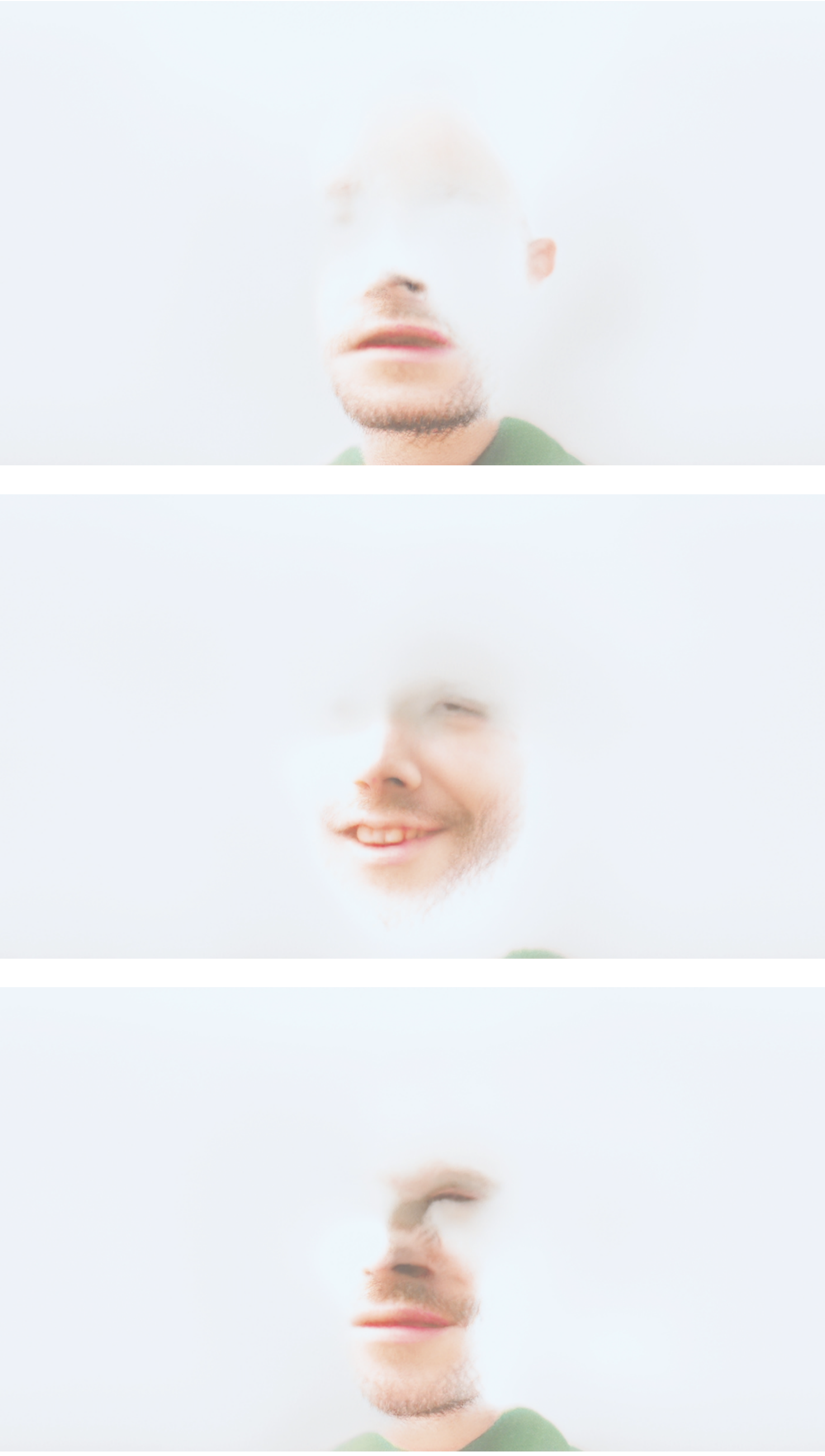


Figure 7.14: Stills from *Fragments of Self* (2021).

produce a selective generation of aspects of those archival images. Sykes likens this process to 'historic in-camera editing techniques of cameraless film-making' [Sykes, 2021] as the manipulation is happening to the features present within a dataset, with no new footage being needed.

In *The Offing* (Fig. 7.15), the same dataset and network bending transformations to manipulate landscape images. Here, clusters have been isolated that relate to the sky and the land, and these images are rotated in an animated sequence. The work produces a 'narrative stitched together through layers of the horizon; where land meets the sea' [Sykes, 2022], which is colloquially referred to as the offing.

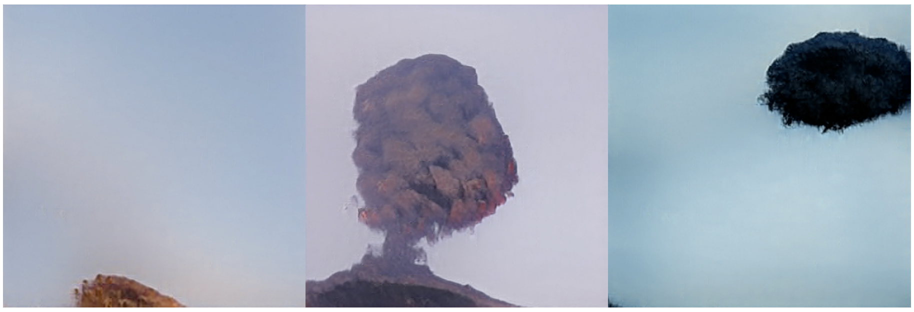


Figure 7.15: Stills from *The Offing* [Sykes, 2022]. Images courtesy of Jen Sykes.

7.4.6 Derrick Schultz's *You Are Here*

Derrick Schultz is an artist, designer and educator based in Brooklyn, New York. Schultz teaches at the Interactive Telecommunications Program at the New York University Tisch School of the Arts, and online under his own range of popular online courses for making AI art called Artificial Images. Schultz has used network bending in a number of his own artworks and has even produced tutorials showing others how to use it [Schultz, 2020b].

To create the video work *You Are Here*, [Schultz, 2020c] combined network bending with other machine-based forms of image manipulation and processing to produce original results divergent from any original training data, in a process that I categorised as 'model chaining' in the active divergence taxonomy (§6.3.5). Schultz uses a custom StyleGAN2 model trained on illustrations of flowers and renders from latent interpolation. While rendering Schultz applied network bending transformations to add rotation to the image and have that processing in real-time. That image is then fed into an image translation model (BigBiGAN) [Donahue and Simonyan, 2019] to get a different, further divergent image (Fig. 7.16 shows a visual representation of this process). The final machine learning step Schultz uses is SuperSlowMo [Jiang et al., 2018] to interpolate frames in the original sequence to extend the duration to 1000x the original duration.

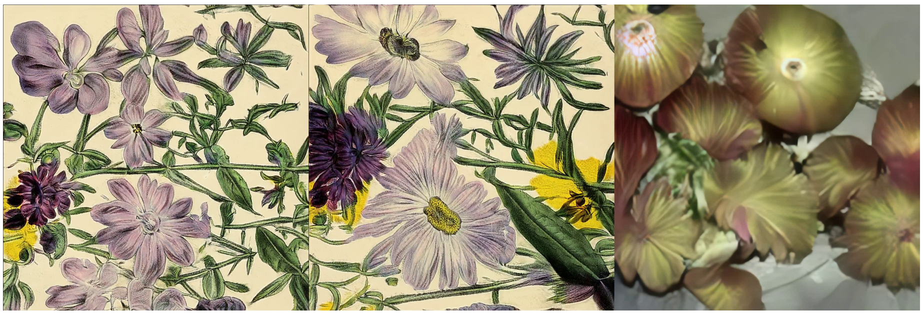


Figure 7.16: Example of the process behind making *You Are Here* [Schultz, 2020c]. Left: custom trained StyleGAN2 model. Middle: network bending rotation on StyleGAN Model. Right: BigBiGAN reinterpretation of output after network bending. Images courtesy of Derrick Schultz.

For Schultz, using this esoteric and complex chain of computational models was a way to separate himself from other AI artists who were training StyleGAN models on similar datasets, and create results that could not be produced with a generative model designed to imitate a single dataset. A further discussion of how Schultz uses network bending, in combination with other generative models and image translation techniques is given in Section 6.3.5.

7.4.7 Hans Brouwer's *Ouroboromorphism*

Ouroboromorphism [Brouwer, 2020b] (Fig. 7.17) is an audiovisual work made by Hans Brouwer, an artist and researcher working towards his Masters degree at the Delft University of Technology. This work was created as part of a broader investigation into developing audio-reactive StyleGAN latent interpolations [Brouwer, 2020a].

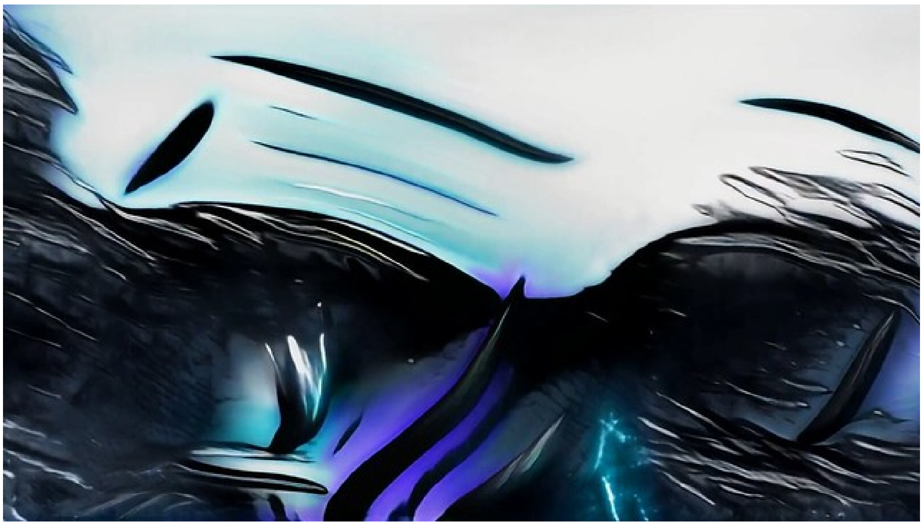


Figure 7.17: Still from *Ouroboromorphism* [Brouwer, 2020b]. Image courtesy of Hans Brouwer.

Using a custom StyleGAN model, trained on abstract imagery that resembles abstract paintings and illustrations. Brouwer uses audio features to manipulate the latent vector codes for real-time generation. In addition, he uses network bending transformations to add an additional level of control to support manipulating the visuals in response to audio, which would be possible with latent vector manipulation alone, which allows for 'increasing the musical information that can be conveyed in a given period of time' [Brouwer, 2020a].

Network bending was used in response to two sets of audio features that are recognised. If kicks (sound resembling a kick drum) are found to be in the audio sequence, a zooming effect will be made to correspond to that sequence in time. If a snare (sound resembling a snare drum) is made, then a horizontal translation is made of the visuals. Brouwer also achieves a larger resolution and 2:1 aspect ratio by mirroring the activation maps in the earlier layer of the gan so that all of the generations are doubled with then onwards, in a similar manner to how I achieved the wider aspect ratio with *Disembodied gaze* (§7.4.2) and *Fragments of self* (§7.4.4).

In addition to network bending, Brouwer also adopted model rewriting [Bau et al., 2020] as an additional active divergence method that can be used to manipulate the visual representations (§6.3.8).

7.5 Technical impact of Network Bending

The experiments presented in Chapter 5 have gone on to influence further technical development of network bending being applied to other kinds of generative models, the design of generative model architectures themselves, and have also been integrated into many user interface designs. All of these developments of network bending by others are detailed in the rest of this section.

7.5.1 Alias-Free GAN (aka StyleGAN3)

Alias-free GAN (later renamed StyleGAN3) by Kerras et al. (2021) was NVIDIA corporation's successor to their flagship StyleGAN1 and 2 neural networks. The alias-free GAN approach was designed from the ground up to be fully equivariant to the transformation of their internal representations (aka network bending). This architecture can produce internal features that are equivariant to either two kinds of transformation, translation or rotation.

One of the artefacts revealed when network bending was performed on StyleGAN2 models was the 'texture sticking' effect, which can be seen when animating a transformation, such as a translation or rotation, where the fine details are stuck to specific pixel coordinates. The authors attributed the texture sticking to 'unintentional positional references made to intermediate layers from the borders of the image, per-pixel noise inputs and aliasing between layers'. The traditional network architecture 'ha[ve] the means and motivation to amplify even the smallest amounts of aliasing and combine it over multiple scales to build a basis for texture motifs that are fixed in screen coordinate space' [Karras et al., 2021]. Reflecting on this, they make a major overhaul to the convolutional framework used in the generative model and replace the convolutional layers in the generator with pointwise convolutional layers.

