

---

# Searching for an *(un)stable equilibrium*: experiments in training generative models without data

---

**Terence Broad**

Department of Computing  
Goldsmiths, University of London  
t.broad@gold.ac.uk

**Mick Grierson**

Creative Computing Institute  
University of the Arts London  
m.grierson@arts.ac.uk

## Abstract

This paper details a developing artistic practice around an ongoing series of works called *(un)stable equilibrium*. These works are the product of using modern machine toolkits to train generative models without data, an approach akin to traditional generative art where dynamical systems are explored intuitively for their latent generative possibilities. We discuss some of the guiding principles that have been learnt in the process of experimentation, present details of the implementation of the first series of works and discuss possibilities for future experimentation.

## 1 Introduction

In this work we are utilising toolkits for data driven optimisation (such as PyTorch [1]) and pre-existing generative model architectures with strong inductive biases, to explore the latent possibilities of potential generative outputs that don't mimic any existing data distribution. We have set out to achieve this by finding ways of 'training' these generative models without any training data. This approach can be seen 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 [2]. We see this as a continuation of artistic practice describe by Bense as *Generative Aesthetics*, where we are using the modern tools of gpu-optimised linear algebra libraries, differentiable objective functions and gradient-based optimisation to design and explore the characteristics of these new 'aesthetic structures' [3].

## 2 Guiding Principles

Here we discuss some of the useful concepts and fruitful techniques that have been learnt through the process of experimentation and utilised in the works in *Series 1* (see Section 3 for details).

### Complexity | Stochasticity

A lot of the trial and error in the practice is finding the right balance of complexity and stochasticity. Often finding the right batch size is key, too low and gradients quickly explode, too high and the error signal averages out any potential system dynamics, resulting in stasis.

### Relational Constraints

We often utilise constraints that are relative to the output of a given batch. These constraints may be distances in embedding spaces using techniques from metric learning [4], or measure of diversity in pixel space of a generators batch-wise output.

## Exploiting Boundaries

We exploit the small differences in the way different differentiable functions measure distance and difference. These discrepancies can be exploited to create internal system dynamics that continually inject a level of randomness into the training dynamics.

## Diametric Optimisation

In some of the arrangements for the works in *Series 1*, we train some of the networks in the system with two diametrically opposed loss functions (propagated after exposure to different batches to prevent them from completely cancelling out). While this may be counter-intuitive from the perspective of optimisation, it provides an anchor of stability in networks ensembles where the other networks are relying on that network's output in the process of training.

## Discovering (Un)stable Equilibria

The previous guiding principles are all techniques that serve the goal of finding a balance of randomness and stability: to find an equilibrium in the space of potential system dynamics which is stable enough to prevent gradients collapsing or exploding, but unstable enough to produce unexpected results.

## 3 Case Study: *Series 1*

For the works in *Series 1*, the setup resembles the popular generative adversarial networks [5] ensemble, however here we have two generators (both using progressively-growing, style-based generator architectures [6]). The 'discriminator' sometimes acts in the traditional way as a binary classifier, trying to correctly classify which generator has produced which image. Alternatively, it is sometimes trained simultaneously with both diametrically opposing adversarial loss functions (this is true for the works *1:1*, *1:2* and *1:6*). In either case, the discriminator's classification output, and distance measurements in the discriminator's embedding space are utilised for training the generators.

The generators compete in having their output as being recognized as the output of the other network, either using the classification output of the discriminator or by having the distance of their embeddings (in discriminator space) as close as possible to the other generators. Both generators also compete to have more variety in the colours they output at pixel level (in their respective batch) than the other generator. These arrangements result in abstract, sometimes orthogonal compositions from the two generators. After training is completed, the resulting images from the two generators is presented side-by-side, as a video piece showing a synchronised interpolation between their respective latent spaces (see Figures 1-6 in the Appendix for stills from and links to the video works in the series).

## 4 Discussion and Future Work

In this work, we have developed a practice that relies heavily on the subjective aesthetic analysis of the output. Through subjective interpretation of the output of these systems (often through closely monitoring results visually throughout training), an intuition has been developed that has informed decisions in the iterations of model design. We find this practice to resonate strongly with Stanley's description of the role of artistic understanding in the process of researching artificial systems [7].

