# *Diversity in Generative Machine Learning to Enhance Creative Applications*

Sebastian Berns
July 2024

A thesis submitted in partial fulfilment
of the requirements for the degree of
*Doctor of Philosophy*

ADVISERS

Simon Colton
Queen Mary University of London

Laurissa Tokarchuk
Queen Mary University of London

Christian Guckelsberger
Aalto University
Queen Mary University of London

EXAMINERS

Matthew Purver
Queen Mary University of London

Kazjon Grace
University of Sydney

# STATEMENT OF ORIGINALITY

I, Sebastian Berns, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged in the following and my contribution indicated. Previously published material is also acknowledged in the following section.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.



*London, July 2024*

Sebastian Berns

# PUBLICATIONS & CONTRIBUTIONS

The research outputs presented in this thesis contribute to the topic of diversity in generative machine learning for visual arts and video games. The following is a list of seven peer-reviewed publications and their contributions.

Berns, S., & Colton, S. (2020). Bridging Generative Deep Learning and Computational Creativity. *Proceedings of the 11th International Conference on Computational Creativity (ICCC)*.

- Introduction of the term *active divergence* to define strategies that consciously break, tweak or otherwise intervene in data-driven generative modelling methods for art production.
- An illustrative overview of some active divergence techniques.

Broad, T., Berns, S., Colton, S., & Grierson, M. (2021). Active Divergence with Generative Deep Learning - A Survey and Taxonomy. *Proceedings of the 12th International Conference on Computational Creativity (ICCC)*.

Berns, S., Broad, T., Guckelsberger, C., & Colton, S. (2021). Automating Generative Deep Learning for Artistic Purposes: Challenges and Opportunities. *Proceedings of the 12th International Conference on Computational Creativity (ICCC)*.

- A framework for automating generative machine learning for artistic applications by handing over more creative responsibilities to a generative system.
- Targets for automation in the generative machine learning pipeline for artistic purposes

Hagg, A., Berns, S., Asteroth, A., Colton, S., & Bäck, T. (2021). Expressivity of Parameterized and Data-driven Representations in Quality Diversity Search. *Proceedings of the Genetic and Evolutionary Computation Conference*.

- A principled study evaluating the *expressivity* of generative models, i. e. the ability to produce a wide range of different types of artefacts.
- Evidence for the limitations of generative models in terms of output diversity.
- Recommendations for the use of generative model latent spaces in quality diversity search.

Berns, S. (2022). Increasing the Diversity of Deep Generative Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36* (11).

Berns, S., Colton, S., & Guckelsberger, C. (2023). Towards Mode Balancing of Generative Models via Diversity Weights. *Proceedings of the 14th International Conference on Computational Creativity (ICCC)*.

- *Diversity weights*, a training scheme to increase a generative model's output diversity by taking into account the relative contribution of individual training examples to overall diversity.
- *Weighted Fréchet Inception Distance (wFID)*, an adaptation of the FID measure to estimate the distance between a model distribution and a target distribution modified by weights over individual training examples.
- A proof-of-concept study, demonstrating the capability of our method to increase diversity, examining the trade-off between artefact typicality and diversity.
- Implementation of our diversity weights optimisation algorithm: https://github.com/sebastianberns/diversity-weights

Berns, S., Volz, V., Tokarchuk, L., Snodgrass, S., & Guckelsberger, C. (2024). Not All the Same: Understanding and Informing Similarity Estimation in Tile-Based Video Games. *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*.

- A quantitative study comparing similarity judgements from participants (N=456) against seven existing computational similarity metrics in twelve configurations.
- A qualitative interpretation study, in which four focus groups of game experts (N=4×2) provide their interpretations of the dimensions underlying the human similarity judgement in this domain.
- Specific recommendations for the use of existing similarity metrics, based on our systematic analyses of the data.
- A large dataset of human similarity judgements in tile-based video game levels and an implementation of the metric test suite: https://github.com/sebastianberns/similarity-estimation-chi24

## OPEN-SOURCE IMPLEMENTATIONS

In addition to publications, the research in this thesis has led to the development of various open-source implementations.

*Clean Features* This software package follows best practices to compute anti-aliased image embeddings with pre-trained computer vision models (Parmar, Zhang & Zhu, 2022). A variety of standard embedding models are available (e. g. CLIP, or Inception v3) and it is easy to integrate custom models. https://github.com/sebastianberns/cleanfeatures

*Clean FID* Compute Fréchet Inception Distance (FID) and Weighted Fréchet Inception Distance (wFID) from clean image embeddings (*cleanfeatures*, above) of two data sources (tensor, generator model, or dataset) to evaluate the performance of a generative model.
https://github.com/sebastianberns/cleanfid

*Clean Precision–Recall* Compute Precision–Recall (PR) from clean image embeddings (*cleanfeatures*, above) of two data sources (tensor, generator model, or dataset) to evaluate the performance of a generative model.
https://github.com/sebastianberns/cleanpr

*Vendi Score* Improved implementation of the Vendi Score (VS) in Numpy and Pytorch.

*Generalised Non-Metric Multi-Dimensional Scaling (GNMDS)* Python implementation

*t-distributed Stochastic Triplet Embedding (t-STE)* Python implementation

# ABSTRACT

Generative machine learning methods are trained on raw data, modelling the primary patterns that constitute typical examples. They enable the production of high-quality artefacts in very complex domains and provide useful models for generative systems, in particular in the visual arts and video games. However, modelling a training data distribution perfectly is less valuable for applications in art production and video games. In particular, our analysis of the use of generative models in visual art practices motivates the need to increase the output diversity of generative models.

In this thesis, we focus on diversity, and similarity as one of its underlying relations, in generative machine learning for visual arts and video games. We make several contributions that are covered in four main chapters.

We coin the term *active divergence* to define strategies that consciously break, tweak or otherwise intervene in data-driven generative modelling methods for art production. We propose a framework for automating generative machine learning for artistic applications by handing over more creative responsibilities to a generative system.

We systematically evaluate the *expressivity* of generative models, i. e. the ability to produce a wide range of different types of artefacts. We provide evidence for the limitations of generative models in terms of output diversity and give recommendations for the use of generative model latent spaces in quality diversity search.

We propose a *diversity weights* training scheme for generative models to increase a model's output diversity by taking into account the relative contribution of individual training examples to overall diversity. In a proof-of-concept study, we demonstrate the capability of our method to increase diversity.

In two human participant studies, we evaluate how well computational metrics of similarity can approximate the human perception of similarity in tile-based video games. Our findings inform the selection of existing similarity metrics and highlight requirements for the design of new metrics to substitute human similarity evaluation.

Our findings benefit the application of generative models in generative systems, quality diversity search, art production and video games. Rather than a 'ground truth' that needs to be modelled perfectly, we argue that training datasets are merely a limited snapshot of a complex world with inherent biases. To be useful for applications in visual arts and video games, generative models require higher output diversity. Relatedly, our *diversity weights* method could contribute to efforts of equity, diversity and inclusion by reducing harmful biases in generative models.

# ACKNOWLEDGMENTS

This thesis is a tangible artefact that documents the fruits of my effort over several years. Not as well documented, however, is the impact the PhD experience had on me and the changes it caused. Today, I am a different person from when I started, possibly a new and improved me: more resilient, patient and humble, better at executing and effective communication, better at doing science, yet as curious as ever.

Many people have contributed to this improvement and helped me along the way. I am indebted to my advisors Simon Colton, Laurissa Tokarchuk and Christian Guckelsberger supporting, encouraging but also challenging me. Thank you to my co-authors, in particular Terence Broad and Alexander Hagg. Thank you to Sam, Vanessa and the whole modl.ai team for the support during my internship. Thank you to my examiners Matthew Purver and Kazjon Grace for an enjoyable viva and the valuable feedback which helped make this thesis much better.

Thank you to my friends and colleagues at QMUL and IGGI from whom I have learned a lot: Nick and Yelena, Remo, James, Bamford, Cristiana, the whole IGGI 2019 cohort, Susanne, Simon Lucas, Diego, Raluca, Amaya and the guys from the third floor, as well as many more people from the QMUL PhD community.

Thank you to my family and friends for their unconditional support. Principalmente a Maro por su amor, compañía y por bancarme en todas. Danke an die beste Mutter und den besten Vater der Welt, die niemals an mir zweifeln und alle meine verrückten Pläne unterstützen. Danke an Markus und Miriam für die Freundschaft über viele Jahre. Gràcies també als amics a Barcelona per fer-me una persona més oberta i afectosa.

This dissertation was supported by the EPSRC Centre for Doctoral Training in Intelligent Games & Games Intelligence (IGGI) [EP/S022325/1].

Work was carried out as part of the Game AI Research Group, School of Electronic Engineering and Computer Science, Queen Mary University of London. Further work was supported by modl.ai during a research internship.

Compute infrastructure was provided by the Queen Mary University of London Apocrita HPC facility, supported by QMUL Research-IT (King, Butcher & Zalewski, 2017).