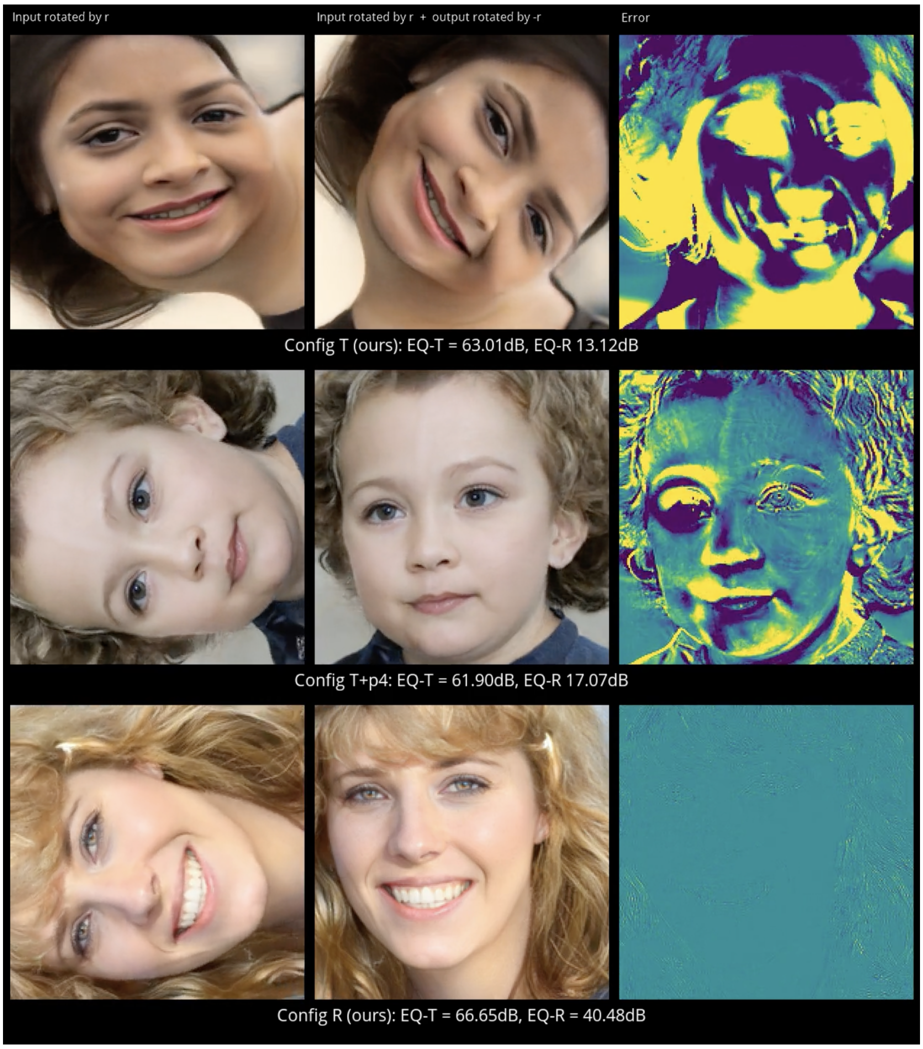

Input rotated by r
Input rotated by r + output rotated by -r
Error
Config T (ours): EQ-T = 63.01dB, EQ-R 13.12dB
Config T+p4: EQ-T = 61.90dB, EQ-R 17.07dB
Config R (ours): EQ-T = 66.65dB, EQ-R = 40.48dB

Figure 7.18: Network bending in Alias-Free GAN (StyleGAN3). [Karras et al., 2021]. Image reproduced under the Creative Commons CC BY-NC 4.0 licence.

The StyleGAN3 paper cites the original network bending paper from EvoMUSART [Broad et al., 2021b]. It is clear, from the direction of the research and its evaluation in the technical and public-facing demos that network bending has informed the advancement of the technical development work of the architecture of StyleGAN's development, as well as the evaluation of those improvements and how that is communicated publicly (Fig. 7.18). Having a network architecture that was better suited to manipulation of the internal representations of the model has, in their words: ‘pave[d] the way for generative models better suited for video and animation.’ [Karras et al., 2021].

StyleGAN3 was and largely still is the state-of-the-art in fidelity and controllability of GAN architectures. This went on to be superseded in terms of fidelity and flexibility of image generation by diffusion-based models, particularly, latent diffusion [Rombach et al., 2022] which is very intuitive and flexible to control with text-to-image conditioning. At the time of writing StyleGAN3 is still one of the leading architectures for feed-forward image generation, with network bending being core to the development of its improvements on StyleGAN2.

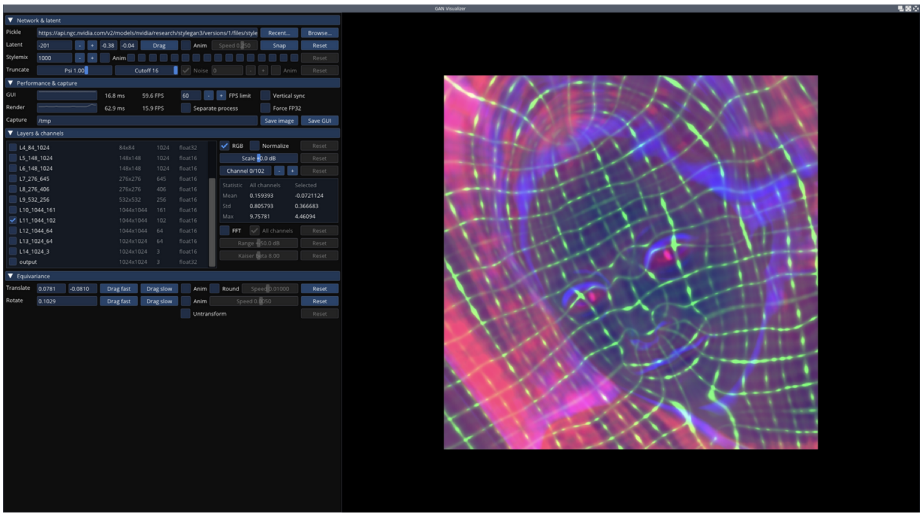
7.5.2 Interfaces Developed for Network Bending

Building an interface was something that I had originally planned as follow-on work from the original network bending paper in 2020. Unfortunately, illness and other restrictions from the pandemic impeded my ability to do that. However, in the intervening time many other people have developed their own interfaces for network bending for both image and audio generation.

7.5.2.1 StyleGAN3 Visualiser

In the release of the StyleGAN3 codebase on GitHub, NVIDIA corporation built and provided a user interface for interactively generating samples, visualising internal feature representations, and applying the x-y translation and rotation transformations (Fig. 7.19).

These translations are only applied layer-wide, but the code is configured such that animations of these transformations being applied with linearly changing parameters can be applied.



GAN Visualizer

▼ Network & latent

Pickle: https://api.ngc.nvidia.com/v2/models/nvidia/research/stylegan3/versions/1/files/style

Latent: -201 | -0.38 | -0.04 | □ Anim | Speed 0.250

Stylemix: 1000 | □ Anim

Truncate: Psi 1.00 | Cutoff 16 | ☑ Noise 0 | □ Anim

▼ Performance & capture

GUI: 16.8 ms | 59.6 FPS | 60 FPS limit | □ Vertical sync

Render: 62.9 ms | 15.9 FPS | □ Separate process | □ Force FP32

Capture: /tmp

▼ Layers & channels

□ L4_84_1024	84x84	1024	float32
□ L5_148_1024	148x148	1024	float16
□ L6_148_1024	148x148	1024	float16
□ L7_276_645	276x276	645	float16
□ L8_276_406	276x276	406	float16
□ L9_532_256	532x532	256	float16
□ L10_1044_161	1044x1044	161	float16
☑ L11_1044_102	1044x1044	102	float16
□ L12_1044_64	1044x1044	64	float16
□ L13_1024_64	1024x1024	64	float16
□ L14_1024_3	1024x1024	3	float16
□ output	1024x1024	3	float32

☑ RGB | □ Normalize

Scale +0.0 dB

Channel 0/102

Statistic	All channels	Selected
Mean	0.159393	-0.0721124
Std	0.805793	0.366683
Max	9.75781	4.46094

□ FFT | ☑ All channels

Range +50.0 dB

Kaiser beta 8.00

▼ Equivariance

Translate: 0.0781 | -0.0810 | □ Anim | □ Round | Speed 0.01000

Rotate: 0.1029 | □ Anim | Speed 0.0050

□ Untransform

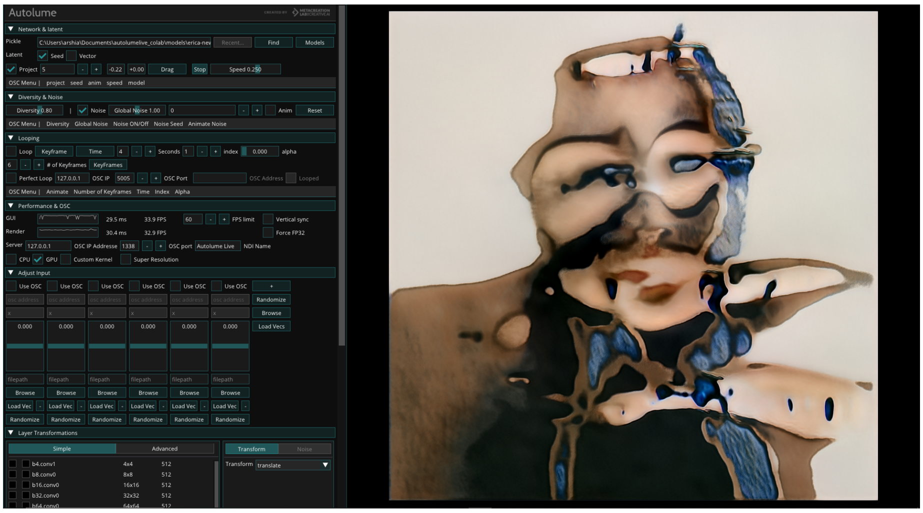
Figure 7.19: Screenshot of StyleGAN3 user interface [Karras et al., 2021]. Image reproduced under the Creative Commons CC BY-NC 4.0 licence.

7.5.2.2 Autolume

In the AutoLume-live system, [Kraasch and Pasquier, 2022, Kraasch, 2023] network bending is one of several features integrated into a real-time GAN-based VJing (Video Jockey) system. The latent vectors are determined by musical features amplitude, pitch and onset strength. These audio features create latent trajectories for the animation created with the GAN. A Graphical User Interface (GUI) which can also be controlled using MIDI (Musical Instrument Digital Interface) is developed to allow the user to improvise and adjust this generative process in real time, with network bending transformations being one of the manipulations that can be made (Fig. 7.20). Autolume can be operated using a physical mixing desk interface using MIDI.

7.5.2.3 StyleGAN-Canvas

StyleGAN-Canvas is a mixed-initiative interface [Zheng, 2023], combining image-to-image translation with rendering performed by StyleGAN3. A custom encoder was trained to perform image to latent real-time encoding, allowing users to take webcam or other input images and use that as the starting point for



Autolume

▼ Network & latent

Pickle C:\Users\arshia\Documents\autolumelive_colab\models\erica-nev Recent... Find Models

Latent ☑ Seed □ Vector

☑ Project 5 - + -0.22 +0.00 Drag Stop Speed 0.250

OSC Menu | project seed anim speed model

▼ Diversity & Noise

Diversity 0.80 | ☑ Noise Global Noise 1.00 0 - + □ Anim Reset

OSC Menu | Diversity Global Noise Noise ON/Off Noise Seed Animate Noise

▼ Looping

□ Loop Keyframe Time 4 - + Seconds 1 - + index 0.000 alpha

6 - + # of Keyframes KeyFrames

□ Perfect Loop 127.0.0.1 OSC IP 5005 - + OSC Port OSC Address □ Looped

OSC Menu | Animate Number of Keyframes Time Index Alpha

▼ Performance & OSC

GUI 29.5 ms 33.9 FPS 60 - + FPS limit □ Vertical sync

Render 30.4 ms 32.9 FPS □ Force FP32

Server 127.0.0.1 OSC IP Addresse 1338 - + OSC port Autolume Live NDI Name

□ CPU ☑ GPU □ Custom Kernel □ Super Resolution

▼ Adjust Input

□ Use OSC □ Use OSC □ Use OSC □ Use OSC □ Use OSC □ Use OSC +

osc address osc address osc address osc address osc address osc address Randomize

x x x x x x Browse

0.000 0.000 0.000 0.000 0.000 0.000 Load Vecs

filepath filepath filepath filepath filepath filepath

Browse Browse Browse Browse Browse Browse

Load Vec - Load Vec - Load Vec - Load Vec - Load Vec - Load Vec -

Randomize Randomize Randomize Randomize Randomize Randomize

▼ Layer Transformations

Simple Advanced | Transform Noise

□ □ b4.conv1 4x4 512

□ □ b8.conv0 8x8 512

□ □ b16.conv0 16x16 512

□ □ b32.conv0 32x32 512

□ □ b64.conv0 64x64 512

Transform translate ▼

Figure 7.20: Screenshot of Autolume-live user interface [Kraasch, 2023]. Image courtesy of Jonas Kraasch.