In future experiments, we look to develop the practice further and explore aesthetic possibilities of other commonly used techniques in machine learning, such as the variety of diversity metrics now used to assess generative models, such as the inception score [8] and Fréchet inception distance [9]. We also want to experiment with integrating meta-information of training performance into the training of the model, as well as adaptive and evolutionary techniques to dynamically change the model architectures and meta-model arrangements.

## Acknowledgments

This work has been supported by UK's EPSRC Centre for Doctoral Training in Intelligent Games and Game Intelligence (IGGI; grant EP/L015846/1).

## References

- [1] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. *PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration*, 6, 2017.
- [2] Jon McCormack, Alan Dorin, Troy Innocent, et al. Generative design: a paradigm for design research. *Proceedings of Futureground, Design Research Society, Melbourne*, 2004.
- [3] Max Bense. Projekte generativer ästhetik. *F. von Cube (Flg.), Was ist Kybernetik I Grundbegriffe, Methoden, Anwendungen, dtv WR*, 4079, 1965.
- [4] Brian Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.
- [5] 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.
- [6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.
- [7] Kenneth O Stanley. Art in the sciences of the artificial. *Leonardo*, 51(2):165–172, 2018.
- [8] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.
- [9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

# Appendices



Figure 1: Still from *(un)stable equilibrium 1:1*. The work can be viewed at: <https://youtu.be/r6MB555mXXM>



Figure 2: Still from *(un)stable equilibrium 1:2*. The work can be viewed at: [https://youtu.be/P\\_LLD8ffgVc](https://youtu.be/P_LLD8ffgVc)

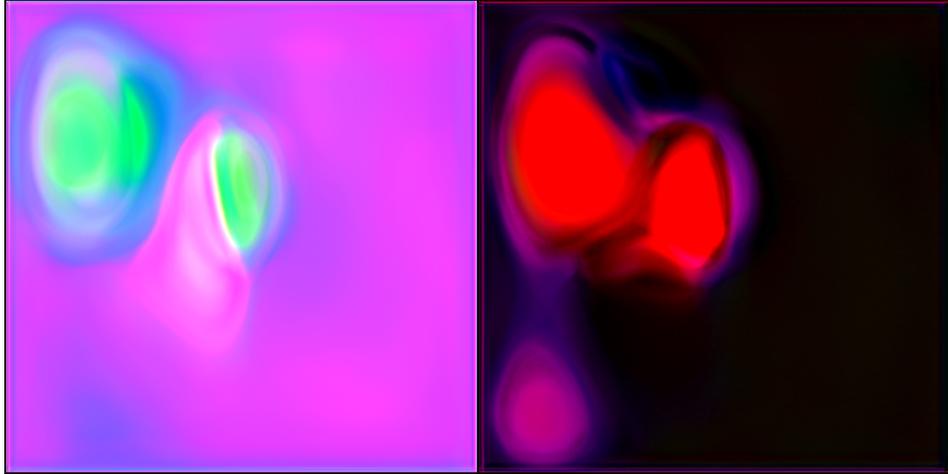


Figure 3: Still from *(un)stable equilibrium 1:3*. The work can be viewed at: <https://youtu.be/A4CsL5TAvQU>