# CONTENTS

# LIST OF FIGURES

## LIST OF TABLES

# LIST OF ALGORITHMS



# LIST OF ACRONYMS

| | |
|---|---|
| ANN | artificial neural network |
| AutoML | automated machine learning |
| CC | computational creativity |
| CLIP | contrastive language-image pre-training |
| DEI | diversity, equity, and inclusion |
| DivW | diversity weights |
| DL | deep learning |
| DPM | diffusion probabilistic model |
| FID | Fréchet Inception Distance |
| wFID | Weighted Fréchet Inception Distance |
| GAN | generative adversarial network |
| IS | Inception Score |
| LDM | latent diffusion model |
| LIGM | large image generation model |
| ML | machine learning |
| MMO | multi-modal optimisation |
| MS | Mode Score |
| NAS | neural architecture search |
| PCG | procedural content generation |
| PD | Pure Diversity |
| PR | Precision–Recall |
| QD | quality diversity |
| SOM | Self Organising Map |
| t-SNE | t-distributed Stochastic Neighbourhood Embedding |
| VAE | variational auto-encoder |
| VE | Voronoi Elites |
| VS | Vendi Score |
| WGAN | Wasserstein GAN |

Chapter 1

# INTRODUCTION

Generative machine learning, through continuous technical improvements and the scaling of compute resources, has made it easier than ever to generate different varieties of media content: images (Rombach et al., 2022) and videos (OpenAI, 2024), music (Evans et al., 2024) and speech (Lux et al., 2024), as well as three-dimensional objects (J. Gao et al., 2022) and scenes (Bautista et al., 2022). Generative machine learning methods are trained on raw data, modelling the primary patterns that constitute typical examples, and yield a fully working generator that can produce artefacts very similar to these training examples. Such models are useful for generative systems, e. g. for procedural content generation in video games (PCGML; Summerville et al., 2018). In contrast, designing generative systems by hand requires expert knowledge, manual analysis of relevant patterns, a lot of effort and continuous evaluation of the generator and its output. Partially automating this process through automatic program synthesis from raw data via machine learning gives significant benefits at scale. Generative modelling thus enables the production of high-quality artefacts in very complex domains which would otherwise be difficult to generate at a similar degree of fidelity.

Artefact fidelity

To better understand the term fidelity, we can consider its use in the context of music. There, the term *high fidelity* (hi-fi) is used to describe the reproduction with electrical equipment of high-quality sound that is very similar to the sound produced by the original instruments. For example, when the recording of an orchestra is played from a vinyl disc on a record player. For a sound reproduction to be of high fidelity, the audio-sensory experience of the reproduced sound has to be faithful to the live experi-

ence. We adopt this term to describe the qualities of generated samples that exhibit a high likeness to the training examples in a dataset.

Distribution fitting

Researchers have directed their efforts primarily to push for improvements in artefact fidelity. In the image domain, for a dataset of natural images, high fidelity generally means high photorealism. This is illustrated well by the rapid progress in the technological capabilities to generate human faces (Figure 1.1). To achieve this, machine learning researchers focus on developing algorithms that can faithfully model a training dataset. That is to say, the model should match the target distribution as closely as possible. Generative modelling in deep learning is therefore generally defined as a *distribution fitting* problem. For example, to produce a colour image, a model needs to determine the correct value for every pixel in three channels (RGB), often dependent on the values of neighbouring pixels. The complex high-dimensional target distribution is thus the distribution over pixel values as defined by the example images in the training dataset.

This thesis rests on the observation that, while modelling the training data distribution perfectly is beneficial for some applications, it is less valuable in an artistic setting. Visual artists embraced early image-generation techniques, in particular GANs, precisely *because* their output is imperfect. In contrast to the efforts of engineers to continuously increase the fidelity of model outputs, some artists consciously work against perfection, *actively diverging* from the target distribution. Rather than reproducing artefacts with high similarity to existing examples, imperfect generative models can yield unexpected, sometimes novel and potentially culturally valuable artefacts. Creative behaviour, judged by its results, is conventionally defined as producing *novelty*, *surprise* and *value* (Boden, 2004). Similar standard definitions highlight *originality* and *effectiveness* (Runco & Jaeger, 2012). The benefit for artistic applications, and creative tasks in general, thus lies in the production of artefacts that meet this definition. The interactions with a generative model for this purpose can be considered acts of co-creativity. Human-computer co-creativity is one of the major themes of computational creativity (CC), which has been defined as "the philosophy, science and engineering of computational systems which, by taking on particular re-

(a) 2014 (b) 2015 (c) 2016 (d) 2017 (scaled down to 25 % of the full size)

Figure 1.1: Progress of image synthesis of human faces over the years, from left to right, ordered by the year of the pre-print manuscript:
(a) Original GAN (2014) on Toronto Face Dataset (48 × 48 px)
(b) DCGAN (2015) on custom web-scraped dataset (64 × 64 px)
(c) CoGAN (2016) on CelebA dataset (128 × 128 px)
(d) ProGAN (2017) on CelebA-HQ (1024 × 1024 px, displayed at 25 %)

sponsibilities, exhibit behaviours that unbiased observers would deem to be creative" (Colton & Wiggins, 2012).

Instead of *novelty*, in this thesis, we primarily focus on *diversity* and the closely related concept of *similarity* as the foundational relation that novelty and diversity are built on. In the following, we motivate this focus on diversity and briefly define the concepts and their relation to each other.

Diversity

The *diversity* of a collection of artefacts indicates the overall dissimilarity in relevant qualities between the artefacts. The important underlying relation of diversity is thus the similarity of artefacts.

Novelty

The *novelty* of an individual artefact is typically measured relative to a collection of artefacts, as a one-to-many relation. It indicates how dissimilar in relevant qualities a novel artefact is to existing artefacts. The novelty of an artefact can also be interpreted as the amount of diversity that the artefact

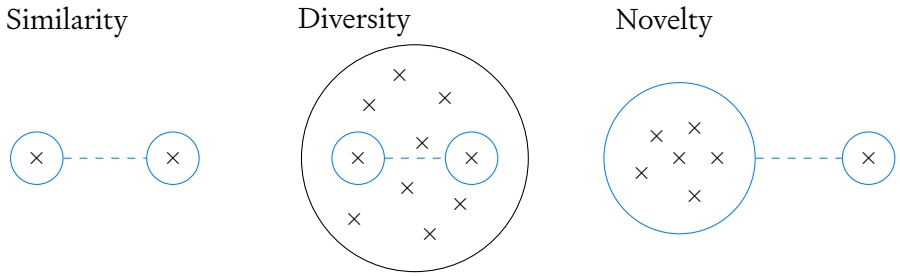



Figure 1.2: Similarity is the basic relation between artefacts (×) that underlies complex concepts like diversity and novelty.

adds to a collection. That is to say, when an artefact is added to a collection that is similar to the other artefacts, diversity will only increase slightly. The newly added artefact thus has low novelty relative to the artefacts in the collection. Vice versa, an artefact of high novelty will increase the collection's diversity by a large amount because it does not resemble the other artefacts in the collection.

Similarity

*Similarity* quantifies the likeness in relevant qualities of two artefacts in a one-to-one relation. Similarity can also be quantified for a one-to-many relation between an individual artefact $a$ and a collection of artefacts $B$ or a many-to-many relation between two collections $A$ and $B$. In these cases, a collection will be represented by the qualities of an individual artefact from the collection, existing or non-existing, e. g. the artefact $b \in B$ with minimum or maximum similarity to the single artefact $a$, the average of all artefacts or, as a prototypical example, the artefact closest to the average or the centroid of the collection. The similarity of a collection is thus still determined through the artefacts it contains. Similarity is the basic relation upon which the more complex relations of novelty and diversity are established (Figure 1.2). That is to say, to measure novelty and diversity, essential properties for creativity, we need to estimate similarity.

The similarity, diversity and novelty of an artefact alone, however, are not enough to guarantee artefact typicality (Ritchie, 2007), i. e. the likeness of an artefact to the typical characteristics of a domain. For example, when we are trying to generate poems, we might expect the text to follow a certain type of register, rhyming scheme, or language. Similarly, we would expect

images of human faces to follow the conventional composition of eyes, noses and mouths. In generative machine learning, these domain requirements are usually enforced by the conventional learning objective (*distribution fitting*), which ensures that the model distribution approximates a training data distribution. Yet, as mentioned before, this objective alone emphasises artefact typicality. Increasing the output diversity of a model requires extending this objective.

Apart from the visual arts, generative models are used in many other applications, e. g. in video games for the generation of levels and other assets (Volz et al., 2018). In robotics, generative models provide a compressed search space over behaviour repertoires which allows for the optimisation of sequences of movement (Cully & Demiris, 2018a). For the synthesis of biologically active small molecules, generative models learn common molecule structures from data and thus provide a compact but expressive generative space to produce candidate molecules (Shin et al., 2021). Architects and industrial designers benefit from generative models as creativity support tools to survey the possibility space of a design problem (Bradner, Iorio & Davis, 2014). Crucially, rather than being the final product, generated artefacts serve as a starting point for further design iterations.

As multi-solution tasks, these applications benefit from several candidates that cover the full range of possibilities, rather than a single optimised solution. What use would it have if we could generate many variations of the perfect video game level but only with barely noticeable differences? Ideally, we would like to be able to adjust the output diversity of a generative model to the requirements of a given application. While it is easy to reduce diversity, in order to *increase* the output diversity, an approach has to overcome the inherent limitations of conventional generative modelling.

In each chapter of this thesis, we address one of the following research questions which focus on the different aspects of diversity in generative machine learning for visual arts and video games.

Research questions

RQ 1: How can generative models support creative applications?
→ Chapter 3

RQ 2: How are the conventional generative modelling approaches limited in terms of output diversity? → Chapter 4

RQ 3: How can the output diversity of generative models be increased? → Chapter 5

RQ 4: How can the measurement of diversity in generative machine learning be aligned with the human perception of diversity? → Chapter 6

Chapter 2 Background

The following is a brief overview of each chapter. In Chapter 2, we give an introduction to generative modelling in deep learning, the most popular approaches used in this thesis and their optimisation objectives, to support our case for higher output diversity. Conventional methods in generative machine learning are primarily concerned with faithfully capturing a given data distribution. While this may be useful for some downstream applications, we argue that this narrow objective, lacking considerations for diversity, is limiting for creative and artistic work, as well as detrimental to algorithmic fairness and equitable representation. We further discuss the conventional approaches to evaluating generative models.

Chapter 3 Artistic and Creative Uses of Generative Models

We further set the scene for this thesis in Chapter 3 by analysing the application of generative models to art production and creative tasks. Our findings support RQ 1 and motivate our work through the specific needs that arise from these use cases. Objectives in artistic and creative contexts differ from conventional applications in two ways. First, in this setting, learning from data is necessary to achieve high artefact *typicality*. However, rather than reproducing artefacts similar to the training examples, creative tasks often require the synthesis of *novel* artefacts. Second, instead of looking for a single optimised 'solution', people often seek to generate a variety of different candidate artefacts from which they can select for further design iterations, thus surveying the design space (Bradner, Iorio & Davis, 2014). The objective of using generative models in artistic and creative settings is thus to (1) facilitate the artefact synthesis in domains that are otherwise difficult to define manually, (2) generate a diverse set of artefacts and (3) enable the production of novel artefacts.