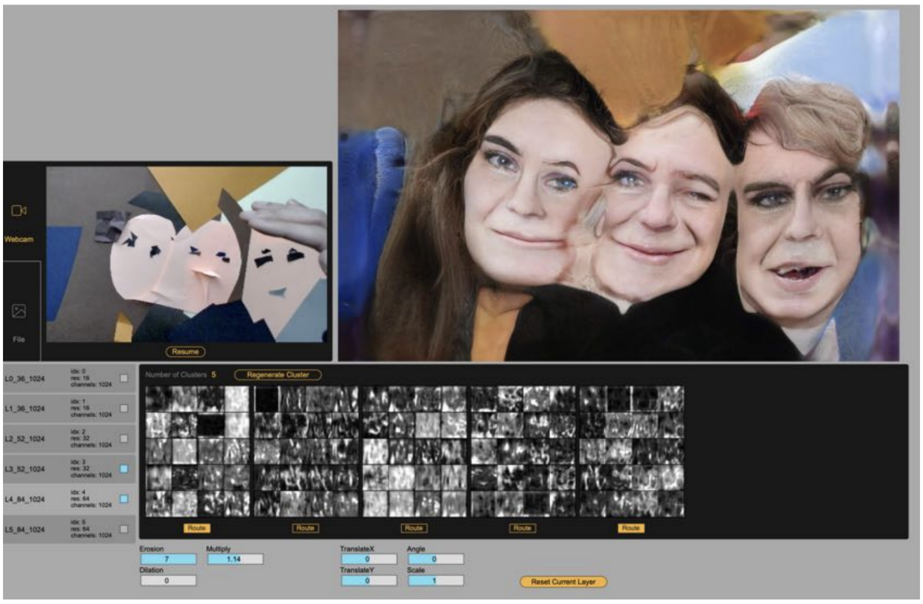
GAN rendering (Fig. 7.21). Parameters for controlling network bending transformations: erosion, dilation, pointwise scalar multiplication, x-y translations, rotation, and scaling. The clustering algorithm described in the previous chapter has also been implemented and clusters for StyleGAN3 models were calculated and integrated into the interface.

7.5.2.4 Network Bending Audio Inteface

The musician and researcher Nao Tokui built his own user interface applying Network Bending to audio [Tokui, 2023]. This was done using a StyleGAN model trained on spectrograms (in a similar fashion to §5.6). This user interface was designed for real-time performance, where the transformations are applied in real-time to a model generating spectrograms, the output of which is being looped for real-time musical performance (Fig. 7.22).

7.6 Further Advancements of Network Bending

Network bending was originally built to be run in feed-forward generative models that use a convolutional architecture, like GANs or VAEs. However, the

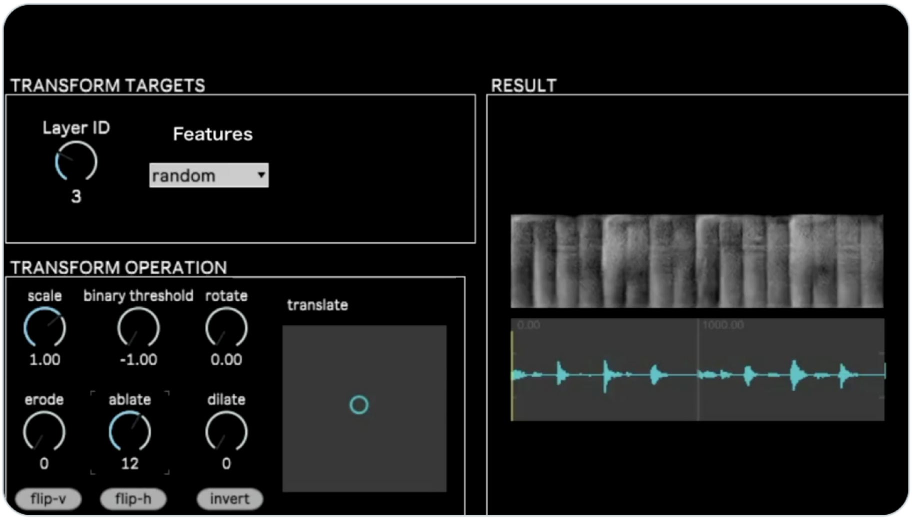


L0_36_1024 idx: 0 res: 16 channels: 1024 □
L1_36_1024 idx: 1 res: 16 channels: 1024 □
L2_52_1024 idx: 2 res: 32 channels: 1024 □
L3_52_1024 idx: 3 res: 32 channels: 1024 ☑
L4_84_1024 idx: 4 res: 64 channels: 1024 ☑
L5_84_1024 idx: 5 res: 64 channels: 1024 □

Number of Clusters 5

Erosion 7
Multiply 1.14
Dilation 0
TranslateX 0
TranslateY 0
Angle 0
Scale 1

Figure 7.21: Screenshot of StyleGAN-Canvas user interface [Zheng, 2023]. Image courtesy of Shouyang Zheng.



Layer ID 3
Features random
scale 1.00
binary threshold -1.00
rotate 0.00
erode 0
ablate 12
dilate 0

Figure 7.22: Screenshot of Network bending audio user interface [Tokui, 2023]. Image courtesy of Nao Tokui.

principal can be applied to other architecture of generative models and does not need to even use the paradigm of deterministically controlled filters. The rest of this section details extensions of network bending beyond the original paradigm described in Chapter 5.

7.6.1 Network Bending DDSP

The first extension of network bending was undertaken by Matthew Yee-King and Louis McCallum [McCallum and Yee-King, 2020, Yee-King and McCallum, 2021]. In this work, they took the Differential Digital Signal Processing model (DDSP) from Google Magenta, which is a neural network for audio synthesis and manipulation for tasks such as timbre transfer. The DDSP model takes frequency and amplitude values and it outputs 101 control values for an oscillator and noise filter parameters. In the network bending DDSP framework, network bending transformations are applied to the three layers in the neural network where all of the features are combined. There are four different types of transformation in this work: ablate, invert and binary threshold have been kept from the work described in the last chapter. In addition, Yee-King and McCallum implemented an oscillate transformation that performs a sin wave transformation based on the frequency and the depth of the layer in the network.

7.6.2 Network Bending Diffusion Models

Dzwonczyk et al. [2024] apply the standard approach of network bending to denoising diffusion generative models (Fig. 7.23). In this work, they apply the same point-wise, affine and morphological transformations as described in Section 5.3 to convolutional activation maps in the U-Net model [Ronneberger et al., 2015] that is used in latent diffusion models [Rombach et al., 2022]. Latent diffusion models can be conditioned on text, to perform text-to-image generation. By using network bending in the text-to-image pipeline, it is shown that altering the features can cause semantic shifts to occur between concepts, depending on the network bending parameters used.

7.6.3 Differentiable Network Bending

In differentiable network bending, Aldegheri et al. [2023] extend the concept of network bending from inserting deterministically controlled filters into models,

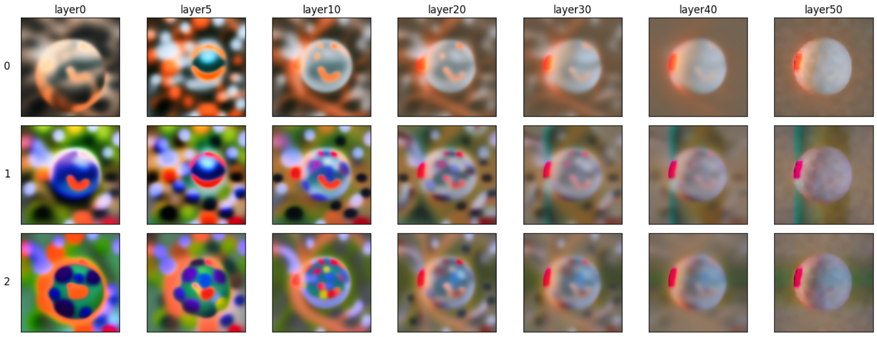


Figure 7.23: Network bending in diffusion models [Dzwonczyk et al., 2024]. Applying erosion with normalization at different layers to the prompt 'a floating orb'. Each column shows normalization happening across a different dimension. Image courtesy of the Luke Dzwonczyk.

to inserting additional modules into models that can be trained using gradient descent optimisation. In the work, they insert additional layers into pre-trained GANs that transform the activation maps of all of the convolutional filters in a layer. They optimise the weights of this new layer using CLIP [Radford et al., 2021] towards matching a pre-set text prompt (Fig. 7.24). In the paper, they note that using CLIP to optimise text prompts is just one possible way that this kind of system could be optimised.

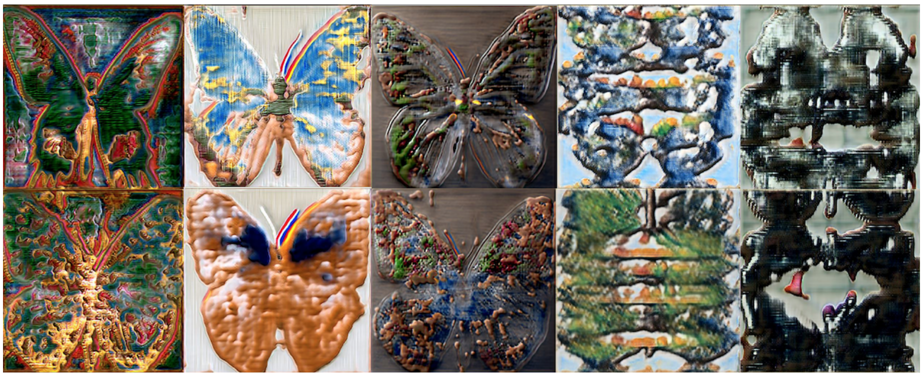


Figure 7.24: Examples of differentiable network bending [Aldegheri et al., 2023] applied to a GAN model trained on butterflies, with the differential network bending model trained to optimise various text prompts. Image courtesy of Giacomo Aldegheri.

7.7 Conclusion

This chapter has detailed the impact the experimental work in my PhD has had, in both the cultural and technical sectors. This includes detailing the artworks made by myself and other practitioners, and the work done to extend and build interfaces to interact with the work I have developed. In particular, the network bending framework has been the work that has been most adopted by others. In all these cases, network bending has been adapted to allow for further generative possibilities that were available with traditional training of generative models. Referring back to the title of this thesis, *Expanding the generative space*, it is this piece of work that has most successfully had an impact in that regard.

The next chapter will reflect on the technical contributions of this thesis, as well as the impact it has had and the broader developments that have happened in the field during the course of my work on this PhD.

Chapter 8

Discussion

In this chapter, I will reflect on the work presented in this thesis, and situate it in the context of other developments and trends that have emerged in CreativeAI and AI-Art during the course of this research. Many of the arguments presented here were also disseminated in the paper *'Using Generative AI as an Artistic Material: A Hacker's Guide'* that I presented at the 2nd international workshop on eXplainable AI for the Arts (xAIxArts) at the ACM Creativity and Cognition Conference [Broad, 2024].

8.1 Hacking as Research Methodology

Hacking has many definitions that encompass technical practices, subcultures and ethical philosophies [Jordan, 2017], though it often gets associated with jailbreaking and circumventing cybersecurity measures, what Stallman [2002] labels as *cracking*. However, hacking encompasses a much broader approach to working with technology. Eryk Salvaggio, one of the members of the Algorithmic Resistance Research Group (ARRG!), labels their approach to understanding complex algorithmic structures through artistic practices as 'the creative misuse of technology' [Salvaggio, 2023b].

Hacking is also understood through practices of making, design and technological experimentation [Hunsinger and Schrock, 2016]. Here, hackers are defined as those 'who are interested in acquiring knowledge about programming systems by venturing beyond their limits' and are understood as 'skilled individuals who possess proficiency in network and computer systems as well as a desire for intellectual challenges' [Richterich and Wenz, 2017]. The widespread phenomenon of hackathons, where technical practitioners are invited to work intensively, usually around a specific technology and develop a new invention by playfully using that technology is now seen as an approach to design research through making [Flus and Hurst, 2021, Falk et al., 2022, Rys, 2023].