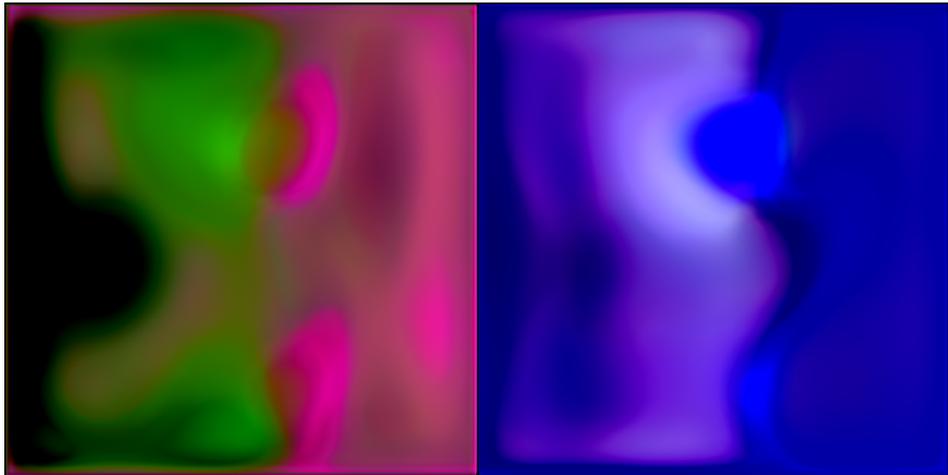


Figure 4: Still from *(un)stable equilibrium 1:4*. The work can be viewed at: [https://youtu.be/SL690ZD\\_-cM](https://youtu.be/SL690ZD_-cM)

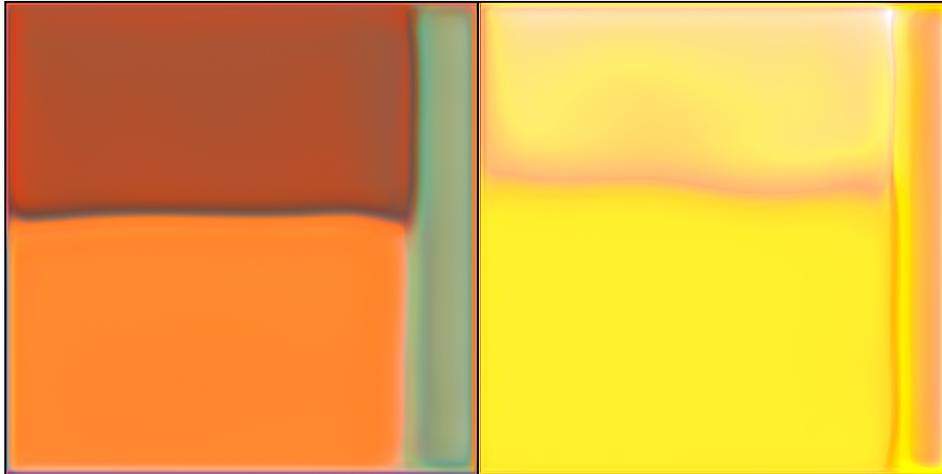


Figure 5: Still from *(un)stable equilibrium 1:5*. The work can be viewed at: <https://youtu.be/b8X-Kr04JzM>

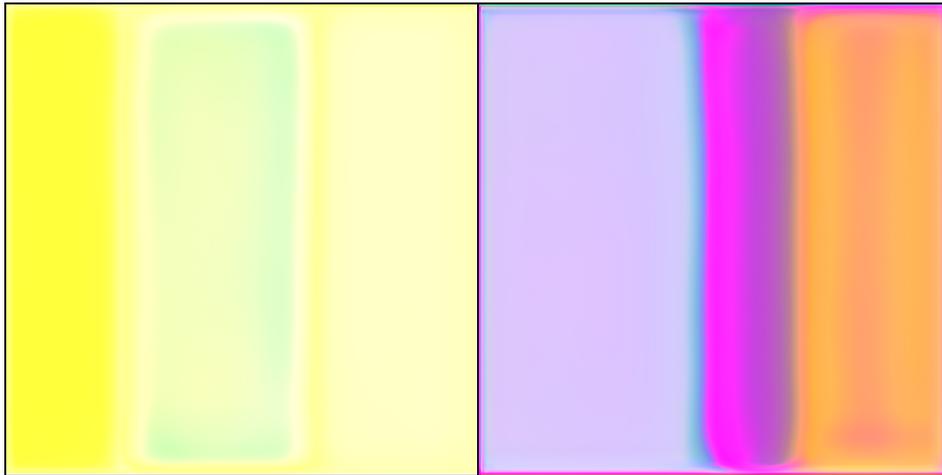


Figure 6: Still from *(un)stable equilibrium 1:6*. The work can be viewed at: <https://youtu.be/Jxhi3P2edVQ>