Our motivation by use-cases demonstrates this thesis as an example of *use-inspired basic research* (Stokes, 2011; Dudley, 2013). That is to say, we perform *fundamental research* on the algorithms that optimise and evaluate generative machine learning models with *consideration for their use* in creative applications. In this context, we critique the conventional development of models, analyse their limitations regarding output diversity and make contributions that result in more useful models for downstream tasks that demand diverse output.

Use-inspired basic research

Conventional statistical data-driven models depend largely on the characteristics of a given dataset. The variety of artefacts that a trained model can produce is dependent on the number of different training examples. In Chapter 4, we present a principled approach to evaluating the *expressivity* of generative models, i. e. the ability to produce a wide range of different types of artefacts. For this, we compare the performance of quality diversity search in a generative model's latent space against the baseline parametric design space. We find that the learned latent space yields artefacts of lower diversity than the corresponding manually-defined parametric space. Our findings contribute to RQ 2 and to understanding the limitations of generative models in terms of output diversity, justifying the following efforts to increase the output diversity of generative models.

Chapter 4 Limitations of Conventional Generative Modelling

In Chapter 5, we frame the limitation in output diversity of a generative model as a problem of data bias, where the likelihood under the model of a type of artefact is proportional to its prevalence in the training data. That is to say, the more often a type of artefact appears in a dataset, the more likely this type is going to be generated. We thus deal with a *data imbalance* bias.

Chapter 5 Increasing the Output Diversity of Generative Models

While there is no uniformly accepted way of dealing with data bias in generative machine learning more generally, and CC specifically, two fundamentally different approaches can be distinguished by their target of intervention, proposing to either (1) fix the data or (2) adjust the learning algorithm. We discuss both in the following paragraph.

Data bias in generative machine learning

In the first approach, data bias is addressed by improving the quality of a dataset, typically by gathering more and better data or by carefully curating the existing dataset and removing low-quality examples. However, two is-

sues complicate this approach. First, it is not always possible or practical to augment a dataset since collecting, curating and pre-processing new data is notoriously laborious, costly, or subject to limited access. Second, to address a specific bias, we require knowledge about the type of data examples that can reduce the bias. For example, even when a dataset can be augmented with synthetic data, the characteristics of the specific bias have to be identified and described, e. g. via text prompts for sampling from a text-to-image model (Chang et al., 2023). This is easier to achieve for supervised datasets with clear class separations, but can still become infeasible when fixing a bias that affects a single class implies adding examples to all other classes (see the discussion on related data bias work in Section 7.1). In unsupervised cases and generative machine learning in particular, due to a lack of clearly separable classes, inter-dependencies between examples become even more complicated, requiring more sophisticated approaches. Instead, the alternative is to adjust the methodology of learning from data such that a known data bias is mitigated. In this thesis, we focus on this strategy, addressing the representation imbalances between a dataset's majority and minority features by intervening in the learning process. We propose an algorithm that determines the individual contribution of training examples to the overall diversity of the dataset, thus increasing a model's output diversity through a diversity-weighted training scheme. Our work builds on the Vendi Score family of diversity measures (Friedman & Dieng, 2023; Pasarkar & Dieng, 2024). In a proof-of-concept study, we show the effectiveness of this method. Our results highlight a trade-off between artefact typicality and diversity. We contribute to RQ 3 by demonstrating how the output diversity of generative models can be increased.

*Diversity weights* method

For most artificial intelligence and machine learning applications to be useful to people, they need to match human expectations and cognition, and in the image domain in particular human visual perception. We support RQ 4 and the grounding of generative diversity in human perception in the following way. As laid out above, we identify similarity as a foundational relation between artefacts upon which more complex relations such as novelty and diversity are built. In particular, the Vendi Score measure

Chapter 6 Similarity Estimation for the Evaluation of Diversity

used in our *diversity weights* method requires estimating the similarity of artefacts in a collection to quantify its diversity. We take a step towards a better understanding of human similarity perception and how well it can be approximated by computational measures of similarity. We focus on the perception of similarity in tile-based video game levels. In two human participant studies, we collect data that allows us to compare computational approximations of similarity estimation to the human perception of similarity in this specific domain.

Work that is *related* to our contributions and thematically relevant to the main themes of the thesis is discussed toward the end of the thesis in a dedicated Chapter 7. This includes an overview of diversity measures, common data biases in machine learning and de-biasing methods to address these shortcomings. Other works relevant to the methodologies of our contributions are discussed within the individual chapters.

Chapter 7
Related Work

We conclude the thesis with a summary of the contributions in Chapter 8, covering the generated knowledge, proposed methodologies and research artefacts such as code and data repositories. We further discuss future avenues of research.

Chapter 8
Conclusions

# Chapter 2

# BACKGROUND

In this chapter, we provide all the necessary background information to contextualise the work performed and understand the methods used in the present research. For this, we first give an introduction to generative modelling in deep learning, the most popular approaches and their objectives. We next discuss how frameworks for creative systems apply to generative modelling. We then discuss the conventional approaches to evaluating generative models, including approaches to measuring output diversity. Finally, we review the literature on the human perception of similarity, a fundamental part of the human-centred evaluation of diversity.

We motivate the research presented in this thesis by critiquing the field's canonical practice and highlighting important shortcomings. First, none of the evaluation measures is suited for independent assessment, i. e. they require the training dataset for reference. Second, the predominant modelling paradigm is primarily concerned with faithfully capturing a given data distribution. While this may be useful for some downstream applications, we argue in this thesis that this narrow objective, lacking considerations for diversity, is limiting for creative and artistic work, as well as detrimental to algorithmic fairness and equitable representation. Crucially, at the time when research presented in this thesis began (late 2019), there was little to no consideration for objective diversity in generative machine learning. Only more recently, some progress has been made in related work (Section 2.5) with the proposals of dataset-independent measures of diversity and modifications of training algorithms to de-bias generative models (Section 7.2).

CONTENTS

Terminology

Following the statistical terminology, we use *sample* to denote a single data point taken to estimate the characteristics of a larger population. In the context of this thesis, we are primarily concerned with the output of a generative model. We thus take samples from a model to estimate the modelled distribution. In contrast, an *example* is a single data point from a dataset used to optimise a model's parameters with a learning algorithm. Using these two terms, we delineate what is an *output* from a model (sample) and what is an *input* to a model (example). The term *artefact* conventionally describes a human-made object that provides insight into the culture of its creators and users. We extend the term here to include objects that are machine-made for the benefit of humans. Arguably, if humans are involved in the production of the machine, the objects made by the machine may still reflect the culture of the human machine-creators. The output of a generative model can thus simultaneously be a sample and an artefact.

Notation

In the equations of this chapter, we employ a common notation. The natural exponential function $\exp(x) = e^x$ is not expanded for readability. Unless otherwise noted, we use the natural logarithm log with base $e$. $\mathbb{E}[X]$ denotes the expected value of $X$. $\nabla_x$ stands for a function's gradient with respect to $x$. We define the standard normal distribution $\mathcal{N}(0, \mathrm{I})$ for the general multi-dimensional case where I is the identity matrix indicating no correlation between the independent variables. $P$ and $Q$ refer to the probability distributions of the real data and the generated data respectively. The Kullback-Leibler divergence $D_{\mathrm{KL}}$ from $Q$ to $P$ is defined as the expectation of the logarithmic difference between the probabilities $P$ and $Q$, where the expectation is taken using the probabilities $P$.

$$D_{\mathrm{KL}}\big(P \parallel Q\big) = \sum_{x \in X} P(x) \, \log \frac{P(x)}{Q(x)} \tag{2.1}$$

## 2.1 GENERATIVE MODELLING IN DEEP LEARNING

The purpose of a *generative model* is to approximate the probability distribution $p(X)$ of the data $X$ or, in the conditional case, the joint probability

$p(X, Y)$ of data $X$ and labels $Y$. That is to say, we want to find the distribution, our model, that best describes the given dataset. In contrast, a *discriminative model* only captures a conditional probability $p(Y|X)$ over labels $Y$ given the data $X$.

Some natural observations (e. g. the height of a group of people) can be modelled with simple theoretical distributions (e. g. Gaussian). But these are not flexible enough to approximate the distribution of more complex domains like natural and synthetic images. In particular, because the random variables are not independent. Here we will thus focus on deep learning techniques that use an artificial neural network to parameterise a probability distribution $p_\theta(X)$, and typically apply gradient-based methods such as back-propagation to optimise the parameters $\theta$. The *likelihood* $\mathrm{LLH}(x, \theta)$ is the joint probability of the observed data $x \in X$ given the model parameters $\theta$. Maximum likelihood estimation (MLE) evaluates the likelihood function for a specific configuration of the model parameters.

We can view a generative model and interpret its utility from different perspectives. Conventionally, it is seen as a way to approximate a given target distribution and its quality corresponds to the distance between the model and the target distributions. Generative modelling from raw data can be related to compression. To find regularities in the data, separating signal from noise, it is necessary to identify and understand the underlying *patterns*. The ability to describe patterns more compactly allows for *compression*, describing the process that generates the observed patterns. The shortest possible description, in terms of a computer program, that can generate the raw data defines the data's *Kolmogorov complexity*. While the artificial neural network used in a generative model is often over-parameterised and thus likely far from the shortest description, a model's latent encoding space can be seen and used as a compressed representation for the training examples which preserves their semantic relations. However, most generative models are incomplete compression algorithms with only a decoder (generator), lacking an encoder. From the point of view of search, a generative model enables interaction with the examples from the training dataset in an indirect and augmented way. We can find artefacts that resemble data examples,

recombining common features through interpolation and extrapolation. In the context of CC therefore, a generative model is a very capable *generator*, giving access to domains and types of artefacts that would otherwise be difficult to re-create.