Hacking can also be viewed as a performative act. In 'Hacking Perl in nightclubs', McLean [2004], describes a musical artistic practice of coding live music in nightclubs, where the playful experimentation of code is the live artistic practice itself. This practice has spawned an entire discipline of creative endeavour and academic research described as live coding [Selvaraj et al., 2021]. In live coding, coding itself is the creative material, explored in a performative setting where code itself is understood as both aesthetic and political expression [Cox and McLean, 2012].

Electronic hardware is something that can be tinkered with, experimented with and hacked [Collins, 2004, Grand et al., 2004]. In hardware hacking, repurposing existing electronic hardware can be used for music-making [Collins, 2009] through the practice of circuit bending [Ghazala, 2005],[1] and other forms of physical expressions [Hartmann et al., 2008]. Exploring the limits of specialist technical equipment, and repurposing it for unintended acts allows for new, divergent possibilities in design to be achieved through playful experimentation [Goddard and Cercos, 2015].

Just like hardware electronics, I see generative neural networks as complex structures, with many contingencies made up of discrete, tinkerable elements. Throughout the research conducted in this thesis, this approach of hacking deliberately tries to break the normal functions of generative neural networks

[1] Circuit bending was the inspiration for the name *network bending* (Ch. 5).

during inference and training. All three of the chapters of original research in this thesis came from deliberately trying to either break the models themselves or the training procedures used to create them. This line of enquiry actively goes against the common assumptions of orthodoxies of AI research, which is both heavily formalised and driven by ideology [Sias, 2021] that conforms to techno-optimist [Andreesen, 2023] and technological determinist [Drew, 2016] views of progress in AI research.

My goal with the research presented in this thesis was to find an alternative way of working with and thinking about generative AI that escapes narratives of technological determinism, and decentering of human agency in creative practice by AI [Zeilinger, 2021]. All of the outcomes of the research experiments are technological interventions that produce aesthetic outcomes that help us better understand the functioning of technological systems used to produce them (§8.4). I would not have been able to come up with the ideas that I have for all of the methods introduced, were it not for this approach of hacking.

The other common thread in the line of enquiry that underpinned the research in this thesis was to deliberately seek out the unknown or unexpected. When formulating a possible technical intervention, if I could not predict what the generated result would look like, then that was a strong motivator for me to carry out this intervention as an experiment. Rather than the hypothesis driven, convergent view of research, conducting research through the deliberate seeking out of the unknown is an established practice in design research, as Downton [2003] states 'the truly inventive [approach to research] demands a divergent view – a seeking of the unknown and unexpected'. This line of enquiry closely aligns with the approach taken in hacking, where seeking out the unexpected affordances of technologies neatly dovetails with Stallman's [2002] definition of hacking as the playful exploration of the limits of what is possible with a given technology.

Each of the experiments conducted in this thesis resulted in artworks. It was the technological intervention itself (aka *the hack*) that was the key to determining the aesthetic outcomes in these works and building a narrative for viewers to understand them. The generated outputs are the means through which we can better understand these complex technical systems by repurposing them. In all of these experiments, I view *the hack* as where the creative agency has occurred, and I do not consider any of these works as having shared creative agency with the algorithms, which is a more common framing in the discourse around AI-art [Moruzzi, 2022]. Instead of viewing AI algorithms as tools, or means to automate human agency out of the creative process, I have viewed these systems as artistic materials in their own right (§8.3).

8.2 The Role of Aesthetic Judgement in Generative AI Research

Aesthetic judgment is a commonly used metric in determining the progress of research in AI [Stanley, 2018], even if this is not explicitly stated in the values used to measure the progress of AI research [Birhane et al., 2022]. The aesthetic values that drive generative AI research are often those of realism, perfect imitation, and creating outputs indistinguishable from human outputs (as was first formulated in the imitation game [Turing, 1950]).

In the research presented in this thesis, aesthetic judgement has been central to the development of and evaluation of the experiments and generated outputs, but here the guiding aesthetic qualities have been novelty and divergence from the qualities of human outputs in the original training datasets. Whilst aesthetic judgment is an imprecise measure, the 'effectiveness of various computational media processes in improving creative output is the most substantial measure of their value' [Brown and Sorensen, 2009]. In my research, the effectiveness has been evaluated as how novel the aesthetic output is. In determining whether an intervention would qualify as being novel, I have used the yardstick of how easily the generated outputs would be able to be reproduced using other, more conventional methods. If the outputs are so distinct as to be impossible to reproduce by other means (including more conventional approaches to using generative AI), that has been the means by which I have considered various experiments as having enough value to the wider world that it should be disseminated through academic publishing and as artworks (which I have discussed in detail in Chapter 7).

8.3 Using Generative AI as an Artistic Material

Throughout all three of the chapters of the original work presented in this thesis, I took the approach of taking generative AI models, and the code that is used to train models as artistic materials themselves. Taking a hacking approach, artists can use AI in non-normative ways to create new methods of working with AI that both reveal its inner workings (§8.4) and produce new routes for artistic expression. In [Broad, 2024], I classify this into four approaches: *subverting a network's inputs*, *upending a network's training*, *corrupting a network's weights*, and *hacking the computational graph*. The original work presented in this thesis falls into two of these categories: upending a network training and hacking the computational graph.

Artists projects like Phillip Schmitt's *Introspections* [Schimtt, 2019] and Eryk Salvaggio's *Writing noise into noise* [Salvaggio, 2023c] are examples of *subverting a networks inputs.* In *Introspections* Schimtt [2019], the artist Philipp Schmitt took off-the-shelf image translation models, designed to translate photographs into line drawings and vice-versa and fed into them blank images. At first, the images returned were themselves blank, but after the outputs were repeatedly fed back into the same model many times, detailed artefacts emerged, showing complex hallucinations from the model's internal operations. In *Writing noise into noise*, Salvaggio prompted denoising diffusion models (§2.5.1.4) to generate images 'Gaussian noise', something that they are ironically very bad at doing.

Mario Klingemann's *Neural glitch* (also discussed in §2.8.4; §6.3.8) is an example of *corrupting the weights of a network.* Through the processes of altering and corrupting the learned parameters of the network, Klingemann helps to reveal its inner functionality.

Chapters 3 & 4 both represent approaches to *upending a network's training.* The experiments in Chapter 3 are the most explicit way of approaching the code frameworks of a generative neural network as an artistic material. In this work, I view the approach as akin to practices in traditional generative art, where dynamic systems are built and the role of the artist is to design or influence this process to some degree, based on intuition and exploration McCormack et al. [2004].

I consider the artworks presented in this thesis to be in the category of artistic practice described by Bense as *Generative Aesthetics*, which he describes as 'is the artificial production of probabilities, differing from the norm using theorems and programs' [Bense, 1965]. The only difference is that instead of using deterministic computer code, to developed software to produces artworks from programmes that define new statistical distributions, I have used the modern tools of GPU-optimised linear algebra libraries, differentiable objective functions and gradient-based optimisation to design and explore the characteristics of new 'aesthetic structures' that result from the embedding and production of complex statistical distributions that modern artificial neural networks make possible.

Chapter 4 more explicitly applies hacking to subvert the normal functioning of pre-existing loss functions (Berns and Colton [2020] explicitly label this approach as *loss hacking*). By inverting the adversarial loss, I was able to both reveal an otherwise unseen aspect of the discriminator's hidden perception (which is crucial to effectiveness in the fidelity of GANs) and create an explicit approach to actively diverging from data.

The original working title for the network bending paper (detailed in Ch. 5) was *hacking the computational graph.* The computational graph is the term given for the chain of computations, as defined by the input data, learned parameters, network topology, and computational functions that define the forward pass of a neural network (aka inference). Network bending allows for interventions into the computational flow of a model during inference. This approach allowed for a flexible and direct method of artistic manipulation of the internal representation of a generative model, using deterministically controlled filters that are inserted as their own layers into a generative model. The goal of this was to allow artists a direct and expressive mechanism over the flow of computation within the models themselves.

The breadth of artworks made with network bending (§7.4) and ways in which network bending has been extended (§7.5) shows the flexibility and appeal that this approach has had. It is clear that many artists want to intervene in the generative processes afforded by generative neural networks, not simply to regurgitate existing data, but to intervene in the computational processes underlying it.

8.4 Explaining AI through Artistic Enquiry

Generative neural networks produce media through a complex fabric of computation, contingent on large scraped datasets, where features and representations get encoded into the weights of unfathomably large data arrays, which in turn are enmeshed through complex chains of computation. The ease and realism through which this generated media is mass-produced and its almost uncanny flawlessness [Smith and Cook, 2023] makes it easy to forget the complex computational contingencies that produce it. I argue that the work presented in this thesis shows that rather than simply using generative neural networks as a tool, treating it critically as an artistic material can help bring this complex fabric of computation to the fore.

These approaches are not dissimilar to the *glitch art* and *databending* movements that were likewise seeking to reveal, through imperfection, otherwise hidden aspects and material functionality of digital media [Kemper, 2023]. By making targeted interventions to inputs, weights, training and inference of generative neural networks, artists are able to make critical works that reveal to us otherwise unseen aspects of these models. Taking a hacker's ethos to generative neural networks provides a critical approach for explainable AI (XAI) in the arts, where the artworks themselves present new ways of understanding and making sense of these unfathomably complex computational systems.

8.5 Assessing Impact Through Generalisation

An important measure of impact, when it comes to practice-led interventions and hacks in creative media technologies is generalisation. Being able to reuse, and reapply an intervention 'is important not only for making contributions to society at large through effective knowledge transfer but also to empower the researcher/practitioner in their future work' [Brown and Sorensen, 2009]. If we assess various chapters of original research, presented in this thesis (Chs. 3, 4 & 5), then the network bending framework (Ch. 5) is clearly the original research contribution that has had the most impact. Network bending has been widely used by both artists (§7.4) and extended into new generative domains and paradigms for interactions by other researchers (§7.5). This demonstrates the generalisability of network bending as an approach to both hacking and intervening in models, and in increasing the creative agency of artists over the functioning of generative neural networks. In addition to this, network bending is the most flexible way of *expanding the generative space* of AI models, beyond the straightforward imitation of data, and towards the repurposing of these models towards new aesthetic possibilities.

Chapter 9

Conclusion

This thesis presents three novel approaches to training, fine-tuning, and intervening in the process of inference with generative neural networks, that allow for data-divergent generation, aka *active divergence* (Ch. 6). All of the methods for achieving this are data-free, meaning no data is used in the technical intervention needed for achieving active divergence. This distinction is an important one, as much of the research and development in generative AI increasingly relies on the widespread use of data, often scraped from the web, without the consent of either the creators or publishers. This crisis of consent [Longpre et al., 2024], and the major backlash against generative AI from communities of creative practitioners [Whiddington, 2022], provide clear evidence that finding ways of using generative AI that does not directly derive its value from the aggregated efforts of human labour (even if done lawfully under the legal doctrines of 'fair-use' [Sobel, 2017, Alhadeff et al., 2024] and 'fair-dealing' [Guadamuz, 2023]) is an important direction of research.

As well as legal arguments, there is both a moral and aesthetic argument, that we should be striving to move beyond simply faithfully imitating data and replicating existing cultural capital with generative AI. Rafferty [2016] argues that a lot of contemporary cultural production is simply replicating existing cultural capital without furthering it[1] (an observation also made by Mark Fisher [2009]) and that this tendency has only been entrenched and codified by modern developments in generative AI. With the work in this thesis, I have sought to find alternative ways in which generative AI can be used, and not using data in these methods has been key to moving beyond the orthodoxies of generative AI and its derivation of value from existing cultural capital.

9.1 Contributions

In this section, I will outline the four major contributions of this thesis, including three categorical contributions to methods for achieving active divergence, and finally a formal taxonomy of active divergence methods.

9.1.1 Training without Data

Chapter 3 documents the first peer-reviewed and published approach to training generative neural networks without data, one of the three categorical contributions to active divergence methods (§6.3.3) presented in this thesis. Whilst this is not an approach that has been widely adopted by others, the series of artworks *(un)stable equilibrium* that came from these experiments has received well in the art world (§7.2), winning the Grand Prize in the ICCV Compter Vision Art Gallery, and being exhibited internationally in arts festivals, and in both commercial and non-commercial art galleries.

9.1.2 Divergent Fine-Tuning

Chapter 4 documents the first peer-reviewed and published approach to the divergent fine-tuning of generative AI models without the use of imitation-based learning. This experiment went on the inform the initial definition of *active divergence* [Berns and Colton, 2020] and can be viewed as a categorical contri-

[1] Rafferty [2016] makes this argument in discussion of my 2016 artwork *Blade Runner - Autoencoded*, which is detailed in Section 1.2.

bution to active divergence, an approach that has had many other approaches to implementation (§6.3.4).