Early deep generative models, most prominently Deep Boltzmann Machines (DBM) (Salakhutdinov & Hinton, 2009), provide a parametric specification of a probability density function and are trained by maximising the log-likelihood of the data. For many practical applications, however, model parameters and latent variables are high-dimensional. Their summation or integration and thus the calculation of the exact marginal likelihood is often computationally intractable. The limitations of Boltzmann machines motivated the development of GANs, a type of *implicit* generative model, that produce artefacts of high fidelity without an explicit likelihood representation. Developed almost in parallel, VAEs implement an auto-encoder-based approach and use variational Bayesian methods to approximate the intractable integrals, maximising the evidence lower bound on the marginal likelihood. These advancements proved to be crucial in demonstrating the potential of generative modelling. Other techniques, e. g. vision transformers (Esser, Rombach & Ommer, 2021; Dosovitskiy et al., 2022) and diffusion models (Dhariwal & Nichol, 2021; Rombach et al., 2022) have since continued to push the boundaries of generative capabilities. But we focus here on the two aforementioned modelling approaches, GANs and VAEs, for their wide adaptation and to limit the scope to methods used in the work presented in the following chapters.

### 2.1.1 GENERATIVE ADVERSARIAL NETWORKS

Adversarily trained generative models, more commonly known as generative adversarial networks (GANs), consist of two networks: a generator and a discriminator or critic. As *implicit* generative models, GANs do not explicitly model the likelihood function. Instead, the generator is trained to produce samples that resemble examples from the training dataset, ef-

fectively mapping from a latent space to the feature space $G : \mathcal{Z} \to \mathcal{X}$. The discriminator is simultaneously trained to distinguish generated samples from training examples, mapping from feature space to an output probability $D : \mathcal{X} \to \mathbb{R}^{[0,1]}$. As a result, GANs do not provide an encoding function that maps from feature space to latent space, making it difficult to recover the latent encoding of a data point.

The generator network is defined as a differentiable function $G(z; \theta_G) = \tilde{x}$ with parameters $\theta_G$ that learns to transform a sample from a known noise distribution $z \sim p_Z$, typically a standard normal $\mathcal{N}(0, \mathrm{I})$, to the target distribution. Given both data examples drawn from the training dataset $x \sim p$ and samples produced by the generator, the discriminating network $D(x; \theta_D) = \tilde{y}$ with parameters $\theta_D$ transforms a given input into a Bernoulli distribution, a discrete probability distribution with Boolean-valued outcomes that indicates with probability $\tilde{y}$ that the input $x$ belongs to the training dataset. During training, we use the target labels $y = 0$ for generated samples and $y = 1$ for training examples. The discriminator is trained to maximise the probability of assigning the correct labels. Simultaneously, the generator is trained to maximally confuse the discriminator, such that it equally assigns a probability of $D(x) = 0.5$ to both training examples and generated samples. The full GAN objective is given below.

$$\min_G \max_D \mathop{\mathbb{E}}_{x \sim p} \big[ \log(D(x)) \big] + \mathop{\mathbb{E}}_{z \sim p_Z} \big[ \log(1 - D(G(z))) \big] \tag{2.2}$$ Objective

Note the similarities to the definition of the Bernoulli distribution. The GAN objective is the sum in log space (product in linear space) of the expectation that data examples $x$ are classified as coming from the dataset and the expectation that generated samples $\tilde{x} = G(z; \theta_G)$ are synthetic.

Due to the adversarial setup of the generator and discriminator networks and the minimax training objective above, the GAN framework is often framed in game theoretic terms as a zero-sum game between two opponents, where one's gain is another's loss. Since there is no explicit density estimation of the dataset, the generator depends entirely on the gradient signal from the discriminator for optimisation. However, at the beginning of training,

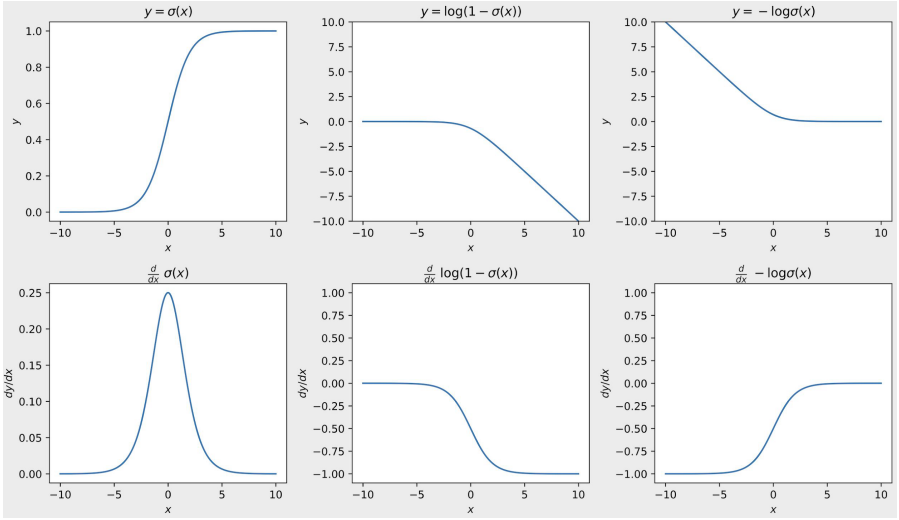



Figure 2.1: Three illustrative functions (top row) and their first derivatives w.r.t. the input (bottom row). Left: sigmoid activation function of the discriminator's final network layer. At the beginning of training, the discriminator assigns negative values (x-axis top left subgraph) to samples produced by the generator, as they are easily distinguishable from training examples, thus yielding a low output probability (y-axis top left subgraph). Middle: saturating discriminator loss. For negative input values, the output is very close to or almost zero, providing very small gradient updates for the generator. Right: non-saturating discriminator loss. By changing the optimisation objective, the discriminator provides a better gradient update signal to the generator for the early stages of training.

the generator produces mostly noise, easily distinguishable from training examples. The discriminator will confidently assign a very low probability to generated samples and thus, due to the second term of the objective ($\mathbb{E}_{z\sim p_Z}[\log(1 - D(G(z)))]$), provide only very small, almost non-existent gradient update signals to the generator. See Figure 2.1 for a visualisation of the relevant functions. Therefore, in practice, training typically consists of minimising a non-saturating loss formulation for the generator.

$$\underset{D}{\mathcal{L}} = - \underset{x\sim p}{\mathbb{E}} \left[\log(D(x))\right] - \underset{z\sim p_Z}{\mathbb{E}} \left[\log(1 - D(G(z)))\right] \tag{2.3}$$

Non-saturating loss

$$\underset{G}{\mathcal{L}} = - \underset{z\sim p_Z}{\mathbb{E}} \left[\log(D(G(z)))\right] \tag{2.4}$$

As the generator loss function is not the exact inverse of the discriminator loss $L_G \neq -L_D$, this is no longer a zero-sum game in the strict sense.

GANs have notoriously unstable training since the two networks, generator and discriminator, are optimised simultaneously and the generator in

particular depends on the gradient signal from the discriminator for weight updates. Several works propose small changes and improvements to stabilise the training procedure (Salimans et al., 2016; Arjovsky & Bottou, 2017; Heusel et al., 2017). This includes, in particular, adaptations of conventional image classifiers to be used as convolutional GANs for representation learning (DCGAN, Radford, Metz & Chintala, 2016). Further architectural improvements consist of adding self-attention layers to the generator and discriminator (SAGAN, H. Zhang et al., 2019), as well as scaling up the developed building blocks for large-scale training (BigGAN, Brock, Donahue & Simonyan, 2019). Progressively growing GANs (Karras et al., 2018) propose an architecture that starts training at low resolution ($4 \times 4$ pixels), incrementally adding layers to the generator and discriminator during training that double the spatial resolution up to $1024 \times 1024$.

StyleGAN

The StyleGAN class of models adopts the progressively growing architecture as well as techniques from style transfer, like adaptive instance normalisation, for a re-designed style-based generator architecture (Karras, Laine & Aila, 2019). Instead of generating an image directly from a latent code sample $z \in \mathcal{Z}$, a fully connected mapping network $f(z) = w$ transforms it to an intermediate latent space $w \in \mathcal{W}$. This latent code is then injected into the generator after each convolutional layer as a *style vector*, conditioning the synthesis of the target image. Follow-up work by the same authors fixes small image artefacts (Karras, Laine et al., 2020) and adds image augmentation for training with limited data (Karras, Aittala et al., 2020).

### WASSERSTEIN GANS

The objective of Wasserstein GANs (WGANs) is to minimise the Wasserstein distance or earth mover's distance, between the data distribution and implicit model distribution (Arjovsky, Chintala & Bottou, 2017; Gulrajani et al., 2017). The authors argue that the training instabilities of the original GAN objective arise due to the divergence measure between the two distributions in other objectives potentially not being continuous with respect

to the generator parameters. Therefore, they propose to enforce *Lipschitz continuity* on the discriminator function. Such a function $D : \mathbb{R}^n \to \mathbb{R}$ is *K-Lipschitz* if for a distance function $d$ on $\mathbb{R}$ the distance in the projected space is smaller than or equal to $K$ times the distance in the original space. $K$ is then referred to as the *Lipschitz constant*.

Lipschitz continuity

$$\forall x, y \in \mathbb{R}^n, \quad d\big(D(x), D(y)\big) \leq K\, d\big(x, y\big) \tag{2.5}$$

For this, the authors relax the output space to cover the complete space of real numbers $\mathbb{R}$. Rather than a discriminator network, WGANs thus make use of a *critic*. While a discriminator is optimised to predict the correct target labels for training examples and generated samples, the critic is simply trained to maximise the difference between its predictions for the two types of inputs. Using the Kantorovich-Rubinstein duality (Villani et al., 2009), the objective is changed to minimising the Wasserstein-1 distance between the data and the model distributions, implicitly defined by the generated samples $\tilde{x} = G(z)$. Here, the critic function $D$ is 1-Lipschitz continuous, meaning the norm of its gradient at no point is larger than 1.

$$\min_G \max_D \mathop{\mathbb{E}}_{x\sim p} \big[\log(D(x))\big] - \mathop{\mathbb{E}}_{z\sim p_Z} \big[\log(D(G(z)))\big] \tag{2.6}$$

The WGAN formulation has the benefit of better robustness to choices in network architecture, whereas DCGAN (Radford, Metz & Chintala, 2016), for example, fails without batch norm layers.

There are several approaches to ensuring 1-Lipschitz continuity, which give rise to the two WGAN formulations: *weight clipping* and *gradient penalty*. We can simply apply *weight clipping*, constraining the network weights to a fixed range after each optimisation step (Arjovsky, Chintala & Bottou, 2017). Alternatively, we can add a *gradient penalty* loss term to the objective function (WGAN-GP, Gulrajani et al., 2017).

Weight clipping

Gradient penalty

$$\begin{aligned}\mathcal{L} = &\mathop{\mathbb{E}}_{x\sim p} \big[\log(D(x))\big] - \mathop{\mathbb{E}}_{z\sim p_Z} \big[\log\big(D(G(z))\big)\big] \\ &+ \lambda \mathop{\mathbb{E}}_{\hat{x}\sim p_{\hat{x}}} \big[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2\big]\end{aligned} \tag{2.7}$$

The gradient penalty formulation has two drawbacks when compared to gradient clipping. First, this approach is computationally more expensive, as it requires an additional backward pass to obtain the gradients for the penalty loss term. Second, it adds another hyper-parameter ($\lambda$), which might require tuning for best results.

#### SPECTRAL NORMALISATION GANS

A more principled way of enforcing 1-Lipschitz continuity, without weight clipping or gradient penalty, is through spectral normalisation (SNGAN, Miyato et al., 2018). In the linear transform of a single layer $g(h) = Wh$ for an input $h$, the Lipschitz constant is equal to the largest singular value of the weight matrix $W$. Thus, normalising the weight matrix of every layer by its largest singular value is a simple and cost-effective approach to ensuring Lipschitz continuity of the composite function. Moreover, the authors find that for the calculation of the largest singular value through power iterations, one single iteration yields a sufficiently accurate approximation. Spectral Normalisation GANs implement a simpler approach to ensuring Lipschitz continuity and have a computational benefit over WGANs with gradient clipping as they do not require a backward pass to compute the gradient.

In contrast to the conventional GAN objective, where generator and discriminator updates have to be carefully balanced, the WGAN critic should ideally be trained until convergence at each optimisation step to provide the highest-quality gradient signal to the generator. Since this might be too inefficient for most cases, the authors recommend five critic updates for every generator update, and if necessary to increase the number of critic iterations for harder problems.

### 2.1.2 VARIATIONAL AUTO-ENCODERS

The motivation for latent variable models, such as variational auto-encoders (VAEs), is to find a simpler, lower-dimensional yet semantically meaningful representation for the examples in a given dataset. By maximising the

likelihood of the data, the latent variables ideally recover the independent factors of variation underlying the generative process which produced the data. The VAE objective is to maximise the log-likelihood of the data.

Objective

$$\max_{\theta} \sum_{i} \log\, p_\theta(x_i) = \sum_{i} \log \sum_{z} p_Z(z)\, p_\theta(x_i|z) \tag{2.8}$$

For a discrete random variable $z$ with a small domain, the solution to this objective can be calculated exactly. However, when $z$ can take on a very large amount of possible values, i. e. in the case of continuous random variables, the objective becomes intractable and can only be approximated. The key idea of VAEs is to approximate the true posterior distribution $p(z|x)$ with a simple, tractable distribution $q_\phi(z|x)$ via an inference network. The conditional probabilities $p_\theta(x|z)$ and $q_\phi(z|x)$ are parameterised by artificial neural networks with parameters $\theta$ and $\phi$, respectively. VAEs thus follow the standard architecture of auto-encoders: an encoding recognition network $E : \mathcal{X} \to \mathcal{Z}$ maps from feature space to latent space, and a decoding generative network $D : \mathcal{Z} \to \mathcal{X}$ maps back to the feature space.

Instead of the intractable marginal likelihood, we maximise an approximate variational lower bound (first right-hand-side term), or evidence lower bound (ELBO), which is guaranteed to approach the maximum likelihood from below everywhere. That is to say, we know that the likelihood of the data under the model is *at least as high* as the approximate lower bound and potentially higher, which is what we want to achieve. Simultaneously, we minimise the Kullback-Leibler (KL) divergence between the approximate posterior and the prior distributions (second term). Since the KL divergence is always bigger than or equal to zero, it accounts for the difference between the variational lower bound and the log-likelihood.

Approximate variational lower-bound

$$\begin{aligned} \log p(x) = &\mathop{\mathbb{E}}_{z\sim q_x(z)} \big[ -\log q_x(z) + \log p(z) + \log p(x|z) \big] \\ &+ D_{\mathrm{KL}}\big(q_x(z) \,\|\, p(z|x)\big) \end{aligned} \tag{2.9}$$

There are several approximation approaches. We focus here on the most popular, the *reparameterisation trick* (Kingma & Welling, 2014). Each training

Reparameterisation trick

example is passed through the encoding network which yields the parameters of the Gaussian (means $\mu$ and variances $\sigma^2$) that define the approximate latent distribution. We then sample from a normally-distributed auxiliary noise variable $\epsilon \sim \mathcal{N}(0, \mathrm{I})$, where I is the identity matrix indicating no correlation between the independent variables in the multi-dimensional case, and add it to the parameters of the latent distribution to obtain a perturbed encoding $z = \mu + \epsilon \cdot \sigma^2$, which is independent of the network parameters $\theta$ and $\phi$. This step is repeated $L$ times to obtain a better approximation. However, in practice, one sample is often sufficient since the stochastic gradient descent setup already involves repeated sampling over many training examples and iterations. For reconstruction, the samples $z_i$ are passed through the decoding generator to reconstruct the input and evaluate the ELBO. While we have looked at VAEs as a latent variable generative model, the connection to auto-encoders becomes evident in the following equation.

$$\log p_\theta(x) \geq \mathop{\mathbb{E}}_{z \sim q_x(z)} \big[\log p_\theta(x|z)\big] - D_{\mathrm{KL}}\big(q_\phi(z|x) \,\|\, p(z)\big) \tag{2.10}$$

The first right-hand-side term is effectively a reconstruction loss that quantifies the likelihood of $x$ given $z$. That is to say, how well does the generator with the current parameters $\theta$ generate an artefact from the latent encoding? The second term, the KL divergence between the latent distribution and a standard normal distribution, serves as a regularisation loss term forcing the latent distribution to move closer to a simple pre-defined distribution, like a Gaussian. In order to not simply memorise the data, we should not allow the latent distribution to take any complex form. Regularisation is one way to achieve this. The VAE loss function reflects the same loss terms (for simplicity, for a single training example $x$).

$$\begin{aligned} \mathcal{L}(x) &= \frac{1}{L}\sum_{i=1}^{L} \log p(x|z_i) + \frac{1}{2}\sum_{j=1}^{M}\big(1 + \log\sigma_j^2 - \mu_j^2 + \sigma_j^2\big) \\ z &= \mu_j + \epsilon \cdot \sigma_j^2 \qquad \epsilon \sim \mathcal{N}(0, \mathrm{I}) \\ \mu_j &= \mu_j(x, \phi) \qquad \sigma_j = \sigma_j(x, \phi) \end{aligned} \tag{2.11}$$

Loss functions

There exist several functions to evaluate the reconstruction loss. Following the practice of auto-encoders, a common choice is a mean-squared error. Better results are typically achieved using binary cross entropy. But both choices make the model non-probabilistic. Theoretically better suited is the continuous Bernoulli distribution (Loaiza-Ganem & Cunningham, 2019).

As in the above explanation, the prior of the latent distribution is typically a Gaussian, the highest-entropy distribution for continuous variables with known variance. Alternatively, the distribution of the latent variables can be specifically designed for a dataset with known factors of variation. For example, in the case of the MNIST dataset (LeCun, Cortes & Burges, 2010), we know that we deal with images of ten hand-written digits. We could thus specify ten dimensions of the latent space with a Bernoulli prior that are reserved to assign data examples to digit classes via a one-hot encoding, while all other dimensions have a Gaussian prior and remain available to capture other continuous factors of variation, e. g. stroke thickness or rotation.

### LATENT DISENTANGLEMENT: $\beta$-VAE

Standard VAEs are not guaranteed to capture linearly separable factors of variation from the data in individual latent variables. Beta-VAEs (Higgins et al., 2016) aim to improve latent disentanglement by adding the hyper-parameter $\beta$ to the objective. Bigger values for $\beta$ put more emphasis on the encoding distribution mapping to a standard normal distribution, whose variables are independent and identically distributed. When $\beta = 1$, Beta-VAE is identical to the original VAE.

$$\begin{aligned} \mathcal{L} &= \mathop{\mathbb{E}}_{q_\phi(z|x)} \big[\log p_\theta(x|z)\big] - \beta\, D_{\mathrm{KL}}\big(q_\phi(z|x) \,\|\, p(z)\big) \\ &\text{where} \quad p(z) = \mathcal{N}(0, \mathrm{I}) \quad \beta \geq 1 \end{aligned} \tag{2.12}$$

The closer the latent distribution is to a Gaussian, the less dependence there is between individual dimensions.