9.1.3 Network Bending

Chapter 5, presents the network bending framework and is the third categorical contribution to active divergence methods presented in this thesis (§6.3.6). Of the three chapters of original research, this is the one that has had the most impact (§7.4; §7.5), being widely reused and adopted by many other artists and researchers, including inspiring the development of the next generation of StyleGAN models [Karras et al., 2021]. In addition, this is the approach that most successfully achieves the main goal of this thesis, which was to *expand the generative space* of generative AI. Network bending provides a flexible, controllable, and general approach to intervening in the computational process of inference in generative neural networks, and is the method that has the most scope for future research to build upon this method.

9.1.4 Active Divergence Taxonomy

The final contribution of this thesis is the survey and formal taxonomy of active divergence methods presented in Chapter 6. This survey presents a clear delineation and methods of active divergence, and of the eight categories outlined, examples of three of those categorical contributions were first published in the experiments detailed in Chapters 3, 4 & 5.

9.2 Limitations

Whilst three categorical contributions have been made to active divergence methods in this thesis, they have all primarily been demonstrated on feed-forward generative models for image generation. StyleGAN [Karras et al., 2019] and StyleGAN2 [Karras et al., 2020] were the primary models used in these experiments, and of the three experimental approaches, only network bending has been demonstrated to be generalised to domains beyond image generation and with other kinds of models (§7.5).

This work has relied heavily on aesthetic evaluation of these outputs of the generative systems in determining their value (§8.2), and whilst this has been a valuable yard-stick in assessing approaches that do not have clear means of quantitative evaluation, this does restrict the weight of the evaluation. Instead, I have relied on evaluating the impact of these works through their artistic reception and detailing how they have gone on to inspire other developments in research (Ch. 7), focusing heavily on the reuse of these techniques in assessing the impact of knowledge transfer from their dissemination (§8.5). In Section 9.3, I will discuss ways that more formal evaluations of active divergence methods could be undertaken as a possible direction for future research.

9.3 Future Research Directions

Here I will outline some future research directions that could be undertaken by others looking to further the contributions made in this thesis.

9.3.1 Measuring and Evaluating Active Divergence

As outlined in greater detail in Section 6.7, finding ways of measuring and evaluating active divergence methods is one potentially fruitful area of research. This could take the form of both qualitative evaluation with human evaluators, or quantitative evaluation, possibly by reusing the existing extensive literature on distributional divergence in generative models [Gretton et al., 2019]. A further area of research would be to evaluate how well quantitative measures of divergence align with human perception of divergence across distributions.

9.3.2 Alternative Approaches to Active Divergence

Also outlined in greater detail in Section 6.7, is the possibility of other methods for achieving active divergence. The taxonomy presented in Section 6.3 is by no means exhaustive, and I anticipate there could be new approaches that would be substantive categorical contributions to these approaches. Open-ended reinforcement learning (§6.7.4), and divergent RLHF (§6.7.5) are just two possible approaches that I have outlined. In addition, applying active divergence methods to autoregressive models, diffusion models, and the next generation of video generation models would be fruitful areas of research in my view.

9.3.3 Improved Analysis and Manipulation of Models

The method of analysis presented in Chapter 5 for the network bending frameworks is not without its flaws. Training a separate model for each layer of the network is an expensive and time-consuming process, and makes network bending less accessible to artists with limited access to computational hardware who want to work with custom models. Oldfield et al. [2023, 2024] have already improved in this approach by using tensor factorisation to analyse the feature maps of GANs and use that for downstream manipulation. However, analysing the appearance of feature maps still restricts these approaches to feed-forward convolutional generative models. Alternative approaches like analysing influence functions [Koh and Liang, 2017] have already been used successfully to explore the importance of training data in large language models [Choe et al., 2024]. These approaches to model analysis could easily be adapted towards analysing a wider variety of models for expressive manipulations.

9.3.4 Hacking the Next Generation of AI Models

In the years that this research has been undertaken, a huge amount of development has taken place in the architectures and approaches to training generative models. The majority of the research presented in this thesis was undertaken on GANs. There have been some efforts to extend some of this research into denoising diffusion models [Dzwonczyk et al., 2024] (§7.6.2). But there is still massive amounts of potential for hacking other kinds of generative models like transformer-based LLMs [Vaswani et al., 2017], multi-modal LLMs [Zhang et al., 2024], diffusion-transformer hybrid models [Peebles and Xie, 2023] and video diffusion models [Ho et al., 2022].

9.4 Summary

This thesis has presented three categorical contributions to methods for active divergence: training without data (Ch. 3), divergent fine-tuning (Ch. 4), and network bending (Ch. 5), all of which do not rely on any data in the process of their implementation. In addition to this, Chapter 6 presents a formal survey and taxonomy of active divergence methods. Of the three chapters of the original work, the network bending framework is the one that has had the most impact as it has been widely reused by artists (§7.4) and other researchers (§7.5). In addition, this is the approach that has most successfully expanded the generative space of generative models, with its flexibility to the application of models for different domains and architectures and its ability to be used on models trained on any dataset. The goal of this thesis was to *expand the generative space* of generative neural networks, and all three methods presented achieve this and point to a new approach to working with generative AI that does not rely on the imitation of, and derivation of data, for extracting its value and creative possibilities.

Bibliography

Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: How to embed images into the StyleGAN latent space? In *IEEE International Conference on Computer Vision*, pages 4432–4441, 2019.

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

Doron Adler. Deliberate stylegan2 ffhq corruption. fine tuned upon a tiny set [...]. `https://twitter.com/Norod78/status/1218282356391530496`, January 2020. Accessed: 2021-02-05.

Blaise Agüera y Arcas. Social intelligence. In *Advances in Neural Information Processing Systems [Keynote address]*, 2019.

aitrainingstatement.org. Statement on AI training. `https://www.aitrainingstatement.org/`, 2024. Accessed: 2024-10-26.

Henry Ajder, Giorgio Patrini, Francesco Cavalli, and Laurence Cullen. The state of deepfakes: Landscape, threats, and impact. *Amsterdam: Deeptrace*, 27, 2019.

Memo Akten. Learning to see: Hello, world! `https://www.memo.tv/works/learning-to-see-hello-world/`, 2017a. Accessed: 2024-08-21.

Memo Akten. Learning to see: Interactive. `hhttps://www.memo.tv/works/learning-to-see-interactive/`, 2017b. Accessed: 2024-08-21.

Memo Akten. Grannma MagNet – granular neural music & audio with magnitude networks. `https://www.memo.tv/works/grannma-magnet/`, November 2018. Accessed: 2021-9-30.

Memo Akten and Mick Grierson. Real-time interactive sequence generation and control with recurrent neural network ensembles. *Recurrent Neural Networks Symposium, NIPS 2016*, 2016.

Memo Akten, Rebecca Fiebrink, and Mick Grierson. Learning to see: you are what you see. In *ACM SIGGRAPH 2019 Art Gallery*, pages 1–6. ACM, 2019.

Giacomo Aldegheri, Alina Rogalska, Ahmed Youssef, and Eugenia Iofinova. Hacking generative models with differentiable network bending. *NeurIPS 2023 Workshop on Machine Learning for Creativity and Design*, 2023.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-consuming generative models go mad. In *The Twelfth International Conference on Learning Representations*, 2023.

Jacob Alhadeff, Cooper Cuene, and Max Del Real. Limits of algorithmic fair use. *Wash. JL Tech. & Arts*, 19:1, 2024.

Nouf Aljowaysir. Salaf. Presented in the Computer Vision Art Gallery 2021: `https://computervisionart.com/pieces2021/salaf/`, 2021. Accessed: 2024-07-13.

Teresa M Amabile. The social psychology of creativity: A componential conceptualization. *Journal of personality and social psychology*, 45(2):357, 1983.

Marc Andreesen. The techno-optimist manifesto. `https://a16z.com/the-techno-optimist-manifesto/`, 2023. Accessed: 2024-08-27.

Arfafax. Barycentric cross-network interpolation with different layer interpolation rates. `https://colab.research.google.com/drive/1FwOYqtUOkVYDwHrddFKBhDKcs0jJ_zuK`, 2020. Accessed: 2020-02-05.

Julian Ashbourn and Julian Ashbourn. The use of digital audio workstations and the impact on music. *Audio Technology, music, and media: From sound wave to reproduction*, pages 97–105, 2021.

Isaac Asimov. Runaround. *Astounding science fiction*, 29(1):94–103, 1942.

Rodrigo Assaf, Sahra Kunz, and Luís Teixeira. The presence of the uncanny valley between animation and cinema: A communication approach. In *Multidisciplinary Perspectives on New Media Art*, pages 97–118. IGI Global, 2020.

Aydao. Yeah stochastic weight averaging of neural networks is wild [...]. `https://twitter.com/AydaoAI/status/1234614081413406720`, March 2020. Accessed: 2021-02-05.

Thomas Back and H-P Schwefel. Evolutionary computation: An overview. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 20–29. IEEE, 1996.

Dzmitry Bahdanau. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Andrew G Barto. Intrinsic motivation and reinforcement learning. *Intrinsically motivated learning in natural and artificial systems*, pages 17–47, 2013.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proc. IEEE Conference on Computer Vsion and Pattern Recognition*, 2017.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. GAN dissection: Visualizing and understanding generative adversarial networks. In *International Conference on Learning Representations*, November 2018.

David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics (TOG)*, 38(4):1–11, 2019.

David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. Rewriting a deep generative model. In *Proc. European Conference on Computer Vision (ECCV)*, 2020.

Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6): 1554–1563, 1966.

Oren Ben-Kiki, Clark Evans, and Brian Ingerson. YAML ain't markup language (YAML™) version 1.1. *Working Draft 2008-05*, 11, 2009.

Sveinn Steinar Benediktsson. *Human creativity in the augmented age: Interrogating the works of Maurice Conti and Sougwen Chung*. PhD thesis, Iceland University of the Arts, 2019.

Max Bense. Projekte generativer ästhetik. *F. von Cube (Flg.), Was ist Kybernetik1 Grundbegriffe, Methoden, Anwendungen, dtv WR*, 4079, 1965.

Sebastian Berns and Simon Colton. Bridging generative deep learning and computational creativity. In *Proc. 11th International Conference on Computational Creativity*, 2020.

Sebastian Berns, Terence Broad, Christian Guckelsberger, and Simon Colton. Automating Generative Deep Learning for Artistic Purposes: Challenges and Opportunities. In *Proc. 12th International Conference on Computational Creativity*, 2021.

Aatish Bhatia. When a.i.'s output is a threat to a.i. itself. *The New York Times*, 2024.

Battista Biggio, B Nelson, P Laskov, et al. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, pages 1807–1814. ArXiv e-prints, 2012.

Abeba Birhane, Pratyusha Kalluri, Dallas Card, William Agnew, Ravit Dotan, and Michelle Bao. The values encoded in machine learning research. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 173–184, 2022.

Sid Black. Thanks! it's trained on faces then trained a little while [...]. `https://twitter.com/realmeatyhuman/status/1257733313885765638`, May 2020. Accessed: 2021-02-05.

Margaret A Boden. *The creative mind: Myths and mechanisms.* Psychology Press, 2004.

Benjamin David Robert Bogart. *Memory association machine: an account of the realization and interpretation of an autonomous responsive site-specific artwork.* PhD thesis, Simon Fraser University, 2008.

Benjamin David Robert Bogart. "Watching (Blade Runner)" 2016. `https://www.ekran.org/ben/portfolio/2017/02/watching-blade-runner-2016/`, 2016. Accessed: 2024-07-20.

Benjamin David Robert Bogart and Philippe Pasquier. Context machines: A series of situated and self-organizing artworks. *Leonardo*, 46(2):114–122, 2013.

Philip Bontrager, Aditi Roy, Julian Togelius, Nasir Memon, and Arun Ross. DeepMasterPrints: Generating MasterPrints for dictionary attacks via latent variable evolution. In *Proc. IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–9. IEEE, 2018.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

John S Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, pages 227–236. Springer, 1990.

Selmer Bringsjord, Paul Bello, and David Ferrucci. Creativity, the turing test, and the (better) lovelace test. *The Turing test: the elusive standard of artificial intelligence*, pages 215–239, 2003.

Pontus Brink. Dissection of a generative network for music composition. Master's thesis, KTH Royal Institute of Technology, 2019.

Terence Broad. Autoencoding video frames. Master's thesis, Goldsmiths, University of London, 2016.

Terence Broad. (un)stable equilibrium 1:1. Presented in the Computer Vision Art Gallery 2019: `https://computervisionart.com/pieces2019/unstable-equilibrium/`, 2019. Accessed: 2024-07-13.

Terence Broad. Being Foiled. `https://terencebroad.com/works/being-foiled`, 2020a. Accessed: 2021-06-30.