Further work on understanding disentanglement extends the Beta-VAE objective by a capacity control hyperparameter $\gamma$ (Burgess et al., 2017). This

objective minimises the absolute deviation of the KL divergence from the capacity control $\gamma$.

$$\begin{aligned} \mathcal{L} &= \mathop{\mathbb{E}}_{q_\phi(z|x)} \big[\log p_\theta(x|z)\big] - \beta \; | \, D_{\mathrm{KL}}\big(q_\phi(z|x) \,\|\, p(z)\big) - \gamma \, | \\ &\text{where} \quad p(z) = \mathcal{N}(0, \mathrm{I}) \quad \beta \geq 1 \quad \gamma \geq 1 \end{aligned} \tag{2.13}$$

Following the information bottleneck principle, the capacity control is measured in *natural units of information* (nat), which quantifies the information content of observations (here: examples from the training dataset). An observation with the probability $1/e$ has the information content of one nat. The authors recommend increasing the hyperparameter $\gamma$ during training to gradually give more capacity of the encoding to additional factors of variation while maintaining the disentanglement of previously learned factors.

## 2.2 GENERATIVE MODELS AS CREATIVE SYSTEMS

In her book on human and computer creativity, Boden (2004) describes three forms of creativity: *combinational*, *exploratory* and *transformational* creativity. Here, we will focus on the latter two. The *creative system framework* (Wiggins, 2006a; 2006b) builds on Boden's work and formalises the ideas of exploratory and transformational creativity, describing the computational components necessary to implement a creative system.

The creative system framework

Within the framework, a *conceptual space* is defined as a set of artefacts that a system could conceivably produce.

The notion of a *conceptual space* is easily applied to generative models. In the context of generative modelling, a model's latent space can be understood as a *conceptual space*. The latent variables are used as an abstract representation of artefacts in a dataset. Through the training process, they are placed in semantically meaningful relations with each other. The properties of the latent space, the model's conceptual space, are thus defined by the examples in a training dataset. In particular, one can easily imagine that there is a dataset of artefacts which does not contain all possible examples of a type of

artefact. For example, a dataset of chairs may not include any chairs without a backrest. A model trained on this dataset will provide a conceptual space that does not cover all possible instances of the artefact, e. g. chairs without a backrest. A model can be transformed to extend the conceptual space it provides: to follow our example, by augmenting the training dataset with examples of stools.

Exploratory and transformational creativity

In *exploratory creativity*, new artefacts are identified and/or located by traversing the conceptual space with a search strategy. That is to say, artefacts are *found* through exploration rather than *created* through deliberate construction. Consequently, the properties of a conceptual space determine which artefacts can be conceived or not. *Transformational creativity* takes one step further to a meta-level of creativity. By changing the properties of a conceptual space, the conditions that enable the identification and/or localisation of artefacts through exploration can be *transformed*. The variety of different conceptual spaces can be seen as artefacts in a conceptual space of conceptual spaces which itself can be explored. In the framework, transformational creativity is thus understood as exploratory creativity at the meta-level.

## 2.3 USE OF GENERATIVE MODELS WITH EVOLUTIONARY ALGORITHMS

Previous work employed auto-encoders for dimensionality reduction and its latent representations as encodings of behavioural descriptors in control tasks (Meyerson, Lehman & Miikkulainen, 2016; Cully, 2019). In shape optimisation, the latent spaces of generative models have been used to distinguish parameterised representations (Hagg, Preuss et al., 2020; 'A Deep Dive Into Exploring the Preference Hypervolume', n.d.). In robotics, this approach allows robots to autonomously discover the range of their capabilities without prior knowledge (Cully, 2019). Generative models have also been employed to automatically learn an encoding during optimisation, using them as a variational operator (Gaier, Asteroth & Mouret, 2020).

GANs have been used in latent variable evolution (Bontrager et al., 2018) to generate levels for the video games Super Mario Bros (Volz et al., 2018) and Doom (Giacomello, Lanzi & Loiacono, 2019). These approaches are illustrative of *exploratory creativity* (Section 2.2). A model's latent space is searched with an evolutionary algorithm for instances that optimise for desired properties such as the layout or difficulty of a level. While some authors view the generated levels as novel, none have measured exactly how novel or diverse of an output such a system can produce.

To the best of our knowledge, before our principled study (Chapter 4), there existed no evidence of the benefits and drawbacks of using generative models for phenotypic encoding in evolutionary algorithms.

## 2.4 EVALUATION OF GENERATIVE MODELS

This section provides an overview of the conventional approach to evaluating the performance of generative models. We focus here on the evaluation of image-generating models. Different specialised measures may be better suited to assess models that produce other types of artefacts, e. g. audio or text.

We first discuss the use of image embeddings to facilitate the semantic rather than pixel-wise comparison of images. We then guide the reader through the evolution of performance measures specific to generative models, discussing individual limitations and how they were addressed by subsequent work. Crucially, at the beginning of the work presented in this thesis (late 2019), the evaluation of generative algorithms in machine learning was primarily concerned with the fidelity and mode coverage of the generated artefacts. There was little to no consideration for the diversity of examples in a dataset or collection of generated output. And none of the measures discussed in this section are suited to objectively assess this. Only more recently, some progress has been made with the proposal of the Vendi Score (VS), a dataset-independent measure of diversity (Section 2.5).

### 2.4.1 IMAGE EMBEDDINGS

Instead of comparing image data on raw pixels, standard evaluation measures of model performance have relied on image classification networks to be used as embedding models for feature extraction. The Inception model (Szegedy et al., 2016) is most commonly used as a representative feature space and has been widely adopted as part of a standard measurement pipeline, most prominently lending its name to the IS and FID measures (see below for details). There are alternative models, like the VGG models (Simonyan & Zisserman, 2015). The majority of older computer vision models, like Inception and VGG, have been pre-trained on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset (Russakovsky et al., 2015) and are thus limited in three important aspects. First, the ImageNet dataset consists of natural images from 1,000 very specific classes, e. g. zebra, umbrella and forklift. These classes were selected to pose a challenging classification problem in computer vision research. For this reason, the ImageNet classes also include several very similar breeds of dogs. Second, as a result, the models are optimised to capture features relevant to only this limited set of classes. Third, they inherit the dataset's biases, which can lead to unreliable measurements of image qualities that do not agree with human assessment (Kynkäänniemi et al., 2023). Furthermore, between different computational frameworks like PyTorch, TensorFlow and JAX, small numerical differences in model weights, implementations and interpolation operations can compound into bigger discrepancies. For example, image scaling to match the input size of an embedding model can change the computed features and thus affect the subsequent measurements (Parmar, Zhang & Zhu, 2022). Measures like FID further assume that a model's embedding space is approximately Gaussian, which is not always guaranteed. It is therefore preferable to (1) follow the recommendations for anti-aliasing re-scaling and (2) to use a newer image embedding model, like CLIP (Radford et al., 2021), in a feature extraction and measurement pipeline.

### 2.4.2 PERFORMANCE MEASURES

Inception Score

The Inception Score (IS) inherits its name directly from the aforementioned Inception model and is based on the class prediction probabilities of its output layer (Salimans et al., 2016). The IS follows the principle that a good generative algorithm produces artefacts that are individually classified by a computer vision model with high confidence in a single class, while also producing a large variety of different artefacts. The former is quantified by the entropy over the conditional label distribution $p(y|x)$, where lower entropy is better. The latter is defined as the entropy of the marginal $p(y) = \int_z p(y|x = G(z))\, dz$ over a sufficiently large set of samples generated by a generator $G$ from a latent code $z$, where higher entropy is better. The difference between these two distributions is given by the Kullback-Leibler divergence $D_{\mathrm{KL}}$ (Equation 2.1).

$$\mathrm{IS}(Q) = \exp\Big( \mathop{\mathbb{E}}_{x,y\sim Q} \left[ D_{\mathrm{KL}}\big(p(y|x) \parallel p(y)\big) \right] \Big) \tag{2.14}$$

Since the class label distribution and the marginal distribution ideally diverge, a higher score indicates better performance in this regard. While the score's lowest value is 1, there is no theoretical upper limit. This lack of a precise range and the fact that the exponential function breaks linearity make the IS difficult to interpret and unsuitable for direct comparison. A model that achieves a score twice as high as another model is not necessarily twice as good. It can only be speculated whether the authors wanted to explicitly reward small improvements at high values (by analogy, it is notoriously difficult to raise the accuracy of a classifier from 97 to 98 % compared to the increase from 77 to 78 %) since there is no explicit explanation in the original proposal.

Apart from the general drawbacks of using the Inception model which is discussed in the previous section, there are some other specific limitations to this measure. The IS does not measure intra-class diversity. All generated images in one class could look identical or have only slight variations and would still achieve a high score if the evaluating model assigns them a

class with high confidence. Going further, if a model simply memorises and reproduces the training data it similarly would score highly. Several other approaches, discussed below, try to improve on the Inception Score.

Mode Score

The Mode Score (MS) addresses the problem of missing modes (Che et al., 2017), i. e. types of images of high probability in the training dataset that are not well represented in the model distribution. The authors propose an extension to the IS by replacing the marginal label distribution with the label distribution over the training data ($p(y^*)$), and by introducing an additional term that computes divergence from $p(y^*)$ to the label distribution over generated samples $p(y)$. The closer both distributions are, the higher the score.

$$\mathrm{MS}(Q) = \exp\left(\mathop{\mathbb{E}}_{x,y\sim Q}\left[D_{\mathrm{KL}}\big(p(y|x) \,\|\, p(y^*)\big)\right] - D_{\mathrm{KL}}\big(p(y) \,\|\, p(y^*)\big)\right) \quad (2.15)$$

Conceptually, this score links the evaluation of a model closer to the classes of the training data, which is a direct response to the problem of missing modes.