Terence Broad. Disembodied gaze. `https://terencebroad.com/works/disembodied-gaze`, 2020b. Accessed: 2021-10-12.

Terence Broad. Teratome. Presented in the NeurIPS 2020 AI Art Gallery: `https://www.aiartonline.com/highlights-2020/terence-broad-2/`, 2020c. Accessed: 2024-07-13.

Terence Broad. Fragments of self. `https://feralfile.com/artworks/fragments-of-self-tgx?fromExhibition=reflections-in-the-water-9ov`, 2021a. Accessed: 2021-10-12.

Terence Broad. Haunted varations - 01. `https://objkt.com/tokens/hicetnunc/482739`, 2021b. Accessed: 2024-07-13.

Terence Broad. Haunted varations - 02. `https://objkt.com/tokens/hicetnunc/482757`, 2021c. Accessed: 2024-07-13.

Terence Broad. Using generative AI as an artistic material: A hacker's guide. *XAIxArts: 2nd international workshop on eXplainable AI for the Arts at the ACM Creativity and Cognition Conference.*, 2024.

Terence Broad and Mick Grierson. Autoencoding Blade Runner: Reconstructing Films with Artificial Neural Networks. *Leonardo*, 50(4), 2017.

Terence Broad and Mick Grierson. Searching for an (un)stable equilibrium: experiments in training generative models without data. *NeurIPS 2019 Workshop on Machine Learning for Creativity and Design*, 2019a.

Terence Broad and Mick Grierson. Transforming the output of GANs by fine-tuning them with features from different datasets. *arXiv preprint arXiv:1910.02411*, 2019b.

Terence Broad, Frederic Fol Leymarie, and Mick Grierson. Amplifying the uncanny. *Proc. 8th Conference on Computation, Communication, Aesthetics and X (xCoAx)*, 2020a.

Terence Broad, Frederic Fol Leymarie, and Mick Grierson. Network bending: Manipulating the inner representations of deep generative models. *arXiv preprint arXiv:2005.12420v1*, 2020b.

Terence Broad, Sebastian Berns, Simon Colton, and Mick Grierson. Active Divergence with Generative Deep Learning - A Survey and Taxonomy. In *Proc. 12th International Conference on Computational Creativity*, 2021a.

Terence Broad, Frederic Fol Leymarie, and Mick Grierson. Network bending: Expressive manipulation of deep generative models. *Proc. 10th International Conference on Artificial Intelligence in Music, Sound, Art and Design (Evo-MUSART).*, 2021b.

Terence Broad, Frederic Fol Leymarie, and Mick Grierson. Network bending: Expressive manipulation of generative models in multiple domains. *Entropy*, 24(1):28, 2022.

Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

Hans Brouwer. Audio-reactive latent interpolations with StyleGAN. *NeurIPS 2020 Workshop on Machine Learning for Creativity and Design*, 2020a.

Hans Brouwer. Ouroboromorphism. `https://www.youtube.com/watch?v=fhZHjBsa0p4/`, 2020b. Accessed: 2024-07-13.

Andrew R Brown and Andrew Sorensen. Integrating creative practice and research in the digital media arts. *Practice-led research, research-led practice in the creative arts*, pages 154–165, 2009.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Donald Thomas Campbell. Blind variation and selective retention in scientific discovery. *Psychological Review*, 67:380–400, 1960.

Linda Candy. Practice based research: A guide. *CCS report*, 1(2):1–19, 2006.

Linda Candy and Ernest Edmonds. Modeling co-creativity in art and technology. In *Proceedings of the 4th conference on Creativity & cognition*, pages 134–141, 2002.

Anders Carlsson. The forgotten pioneers of creative hacking and social networking–introducing the demoscene. *MEDIA ART HISTORY 09*, page 16, 2019.

Brian Caulfield. MoMA installation marks breakthrough for AI art. `https://blogs.nvidia.com/blog/2022/11/17/moma-ai-art/`, 2022. Accessed: 2023-08-04.

M Emre Celebi, Hassan A Kingravi, and Patricio A Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert systems with applications*, 40(1):200–210, 2013.

Claudio Celis Bueno and María Jesús Schultz Abarca. Memo akten's learning to see: from machine vision to the machinic unconscious. *AI & SOCIETY*, 36:1177–1187, 2021.

Axel Chemla–Romeu-Santos and Philippe Esling. Creative divergent synthesis with generative models. *arXiv preprint arXiv:2211.08861*, 2022.

Nuttapong Chentanez, Andrew Barto, and Satinder Singh. Intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 17, 2004.

Mehdi Cherti, Balázs Kégl, and Akin Kazakçı. Out-of-class novelty generation: an experimental foundation. In *Proc. IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2017.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Minsoo Kang, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, et al. What is your data worth to gpt? llm-scale data valuation with influence functions. *arXiv preprint arXiv:2405.13954*, 2024.

Christies. Edmond de belamy, from la famille de belamy in prints & multiples, new york, 23-25 october 2018. `https://www.christies.com/en/lot/lot-6166184`, 2018. Accessed: 2023-08-04.

Georgia Chryssouli. *The Alchemist of the Surreal and the Uncanny Valley: Jan Švankmajer, the puppet and eerie animation.* PhD thesis, University of Essex, 2019.

Joseph Clemente. The demoscene and the origins of creative computing. `https://www.hf.uio.no/imv/english/research/networks/creative-computing-hub-oslo/pages/c2ho-blog/demoscene.html`, 2025. Accessed: 2024-05-12.

Harold Cohen. The further exploits of aaron, painter. *Stanford Humanities Review*, 4(2):141–158, 1995.

Paul Cohen. Harold cohen and aaron. *Ai Magazine*, 37(4):63–66, 2016.

Nick Collins. *Hardware hacking.* Self Published, 2004.

Nick Collins, Vit Růžička, and Mick Grierson. Remixing AIs: mind swaps, hybrainity, and splicing musical models. In *Proc. The Joint Conference on AI Music Creativity*, 2020.

Nicolas Collins. *Handmade electronic music: the art of hardware hacking.* Routledge, 2009.

Simon Colton. Creativity versus the perception of creativity in computational systems. In *AAAI spring symposium: creative intelligent systems*, volume 8, page 7. Palo Alto, CA, 2008.

Simon Colton. Evolving neural style transfer blends. *Proc. 10th International Conference on Artificial Intelligence in Music, Sound, Art and Design (Evo-MUSART).*, 2021.

Simon Colton and Geraint A Wiggins. Computational creativity: The final frontier? In *Ecai*, volume 12, pages 21–26. Montpelier, 2012.

Kate Compton and Michael Mateas. Casual creators. In *Proc. 6th International Conference on Computational Creativity*, 2015.

Michael Cook and Simon Colton. Neighbouring communities: Interaction, lessons and opportunities. *Association for Computational Creativity (ACC)*, 2018.

Geoff Cox and Alex McLean. *Speaking code: Coding as aesthetic and political expression.* MIT Press, 2012.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

The depot_. The depot_ digs. `https://www.artrabbit.com/events/the-depot-digs`, 2021. Accessed: 2024-07-13.

Christopher Dobrian and Daniel Koppelman. The'E'in NIME: Musical expression with new computer interfaces. In *New Interfaces for Musical Expression*, volume 6, pages 277–282, 2006.

Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *Advances in neural information processing systems*, 32, 2019.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pages 766–774, 2014.

Peter Downton. *Design research.* RMIT Publishing, 2003.

Rob Drew. Technological determinism. *A companion to popular culture*, pages 165–183, 2016.

Luke Dzwonczyk, Carmine Emanuele Cella, and David Ban. Network bending of diffusion models for audio-visual generation. *International Conference on Digital Audio Effects (DAFx)*, 2024.

Benj Edwards. “too easy”—midjourney tests dramatic new version of its AI image generator. *Ars Technical*, 2022. Accessed: 2023-08-03.

Arne Eigenfeldt, Adam Burnett, and Philippe Pasquier. Evaluating musical metacreation in a live performance context. In *Proceedings of the Third International Conference on Computational Creativity*, pages 140–144, 2012.

Ahmed Elgammal, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. CAN: Creative adversarial networks, generating” art” by learning about styles and deviating from style norms. *Proc. 8th International Conference on Computational Creativity*, 2017.

Mohamed Elhoseiny. Creative adversarial network art work at hbo silicon valley (season 5, episode 3). `https://www.youtube.com/watch?v=_5tRYkdztBg`, 2019. Accessed: 2023-08-04.

Luba Elliot. Through the lens of AI into the hidden depths of the datasets. *Feral File*, 2021. Accessed: 2023-08-15.

Jeanette Falk, Alexander Nolte, Daniela Huppenkothen, Marion Weinzierl, Kiev Gama, Daniel Spikol, Erik Tollerud, Neil Chue Hong, Ines Knäpper, and Linda Bailey Hayden. The future of hackathon research and practice. *arXiv preprint arXiv:2211.08963*, 2022.

Jiaxin Fan, Qi Yan, Mohan Li, Guanqun Qu, and Yang Xiao. A survey on data poisoning attacks and defenses. In *2022 7th IEEE International Conference on Data Science in Cyberspace (DSC)*, pages 48–55. IEEE, 2022.

Sara Salevati Feldman. Co-creation: human and AI collaboration in creative expression. In *Electronic Visualisation and the Arts (EVA 2017)*. BCS Learning & Development, 2017.

Paulo Fernandes, João Correia, and Penousal Machado. Evolutionary latent space exploration of generative adversarial networks. In *Proc. International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pages 595–609. Springer, 2020.

Mark Fisher. *Capitalist realism: Is there no alternative?* Zero Books, 2009.

Meagan Flus and Ada Hurst. Design at hackathons: new opportunities for design research. *Design Science*, 7:e4, 2021.

Edward W Forgy. Cluster analysis of multivariate data: Efficiency versus interpretability of classifications. *biometrics*, 21:768–769, 1965.

Sigmund Freud. The ‘uncanny’. se 17. *London: Hogarth*, 1919.

Brendan J Frey, Geoffrey E Hinton, Peter Dayan, et al. Does the wake-sleep algorithm produce good density estimators? In *Advances in neural information processing systems*, pages 661–670. Citeseer, 1996.

Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

Iason Gabriel. Artificial intelligence, values, and alignment. *Minds and machines*, 30(3):411–437, 2020.

Rinon Gal. StyleGAN2-NADA. `https://github.com/rinongal/StyleGAN-nada`, 2021. Accessed: 2021-06-28.

Philip Galanter. Computational aesthetic evaluation: past and future. *Computers and creativity*, pages 255–293, 2012.

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2426–2436, 2023.

Leon Gatys, Alexander Ecker, and Matthias Bethge. A neural algorithm of artistic style. *Journal of Vision*, 16(12):326–326, 2016.

Reed Ghazala. *Circuit-Bending: Build your own alien instruments*. John Wiley & Sons, 2005.

William Goddard and Robert Cercos. Playful hacking within research-through-design. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*, pages 333–337, 2015.

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.

Kazjon Grace and Mary Lou Maher. Expectation-based models of novelty for evaluating computational creativity. In *Computational Creativity*, pages 195–209. Springer, 2019.

Joe Grand, Kevin D Mitnick, and Ryan Russell. *Hardware hacking: have fun while voiding your warranty*. Elsevier, 2004.

Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

Dejan Grba. Deep else: A critical framework for AI art. *Digital*, 2(1):1–32, 2022.

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In *International conference on machine learning*, pages 1462–1471. PMLR, 2015.

Bruce Grenville. *The uncanny: Experiments in cyborg culture*. arsenal pulp press, 2001.

Arthur Gretton, Dougal Sutherland, and Wittawat Jitkrittum. Interpretable comparison of distributions and models. In *Advances in Neural Information Processing Systems [Tutorial]*, 2019.

Frantisek Grézl, Martin Karafiát, Stanislav Kontár, and Jan Cernocky. Probabilistic and bottle-neck features for LVCSR of meetings. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–757. IEEE, 2007.

Mick Grierson. Personal communication, 2019.

Andrés Guadamuz. A scanner darkly: Copyright liability and exceptions in artificial intelligence inputs and outputs. *GRUR International*, 2:2024, 2023.

Andrés Guadamuz. Personal communication, 2024.

Joy Paul Guilford. Creativty. *American Psychologist*, 5:444–454, 1950.

Joy Paul Guilford. Creative abilities in the arts. *Psychological review*, 64(2): 110, 1957.

Matthew Guzdial and Mark O Riedl. Combinets: Creativity via recombination of neural networks. *Proc. 9th International Conference on Computational Creativity*, 2018a.

Matthew J Guzdial and Mark O Riedl. Combinatorial creativity for procedural content generation via machine learning. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018b.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in Neural Information Processing Systems 31*, 2018.

Chet Haase. A machine learning algorithm walks into a bar [...]. `https://twitter.com/chethaase/status/925715289244819458`, November 2017. Accessed: 2023-08-03.

Jun Han and Claudio Moraga. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International workshop on artificial neural networks*, pages 195–201. Springer, 1995.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering interpretable GAN controls. *arXiv preprint arXiv:2004.02546*, 2020.

Björn Hartmann, Scott Doorley, and Scott R Klemmer. Hacking, mashing, gluing: Understanding opportunistic design. *IEEE Pervasive Computing*, 7 (3):46–54, 2008.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE conference on computer vision and pattern recognition*, 2016.

Michiel Hermans and Benjamin Schrauwen. Training and analysing deep recurrent neural networks. *Advances in neural information processing systems*, 26, 2013.

Aaron Hertzmann. Visual indeterminacy in GAN art. *Leonardo*, 53(4):424–428, 2020.

Aaron Hertzmann. Art is fundamentally social. `https://aaronhertzmann.com/2021/03/22/art-is-social.html`, 2021. Accessed: 2020-03-29.

Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15, 2002.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Xiang Hui, Oren Reshef, and Luofeng Zhou. The short-term effects of generative artificial intelligence on employment: Evidence from an online labor market. *Organization Science*, 2024.

Jeremy Hunsinger and Andrew Schrock. The democratization of hacking and making, 2016.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017.

Laurent Itti and Pierre Baldi. Bayesian surprise attracts human attention. *Advances in neural information processing systems*, 18, 2005.

Laurent Itti and Pierre Baldi. Bayesian surprise attracts human attention. *Vision research*, 49(10):1295–1306, 2009.

Jennifer Jacobs, Sumit Gogia, Radomír Měch, and Joel R Brandt. Supporting expressive procedural art creation through direct manipulation. In *Proc. CHI Conference on Human Factors in Computing Systems*, pages 6330–6341, 2017.

Ernst Jentsch. On the psychology of the uncanny. *Psychiatrisch-Neurologische Wochenschrift*, 8(22):195–8, 1906.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*, 2023.

Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9000–9008, 2018.

Tim Jordan. A genealogy of hacking. *Convergence*, 23(5):528–544, 2017.

Barry L Kalman and Stan C Kwasny. Why tanh: choosing a sigmoidal function. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 4, pages 578–581. IEEE, 1992.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2017.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 2021.

James C Kaufman and Vlad P Glăveanu. An overview of creativity theories. *Creativity: An introduction*, pages 17–30, 2021.

Akın Kazakçı, Cherti Mehdi, and Balázs Kégl. Digits that are not: Generating new types through deep neural nets. In *Proc. 7th International Conference on Computational Creativity*, 2016.

Balázs Kégl, Mehdi Cherti, and Akın Kazakçı. Spurious samples in deep generative models: bug or feature? *arXiv preprint arXiv:1810.01876*, 2018.

Jakko Kemper. Glitch, the post-digital aesthetic of failure and twenty-first-century media. *European Journal of Cultural Studies*, 26(1):47–63, 2023.

Bente Kiilerich. Savedoff, frames, and parergonality. *The Journal of Aesthetics and Art Criticism*, 59(3):320–323, 2001.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2013.

Mario Klingemann. Neural glitch / mistaken identity. `https://underdestruction.com/2018/10/28/neural-glitch/`, 2018. Accessed: 2021-02-05.

Arthur Koestler. *The act of creation.* London Hutchinson, 1964.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

Jonas Kraasch. Autolume-live: An interface for live visual performances using gans. *Simon Fraser University*, 2023.

Jonas Kraasch and Philippe Pasquier. Autolume-live: Turning gans into a live vjing tool. *Proc. 10th Conference on Computation, Communication, Aesthetics and X (xCoAx)*, 2022.

Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Brian Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.

Jan Eric Kyprianidis, John Collomosse, Tinghuai Wang, and Tobias Isenberg. State of the" art": A taxonomy of artistic stylization techniques for images and video. *IEEE transactions on visualization and computer graphics*, 19(5): 866–885, 2012.

Nicholas Lambert, William Latham, and Frederic Fol Leymarie. The emergence and growth of evolutionary art: 1980–1993. *ACM SIGGRAPH 2013 art gallery*, pages 367–375, 2013.

William Latham and Stephen Todd. *Evolutionary Art and Computers.* Academic Press Inc, 1992.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. *International Conference on Learning Representations*, 2017.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11), 1998.

Joel Lehman and Kenneth O Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336, 2008.

Joel Lehman and Kenneth O Stanley. Efficiently evolving programs through the search for novelty. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 837–844, 2010.

Joel Lehman and Kenneth O Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011a.

Joel Lehman and Kenneth O Stanley. Novelty search and the problem with objectives. *Genetic programming theory and practice IX*, pages 37–56, 2011b.

Steven Levy. *Hackers: Heroes of the computer revolution*, volume 14. Anchor Press/Doubleday Garden City, NY, 1984.

Pericles Lewis. The cambridge introduction to modernism. *Cambridge UP*, 2007.

Weishi Li, Yong Peng, Miao Zhang, Liang Ding, Han Hu, and Li Shen. Deep model fusion: A survey. *arXiv preprint arXiv:2309.15698*, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Machine unlearning in generative ai: A survey. *arXiv preprint arXiv:2407.20516*, 2024.

Stuart Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137, 1982.

Andy Lomas. Cellular forms. *ACM SIGGRAPH 2014 Studio*, 2014.

Shayne Longpre, Robert Mahari, Ariel Lee, Campbell Lund, Hamidah Oderinwale, William Brannon, Nayan Saxena, Naana Obeng-Marnu, Tobin South, Cole Hunter, et al. Consent in crisis: The rapid decline of the AI data commons. *arXiv preprint arXiv:2407.14933*, 2024.

Ada Augusta, Countess of Lovelace. Notes by the translator ada augusta, countess of lovelace, on lf menabrea's "sketch of the analytical engine invented by charles babbage". scientific memoirs, selected from the transactions of foreign academies of science and learned societies, vol. 3, 666-731. Richard & John Taylor, London, UK, 1843.

Jieliang Luo. *Reinforcement Learning for Generative Art.* University of California, Santa Barbara, 2020.

Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Generating images from captions with attention. *arXiv preprint arXiv:1511.02793*, 2015.

Matty Mariansky. Transfer learning StyleGAN from ffhq faces to beetles is super weird. `https://twitter.com/mmariansky/status/1226756838613491713`, Feburary 2020. Accessed: 2021-02-04.

Colin Martindale. *The clockwork muse: The predictability of artistic change.* Basic Books, 1990.

Gonzalo Martínez, Lauren Watson, Pedro Reviriego, José Alberto Hernández, Marc Juarez, and Rik Sarkar. Combining generative artificial intelligence (ai) and the internet: Heading towards evolution or degradation? *arXiv preprint arXiv:2303.01255*, 2023.

Louis McCallum and Matthew Yee-King. Network bending neural vocoders. *NeurIPS 2020 Workshop on Machine Learning for Creativity and Design*, 2020.

Pamela McCorduck. *Aaron's code: meta-art, artificial intelligence, and the work of Harold Cohen.* Macmillan, 1991.

Jon McCormack, Alan Dorin, Troy Innocent, et al. Generative design: a paradigm for design research. *Proceedings of Futureground, Design Research Society, Melbourne*, 2004.

Kyle McDonald. How to recognize fake AI-generated images. `https://medium.com/@kcimc/how-to-recognize-fake-ai-generated-images-4d1f6f9a2842`, 2018. Accessed: 2024-08-12.

Declan McGlynn. Happy accidents: The gear that changed electronic music by mistake. *DJ Magazine*, 2017.

Alex McLean. Hacking perl in nightclubs. *on: perl. com*, 2004.

Sarnoff Mednick. The associative basis of the creative process. *Psychological review*, 69(3):220, 1962.

Midjourney. Midjourney: AI art platform. `https://www.midjourney.com/`, 2023. Accessed: 2023-08-04.

Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze the discriminator: a simple baseline for fine-tuning gans. *arXiv preprint arXiv:2002.10964*, 2020.

Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. *Google research blog*, 20(14):5, 2015.

Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. Growing neural cellular automata. *Distill*, 5(2):e23, 2020.

Masahiro Mori. The uncanny valley. *Energy*, 7(4):33–35, 1970.

Caterina Moruzzi. Creative agents: rethinking agency and creativity in human and artificial systems. *Journal of Aesthetics and Phenomenology*, 9(2):245–268, 2022.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

KP Murphy. *Machine Learning: a probabilistic perspective.* MIT press, 2012.

Vaishnavh Nagarajan and J Zico Kolter. Gradient descent gan optimization is locally stable. *Advances in neural information processing systems*, 30, 2017.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, 2010.

James Oldfield, Christos Tzelepis, Yannis Panagakis, Mihalis A Nicolaou, and Ioannis Patras. Panda: Unsupervised learning of parts and appearances in the feature maps of gans. *International Conference on Learning Representations*, 2023.

James Oldfield, Christos Tzelepis, Yannis Panagakis, Mihalis A Nicolaou, and Ioannis Patras. Bilinear models of parts and appearances in generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

OpenAI. Introducing chatgpt. `https://openai.com/index/chatgpt/`, 2022. Accessed: 2024-10-27.

OpenAI. Dall·e 2. `https://openai.com/dall-e-2`, 2022. Accessed: 2023-08-03.

OpenAI. Video generation models as world simulators. `https://openai.com/index/video-generation-models-as-world-simulators/`, 2024. Accessed: 2024-7-12.

Seungwon Park. Generating novel glyph without human data by learning to communicate. *NeurIPS 2020 Workshop on Machine Learning For Creativity and Design*, 2020.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

Micheal Peel. The problem of 'model collapse': how a lack of human data limits AI progress. *Financial Times*, 2024.

Ken Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3): 287–296, 1985.

Phoenix Perry, Rebecca Fiebrink, Mick Grierson, MJ Brueggemann, Stella Doukianou, Matthew Plummer-Fernandez, Anna Troisi, et al. Art in hci: A view from the ual creative computing institute. 2022.

Justin N. M. Pinkney. Awesome pretrained StyleGAN2. `https://github.com/justinpinkney/awesome-pretrained-stylegan2`, 2020a. Accessed: 2020-02-05.

Justin N. M. Pinkney. MATLAB StyleGAN playground. https://www.justinpinkney.com/matlab-stylegan/, 2020b. Accessed: 2021-02-05.

Justin N. M. Pinkney and Doron Adler. Resolution dependent GAN interpolation for controllable image synthesis between domains. *NeurIPS 2020 Workshop on Machine Learning for Creativity and Design*, 2020.

Diego Porres. Discriminator synthesis: On reusing the other half of generative adversarial networks. *NeurIPS 2021 Workshop on Machine Learning for Creativity and Design*, 2021.

Andrew Pouliot. GAN bending. https://darknoon.com/2020/03/03/gan-hacking/, 2020. Accessed: 2021-03-27.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 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. *arXiv preprint arXiv:2103.00020*, 2021.

Penny Rafferty. The future is only an extension of our past: Bb9 + beyond. https://www.aqnb.com/2016/07/01/the-future-is-only-an-extension-of-our-past-bb9-beyond/, 2016. Accessed: 2024-08-27.

Tapani Raiko, Harri Valpola, and Yann LeCun. Deep learning made easier by linear transformations in perceptrons. In *Artificial intelligence and statistics*, pages 924–932. PMLR, 2012.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

Dawid Ratajczyk. Uncanny valley in video games: An overview. *Homo Ludens*, (1 (12)):135–148, 2019.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Deborah Reed-Danahay. *Auto/ethnography: Rewriting the self and the social.* Routledge, 1997.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proc. 31st International Conference on Machine Learning*, 2014.

Piera Riccio and Ludovica Schaerf. Colors of ai. iccc 2024 workshop. `https://networks.h-net.org/group/announcements/20025043/colors-ai-iccc-2024-workshop`, 2024. Accessed: 2024-08-30.

Annika Richterich and Karin Wenz. Introduction. making and hacking. *Digital Culture & Society*, 3(1):5–22, 2017.

Anna Ridler. Mosaic virus. Presented in the Computer Vision Art Gallery 2018: `https://computervisionart.com/pieces/mosaic-virus/`, 2016. Accessed: 2024-07-13.

Graeme Ritchie. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines*, 17(1):67–99, 2007.

Aja Romano. A guy trained a machine to ”watch” blade runner. then things got seriously sci-fi. *Vox*, 2016. Accessed: 2023-08-03.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Rosinality. Style-Based GAN in PyTorch. `https://github.com/rosinality/style-based-gan-pytorch/`, 2019. Accessed: 2024-08-01.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Maciej Rys. Invention development. the hackathon method. *Knowledge Management Research & Practice*, 21(3):499–511, 2023.