Fréchet Inception Distance

The Fréchet Inception Distance (FID) makes use of the Fréchet distance between two distributions, also known as Wasserstein-2 distance, to turn the IS from an unbounded score into a distance measure (Heusel et al., 2017). Features are extracted with an embedding model for the images from the training dataset and images generated by the model. Assuming the embedding space follows a Gaussian distribution, the distributions of the embeddings of training images $P$ and the generated images $Q$ are approximated by computing their means $\bar{x}$ and covariance matrices $S$. With these statistics, the FID measures the distance between the two multi-variate Gaussians.

$$\mathrm{FID}(P, Q) = \|\bar{x}_P - \bar{x}_Q\| + \mathrm{tr}\left(S_P + S_Q - 2(S_P\, S_Q)^{1/2}\right) \quad (2.16)$$

In comparison to the IS, the range of the score is turned around and a lower FID is considered better, indicating that the two Gaussian distributions are close to each other. Consequently, there is a precise lower bound and

optimal score, whereas the IS could in theory increase infinitely. This makes the FID a more easily interpretable distance measure. While it is capable of assessing intra-class diversity, this measure is still susceptible to being fooled by a model with perfect memory of the training data. Furthermore, the two aspects that the measure aims to assess, sample fidelity and mode coverage, are entangled in a single score.

Precision and Recall

The pair of measures Precision–Recall (PR) adapts two conventional evaluation concepts, *precision* and *recall*, to generative modelling (Sajjadi et al., 2018). Precision is the fraction of generated samples that are of high fidelity, i. e. having a sufficient resemblance to training examples. Recall is the fraction of training examples that can be generated by the model, i. e. they are covered by the modelled distribution. In an improved formalisation (Kynkäänniemi et al., 2019), this is determined with a binary membership function $f$, which indicates whether a sample $x$ is supported by a given data manifold $U$ by checking if it falls into the k-nearest neighbourhood ($\mathrm{NN}_k$) of any data point $u$.

$$f(x, U) = \begin{cases} 1, & \text{if} \quad \exists\, u \in U \quad \text{s.t.} \quad \|x - u\|_2 \leq \|u - \mathrm{NN}_k(u, U)\|_2 \\ 0, & \text{otherwise} \end{cases} \tag{2.17}$$

$$\begin{aligned} \text{Precision}(P, Q) &= \frac{1}{|Q|} \sum_{q \in Q} f(q, P) \\ \text{Recall}(P, Q) &= \frac{1}{|P|} \sum_{p \in P} f(p, Q) \end{aligned} \tag{2.18}$$

IS and FID use a one-dimensional measure to estimate a two-factorial evaluation between sample fidelity and mode coverage, which obfuscates their interaction. For example, a low FID (good performance) may be caused by high precision (samples of high fidelity), high recall (good mode coverage), or a combination of the two. In contrast, FID consists of two separate quantities that disentangle this relation and allow for a specific choice in

the trade-off between sample fidelity and mode coverage. This allows us to experimentally confirm that GANs produce 'sharper' images of higher fidelity but but can suffer from mode collapse (high precision, lower recall), while VAEs generate 'blurry' images but provide better mode coverage (lower precision, high recall). We discuss this trade-off between artefact fidelity and diversity in more detail in Chapter 5.

## 2.5 MEASURING OUTPUT DIVERSITY

Several measures have been proposed to evaluate diversity in different research fields with context-specific properties. In this section, we discuss the most widely used.

In quality diversity (QD), local neighbourhoods in the search space, so-called niches, can be used for performance evaluation. The *QD-score* (Pugh, Soros & Stanley, 2016) is defined as the sum of fitness of the highest-performing artefact in each niche. It combines the measure of individual fitness with the coverage of the search space: it is lower if some niches are completely unoccupied (adding zero fitness to the sum) or only occupied with low-quality individuals. A well-performing algorithm should be able to fill a large number of niches with high-quality candidates.

Quality Diversity-Score

In multi-modal optimisation (MMO), measures use *quality indicators* (Zitzler, Knowles & Thiele, 2008) to map a set of artefacts to a real number. Some indicators rely on problem knowledge and are primarily used for benchmarking, i. e. in test scenarios with complete information on global or local optima and their attraction basins (Preuss & Wessing, 2013). Such measures are thus not applicable to most real-world problems where the fitness landscape is not fully known. We focus here on indicators that are problem-agnostic and can be used in real-world applications where specific problem knowledge is not available a priori.

The *sum of distances* (SD) rewards the spread of artefacts but does not penalise duplicates. In an illustrative experiment, the optimal distribution of 100 points in a two-dimensional Euclidean space for the SD measure places

Sum of distances

all points in the four corners of the space, resulting in multiple duplicates of four maximally distant solutions (Ulrich, Bader & Thiele, 2010). Since SD does not explicitly take into account artefact diversity or quality, it is therefore considered an inappropriate measure for diversity (Solow & Polasky, 1994; Meinl, Ostermann & Berthold, 2011).

Sum of distance to nearest neighbours

The *sum of distance to nearest neighbours* (SDNN) penalises clusters of solutions and can be fooled by artefact sets with multiple duplicates of only two maximally distant solutions (Ulrich, Bader & Thiele, 2010).

Average objective value

The *average objective value* (AOV) is defined as the mean of fitness over a set of solutions. While it is comparable to the QD-score, it lacks an indication of the spread of solutions, because it ignores niche memberships.

The field of ecology is concerned with measuring the diversity of species in a given habitat. Ecological measures of diversity primarily rely on the relative abundances of species, i. e. the normalised counts of animals belonging to the species of interest. For this, the fauna of a specific geographic area is surveyed and individual animals are assigned to classes of species following a biological taxonomy. Many ecological measures calculate diversity solely from these relative abundances (Simpson, 1949).

Solow and Polasky (1994), however, make a case for the importance of species similarity (or distance interpreted as dissimilarity) when measuring diversity and, together with Weitzman (1992), lay out three properties as requirements for a diversity measure D of a collection of artefacts $A$ (Ulrich, Bader & Thiele, 2010):

Requirements for an ecological diversity measure

1. *Monotonicity in varieties*

   Adding a new artefact $b$ to a collection increases the diversity.

$$\mathrm{D}(A \cup b) > \mathrm{D}(A) \qquad \text{if} \quad \min_{a \in A} d(a, b) > 0$$

2. *Twinning*

   Adding a duplicate artefact $c$ to a collection does not change the di-

versity, if it is identical to an existing artefact $a_i$ with the same similarity relation to all other artefacts $a_j$.

$$\begin{aligned} \mathrm{D}(A \cup c) = \mathrm{D}(A) \qquad & \text{if} \quad d(a_i, c) = 0 \quad \exists\, a_i \in A \\ & \text{s.t.} \quad d(a_j, a_i) = d(a_j, c) \quad \forall\, a_j \in A \end{aligned}$$

3. *Monotonicity in distances*

   The diversity should not decrease if all pairs of artefacts are at least as dissimilar as before.

$$\begin{aligned} \mathrm{D}(A') \geq \mathrm{D}(A) \qquad & \text{iff} \quad d(a'_i, a'_j) \geq d(a_i, a_j) \\ & \forall\, a_i, a_j \in A, \quad \forall\, a'_i, a'_j \in A' \end{aligned}$$

Solow-Polaski diversity

The corresponding *Solow-Polaski diversity* (SPD) measure defines the diversity of a given set of species as the joint dissimilarity of the species in the set (Weitzman, 1992; Solow & Polasky, 1994). This definition differs significantly from standard definitions of diversity in ecology, as it does not take into account the relative abundances of species but only their dissimilarities, meeting all three requirements above. The measure is computed on pairwise distances and requires a computationally expensive matrix inversion. SPD has been applied to multi-objective optimisation (Ulrich, Bader & Thiele, 2010) and the evaluation of high-dimensional phenotypes (Hagg, Preuss et al., 2020).

Pure Diversity

Pure Diversity (PD), an alternative formulation of SPD, is also solely based on pairwise distances but is implemented as a recursive subset search, eliminating the need for matrix inversion (H. Wang, Jin & Yao, 2017). For large artefact sets it can still be expensive to compute. The PD of a set of artefacts $A$ is calculated recursively by finding the artefact with the maximum distance to the subset of all other artefacts. The PD score is thus the sum of linked dissimilarities between artefacts.

$$\mathrm{PD}(A) = \max_{a \in A} \left( \mathrm{PD}(A \setminus \{a\}) + \mathrm{d}(a, A \setminus \{a\}) \right) \tag{2.19}$$

The distance between a set of artefacts $X$ and an individual artefact $y$ is obtained by finding the most similar artefact in the set $x \in X$ and calculating the dissimilarity between $x$ and $y$.

$$\mathrm{d}(y, X) = \min_{x \in X} \text{dissimilarity}(x, y) \tag{2.20}$$

We typically formalise dissimilarity as the distance between vector representations of artefacts. The $L^p$-norm with $p < 1$ has been recommended for high-dimensional cases (H. Wang, Jin & Yao, 2017).

The PD measure is motivated by the following observations. A singleton, i. e. a set that contains exactly one item, has no diversity $\mathrm{PD}(a) = 0$. Adding an item to a set increases the set diversity by the dissimilarity between the set and the new item $\mathrm{PD}(A \cup b) = \mathrm{PD}(A) + d(b, A)$. We can thus estimate the diversity of a full set by starting from a single item, gradually adding items and summing their dissimilarities. However, unless we specify which item to add next there are too many options. Instead, we add the most dissimilar item (note the maximisation term in Equation 2.19). The PD of a set is thus the maximum joint dissimilarity between its items.

In the context of statistical machine learning, however, considerations of the required properties of a diversity measure differ from those in ecology. In particular, the *twinning* property has to be re-evaluated, since duplicate training examples correspond to an imbalance in a dataset, which can be a cause for bias in a trained model. It is to be expected that such duplicates are assigned a higher probability mass under a statistical model while reducing the overall likelihood of singular examples. Adding a duplicate to a training dataset thus decreases the diversity of the artefacts sampled from a model trained on that dataset. Consequently, relative abundance and similarity are both relevant factors in the evaluation of diversity in machine learning.