Vit Růžička. GAN explorer. `https://github.com/previtus/GAN_explorer`, 2020. Accessed: 2020-12-17.

Babak Saleh and Ahmed Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *International Journal for Digital Art History*, 2016.

Jerry Saltz. How does A.I. art stack up against human art? (hbo). `https://youtu.be/hws1ZTlkz_I?t=155`, 2018. Accessed: 2023-08-03.

Eryk Salvaggio. The algorithmic resistance research group (ARRG!). `https://cyberneticforests.substack.com/p/the-algorithmic-resistance-research`, 2023a. Accessed: 2024-05-02.

Eryk Salvaggio. Cultural red teaming. `https://cyberneticforests.substack.com/p/cultural-red-teaming`, 2023b. Accessed: 2024-05-02.

Eryk Salvaggio. Writing noise into noise. `https://cyberneticforests.substack.com/p/writing-noise-into-noise`, 2023c. Accessed: 2024-05-02.

Helena Sarin. Playing a game of GANstruction. `https://thegradient.pub/playing-a-game-of-ganstruction/`, 2018. Accessed: 2020-12-15.

Raphael Satter. Experts: Spy used AI-generated face to connect with targets. *Associated Press*, 2019.

Rob Saunders. Multi-agent-based models of social creativity. In *Computational Creativity*, pages 305–326. Springer, 2019.

Philipp Schimtt. Introspections. `https://philippschmitt.com/work/introspections`, 2019. Accessed: 2024-05-02.

Jürgen Schmidhuber. Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. In *Workshop on anticipatory behavior in adaptive learning systems*, pages 48–76. Springer, 2008.

Derrick Schultz. Demo: How to mix models in StyleGAN2. `https://www.youtube.com/watch?v=kbRkznsv9dk`, 2020a. Accessed: 2020-02-07.

Derrick Schultz. StyleGAN2 network bending (transforming layers of a neural network), part 1: Generate static images. `https://www.youtube.com/watch?v=pSo-aLWTn14`, 2020b. Accessed: 2024-07-15.

Derrick Schultz. You Are Here. `https://artificial-images.com/project/you-are-here-machine-learning-film/`, 2020c. Accessed: 2021-06-28.

Derrick Schultz. Personal communication, 2021.

Valentin Schwind, Katrin Wolf, and Niels Henze. Avoiding the uncanny valley in virtual character design. *interactions*, 25(5):45–49, 2018.

Jimmy Secretan, Nicholas Beato, David B D Ambrosio, Adelein Rodriguez, Adam Campbell, and Kenneth O Stanley. Picbreeder: evolving pictures collaboratively online. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pages 1759–1768, 2008.

Jimmy Secretan, Nicholas Beato, David B D'Ambrosio, Adelein Rodriguez, Adam Campbell, Jeremiah T Folsom-Kovarik, and Kenneth O Stanley. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary computation*, 19(3):373–403, 2011.

Ana Selvaraj, Eda Zhang, Leo Porter, and Adalbert Gerald Soosai Raj. Live coding: A review of the literature. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, pages 164–170, 2021.

Noor Shaker. Intrinsically motivated reinforcement learning: A promising framework for procedural content generation. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE, 2016.

Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by {Text-to-Image} models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2187–2204, 2023.

Shawn Shan, Wenxin Ding, Josephine Passananti, Stanley Wu, Haitao Zheng, and Ben Y Zhao. Nightshade: Prompt-specific poisoning attacks on text-to-image generative models. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 212–212. IEEE Computer Society, 2024.

Janelle Shane. Trained a neural net on my cat and regret everything. `https://aiweirdness.com/post/615654447163621376/trained-a-neural-net-on-my-cat-and-regret`, 2020. Accessed: 2020-02-05.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of GANs for semantic face editing. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020.

Ben Shneiderman. User interfaces for creativity support tools. In *Proceedings of the 3rd conference on Creativity & cognition*, pages 15–22, 1999.

Ben Shneiderman. Creativity support tools. *Communications of the ACM*, 45 (10):116–120, 2002a.

Ben Shneiderman. Creativity support tools: a tutorial overview. In *Proceedings of the 4th conference on Creativity & cognition*, pages 1–2, 2002b.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*, 2023.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

Leonardo Sias. The ideology of AI. *Philosophy Today*, 2021.

J Simon. Artbreeder, 2020.

Joel Simon. GANBreeder app. `https://www.joelsimon.net/ganbreeder.html`, November 2018. Accessed: 2020-3-1.

Joel Simon. Dimensions of dialogue. `https://www.joelsimon.net/dimensions-of-dialogue.html`, 2019. Accessed: 2020-12-15.

Dean Keith Simonton. Scientific creativity: Discovery and invention as combinatorial. *Frontiers in psychology*, 12:721104, 2021.

Dean Keith Simonton. Serendipity and creativity in the arts and sciences: A combinatorial analysis. In *The art of serendipity*, pages 293–320. Springer, 2022.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Karl Sims. Artificial evolution for computer graphics. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 319–328, 1991.

Karl Sims. Evolving 3d morphology and behavior by competition. *Artificial life*, 1(4):353–372, 1994.

Karl Sims. Evolving virtual creatures. *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 699–706, 2023.

Satinder Singh, Richard L Lewis, Andrew G Barto, and Jonathan Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82, 2010.

Robin Sloan. Dancing the flip flop. `https://www.robinsloan.com/notes/flip-flop/`, 2012. Accessed: 2021-03-27.

Amy Smith and Michael Cook. Ai-generated imagery: A new era for the ready-made'. In *SIGGRAPH Asia 2023 Art Papers*, pages 1–4. 2023.

Benjamin LW Sobel. Artificial intelligence's fair use crisis. *Colum. JL & Arts*, 41:45, 2017.

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.

Pierre Soille. Erosion and dilation. In *Morphological Image Analysis*, pages 49–88. Springer, 1999.

Mal Som. Strange Fruit. `http://www.aiartonline.com/highlights-2020/mal-som-errthangisalive/`, 2020. Accessed: 2021-02-05.

Mal Som. Personal communication, 2021.

StabilityAI. Stable diffusion public release. `https://stability.ai/blog/stable-diffusion-public-release`, 2022. Accessed: 2023-08-04.

StabilityAI. Stability AI: AI by the people for the people. `https://stability.ai/`, 2023. Accessed: 2023-08-04.

Richard Stallman. On hacking. `https://stallman.org/articles/on-hacking.html`, 2002. Accessed: 2023-08-03.

Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8 (2):131–162, 2007.

Kenneth O Stanley. Art in the sciences of the artificial. *Leonardo*, 51(2):165–172, 2018.

Thomas Strothotte and Stefan Schlechtweg. *Non-photorealistic computer graphics: modeling, rendering, and animation.* Morgan Kaufmann, 2002.

Miriam Sturdee, Makayla Lewis, Mafalda Gamboa, Thuong Hoang, John Miers, Ilja Šmorgun, ..., and Anna Troisi. The State of the (CHI)ART. 8 2023. doi: 10.6084/m9.figshare.23921814.v1. URL https://figshare.com/articles/book/The_State_of_the_CHI_ART/23921814.

Richard S Sutton, Andrew G Barto, et al. Reinforcement learning. *Journal of Cognitive Neuroscience*, 11(1):126–134, 1999.

Keisuke Suzuki, Warrick Roseboom, David J Schwartzman, and Anil K Seth. A deep-dream virtual reality platform for studying altered perceptual phenomenology. *Scientific Reports*, 7, 2017.

Jen Sykes. Places you've never been. https://j3nsykes.github.io/PlacesYouveNeverBeen/, 2018. Accessed: 2024-07-13.

Jen Sykes. Fields of view. https://j3nsykes.github.io/FieldsOfView/, 2021. Accessed: 2024-07-13.

Jen Sykes. The offing. https://j3nsykes.github.io/TheOffing/, 2022. Accessed: 2024-07-13.

Hoang Thanh-Tung and Truyen Tran. Catastrophic forgetting and mode collapse in GANs. In *Proc. International Joint Conference on Neural Networks (IJCNN)*, 2020.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. *University of Toronto, Technical Report*, 6, 2012.

Nao Tokui. I enjoyed experimenting with the real-time "network bending" [...]. https://x.com/naotokui_en/status/1663497273274413056, 2023. Accessed: 2024-07-20.

Ellis Paul Torrance. *Torrance tests of creative thinking: Directions manual and scoring guide.* Personnel Press, Incorporated, 1966.

Ragnhild Tronstad. The uncanny in new media art. *Leonardo Electronic Almanac*, 16, 2008.

Alexey Turchin. Ai alignment problem:"human values" don't actually exist. 2019.

Alan Turing. Computing machinery and intelligence. *Mind*, 59(236):433, 1950.

UK Government. Copyright, designs and patents act 1988, section 29a. `https://www.legislation.gov.uk/ukpga/1988/48/section/29A`, 1988. Inserted (1 June 2014) by The Copyright and Rights in Performances (Research, Education, Libraries and Archives) Regulations 2014 (S.I. 2014/1372), regs. 1, 3(2).

Moisés Horta Valenzuela. MelSpecVAE. `https://github.com/moiseshorta/MelSpecVAE`, January 2021. Accessed: 2021-9-30.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Dan Ventura. Mere generation: Essential barometer or dated concept. In *Proceedings of the Seventh International Conference on Computational Creativity*, pages 17–24. Sony CSL, 2016.

Lionello Venturi. The aesthetic idea of impressionism. *The Journal of Aesthetics and Art Criticism*, 1(1):34–45, 1941.

Georgina Voss, Erin Bradner, Stefana Parascho, Truitt Elly, Chung Sougwen, et al. A conversation on automation and agency. *Acadia Publishing Company*, 2021.

Rui Wang, Joel Lehman, Aditya Rawal, Jiale Zhi, Yulun Li, Jeffrey Clune, and Kenneth Stanley. Enhanced poet: Open-ended reinforcement learning through unbounded invention of learning challenges and their solutions. In *Proc. International Conference on Machine Learning*, 2020a.

Xi Wang, Zoya Bylinskii, Aaron Hertzmann, and Robert Pepperell. Towards quantifying ambiguities in artistic images. *ACM Trans. Appl. Percept.*, September 2020b.

Megan Ward. Victorian fictions of computational creativity. *AI Narratives: A History of Imaginative Thinking about Intelligent Machines*, page 144, 2020.

McKenzie Wark. Hackers. *Theory, Culture & Society*, 23(2-3):320–322, 2006.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8: 279–292, 1992.

Paul Werbos. Beyond regression:" new tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*, 1974.

Richard Whiddington. Independent artists are fighting back against A.I. image generators with innovative online protests. *Artnet*, 2022. Accessed: 2023-08-03.

Tom White. Sampling generative networks. *arXiv preprint arXiv:1609.04468*, 2016.

Tom White. Perception engines. `https://drib.net/perception-engines`, 2018. Accessed: 2024-08-2.

Tom White. Shared visual abstractions. *arXiv preprint arXiv:1912.04217*, 2019.

Mitchell Whitelaw. *Metacreation: art and artificial life.* Mit Press, 2004.

Geraint A Wiggins, Peter Tyack, Constance Scharff, and Martin Rohrmeier. The evolutionary roots of creativity: mechanisms and motivations. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1664):20140099, 2015.

Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

Hongji Yang and Lu Zhang. Promoting creative computing: origin, scope, research and applications. *Digital Communications and Networks*, 2(2):84–91, 2016.

Matthew Yee-King and Louis McCallum. Studio report: sound synthesis with ddsp and network bending techniques. *2nd Conference on AI Music Creativity (MuMe + CSMC)*, 2021.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

Eliezer Yudkowsky. The AI alignment problem: why it is hard, and where to start. *Symbolic Systems Distinguished Speaker*, 4:1, 2016.

Telmo Zarraonandia, Paloma Diaz, and Ignacio Aedo. Using combinatorial creativity to support end-user design of digital games. *Multimedia Tools and Applications*, 76:9073–9098, 2017.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

Martin Zeilinger. *Tactical entanglements: AI art, creative agency, and the limits of intellectual property.* meson press, 2021.

Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*, 2024.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.

Jiachen Zhao, Zhun Deng, David Madras, James Zou, and Mengye Ren. Learning and forgetting unsafe examples in large language models. *arXiv preprint arXiv:2312.12736*, 2023.

Shuoyang Zheng. Stylegan-canvas: Augmenting stylegan3 for real-time human-ai co-creation. In *Joint Proceedings of the ACM IUI Workshops*, 2023.

B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5122–5130, 2017.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene CNNs. In *International Conference on Learning Representations*, 2015.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. IEEE international conference on computer vision*, pages 2223–2232, 2017.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Joanna Zylinska. *AI art: machine visions and warped dreams.* Open humanities press, 2020.