Diversity of order q

The authors of *Diversity of order q* ($\mathrm{D}^q$), a family of measures of biological diversity, advocate for the importance of the similarity of species as a criterion that complements information on the relative abundances of species (Leinster & Cobbold, 2012). The measure's sensitivity to the abundance of rare species can be adjusted through the parameter $q \in [0, \infty]$. For

$q = 0$, rare species have the same weight as common ones. For $q = \infty$, only the most abundant species affect the score, rare ones are ignored. As a family of measures, $D^q$ subsumes many popular measures of diversity in ecology. While it is customary to gather relative abundance data of a community of species, this information can be difficult to obtain in complex domains such as images. Animals are assigned to discrete categories following a biological taxonomy, giving a discrete distribution over species. The same cannot be done for images without some difficulties and taking into account important considerations. Though common in ecology, any division into categories is somewhat arbitrary and only sometimes useful. A typology can be more appropriate, allowing for the classification along multiple criteria (e. g. images of scenes in different locations at different times might be classified along two binary criteria, resulting in four classes: *daytime-inside*, *daytime-outside*, *nighttime-inside*, *nighttime-outside*). Dividing training examples into categories can be particularly problematic if the training data involves people (e. g. images of human faces). Dividing aspects that are expressed as continuous variables (e. g. age) can result in crude and limiting classifications when turned into nominal values (e. g. young and old). And inferring certain qualities or characteristics about people from their appearance alone is simply wrong. In any case, we need to remember that any classification is potentially meaningless if we train a generative model unconditionally, that is to say, ignoring the class separations of the training data. Even if it was feasible and useful to assign images to separate categories, we would be facing the same question as ecologists: What is the similarity between the different categories? In ecology, this question demands a conscious decision to measure diversity in terms of a specific aspect of the species. For example, if we are interested in genetic diversity, we require the genetic similarity between the categories of species. Similarly, a diversity measure can be tuned to a different aspect of diversity when we use the functional or morphological similarity between species. To do the same for images, we need to decide which aspects are important for diversity and how to determine the similarity between categories in these aspects.

Vendi Score

The Vendi Score (VS) is specifically designed to measure diversity in a machine learning context (Friedman & Dieng, 2023) and through its unsupervised approach has important advantages over the measures presented above. The relative abundances of artefacts are implicitly considered through a similarity function that reflects the correlations across artefacts. Thus, instead of relying on a clear separation of artefacts via distinct classes, as is customary for measures of biodiversity, the VS infers abundances from similarities. If there are many artefacts in a dataset which are very similar to each other, their pairwise distances will be very low and they will implicitly form a group of artefacts with the same likeness. The selection of the similarity function is thus an important domain-dependent choice. Yet, the measure is still problem-agnostic, as it does not require any a priori problem knowledge. Specifically, the diversity of a collection of $N$ artefacts $x_1, \ldots, x_N$ is calculated purely from the pairwise similarity matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ of artefacts, which is obtained via a similarity function $k(x_i, x_j) = \mathbf{K}_{ij}$. The VS is defined as the exponential of the Shannon entropy of the eigenvalues $\mathbf{\Lambda}$ of the normalised similarity matrix $\mathbf{K}/N$.

$$\mathrm{VS}(\mathbf{K}) = \exp\Big(-\sum_{i=1}^{N} \lambda_i \log \lambda_i\Big) \tag{2.21}$$

Here $\mathbf{K}$ is a positive semi-definite similarity matrix ($N \times N$) between pairs of artefacts such that $k(x, x) = \mathbf{K}_{ii} = 1$ for all $x$. The eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$ of the normalised similarity matrix can be obtained via the eigendecomposition $\mathbf{K}/N = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$ as the diagonal elements of the diagonal matrix $\lambda_i = \mathbf{\Lambda}_{ii}$. VS computes the effective rank of the similarity matrix $\mathbf{K}$, the exponential of the entropy of its normalised singular values.

Probability-weighted Vendi Score

In the probability-weighted formulation of the VS (Friedman & Dieng, 2023), relative abundances can be explicitly quantified via a probability vector $\mathbf{p} = (p_1, \ldots, p_N)$. The definition is the same as before (Equation 2.21), but the similarity matrix is instead normalised by the probability weights $\mathbf{K}_\mathbf{p} = \mathrm{diag}(\sqrt{\mathbf{p}})\,\mathbf{K}\,\mathrm{diag}(\sqrt{\mathbf{p}})$.

Family of Vendi Scores

Similar to $D^q$ adopting the Hill numbers (Hill, 1973), the family of Vendi Scores extends the VS measure to a collection of diversity measures with a parameter $q$ that controls its sensitivity to under-represented artefacts (Pasarkar & Dieng, 2024). As in $D^q$, the higher the parameter $q$, the less sensitive the measure is to rare examples. The authors thus recommend measuring diversity with a VS of small order $q \in [0.1, 0.5]$. The basic formulation of the VS is equal to the special case $q = 1$ (Equation 2.21).

$$\mathrm{VS}_q(\mathbf{K}) = \exp\left(\frac{1}{1-q}\log\sum_{i=1}^{N}(\lambda_i)^q\right) \tag{2.22}$$

As before, the similarity matrix $\mathbf{K}$ is either normalised by the number of artefacts $N$ or the probability weights $\mathbf{p}$. To calculate the score we then compute the eigenvalues $\boldsymbol{\lambda}$ of the normalised similarity matrix. We adopt the probability-weighted VS to calculate the *diversity weights* for our method to increase the output of a generative model (Chapter 5).

Measure inconsistencies

Both the VS and $D^q$ claim to give as output an *effective number* (Hill, 1973), representing the count of absolutely dissimilar items of equal abundance in a dataset. However, given the same data, the measures do not agree, producing different scores and thus different 'effective numbers'. Which of the two measures does in fact give an *effective* estimate? And, if the two measures are related, what is the transformation of one score to the other that explains their disagreement? Resolving this inconsistency and identifying potential relationships between measures is subject to future work.

## 2.6 HUMAN PERCEPTION OF SIMILARITY

Studies on the human perception of similarity are at the core of psychophysics. They cover a large variety of stimuli, from more basic stimuli such as sound or colour to complex ones such as motion or 3D models. However, to the best of our knowledge, in the domain of video games, there exist no empirical studies on human similarity judgement and its comparison to surrogate metrics. A taxonomy of game evaluation metrics put forward by

Volz (Volz, 2019, Appendix A) suggests that very few of such metrics in games draw on insights into human perception, and of those few, none measure similarity. Previous related work on the alignment of computational metrics with human perception (Mariño, Reis & Lelis, 2015; Summerville et al., 2017) focuses on human perception of fun, difficulty, and aesthetics within individual levels. Arguably the closest predecessor to the present study, Mariño, Reis and Lelis (2015) investigate whether a series of computational metrics used in PCG adequately capture player's perceptions of levels of Super Mario Bros. Amongst unrelated metrics, they calculate Compression Distance as a metric of structural dissimilarity between pairs of levels. Crucially though, they do not correlate it with the player's perception, likely because the experimenters did not find significant differences in compression distance between the generated levels examined in the user study. In contrast, our present work focuses specifically on comparing similarity metrics to people's perception of similarity of game levels.

Having covered the technical background of generative modelling approaches in deep learning, in the next chapter, we discuss how such techniques have been put to use in the context of artistic and creative practices to generate artefacts of high cultural value.

Chapter 3

# ARTISTIC AND CREATIVE USES OF GENERATIVE MODELS

In this chapter, we introduce the term *active divergence* to describe a common theme in the artistic uses of generative deep learning (DL) (Berns & Colton, 2020), thereby addressing RQ 1. Artists often consciously break, tweak or otherwise intervene in data-driven generative processes in order to produce artefacts, that are culturally valuable, but sub-optimal from a pure modelling perspective. We present an introductory overview of some active divergence techniques, many of which require human supervision of important tasks and decisions.

We further present a framework for automating generative deep learning (DL) with a specific focus on artistic applications (Berns et al., 2021). The framework adopts core concepts from automated machine learning (AutoML) and is informed by the theory and practice of computational creativity (CC). To motivate the framework, we argue that automation techniques are a pathway to increasing the *creative responsibility* of a generative system, a central theme in CC research. The interaction between the engineer and the generative system can be framed as a *co-creative* act. We describe the standard pipeline for the development and deployment of generative DL models and highlight how artistic practices differ from this standard. Both pipelines, in the standard and artistic settings, include many tasks and decisions that normally would be performed or taken by a person. In our framework, we formalise such decisions as *targets for automation*: opportunities for addressing manual tasks with computational means. The framework, through its targets, makes a central contribution to integrating the concept of active divergence into CC research.

The work in this chapter was presented at ICCC in 2020 and 2021:

Berns, S., & Colton, S. (2020). Bridging Generative Deep Learning and Computational Creativity. *Proceedings of the 11th International Conference on Computational Creativity (ICCC)*.

Broad, T., Berns, S., Colton, S., & Grierson, M. (2021). Active Divergence with Generative Deep Learning - A Survey and Taxonomy. *Proceedings of the 12th International Conference on Computational Creativity (ICCC)*.

Berns, S., Broad, T., Guckelsberger, C., & Colton, S. (2021). Automating Generative Deep Learning for Artistic Purposes: Challenges and Opportunities. *Proceedings of the 12th International Conference on Computational Creativity (ICCC)*.

CONTENTS

## 3.1 INTRODUCTION

In recent years, methods in generative machine learning have seen steady improvements in their training stability, scale and output fidelity. Over the course of only a few years, research contributions have pushed models from generating crude low-resolution images to highly photo-realistic output. Cultural value, however, does not necessarily correlate to high artefact fidelity, i. e. photorealistic images. While imperfect from a pure machine learning perspective, earlier modelling approaches, GANs in particular, lend themselves to co-creative work in the visual arts. Rather than produce photorealistic imagery, they evoke visual indeterminacy, i. e. produce images which "appear to depict real scenes, but, on closer examination, defy coherent spatial interpretation" (Hertzmann, 2019). An online community that formed under the hashtag 'creative AI' (Cook & Colton, 2018) has been particularly eager to embrace this aesthetic and successful in exploring unconventional applications of generative models. Visual artists like Mario Klingemann, Sofia Crespo and Bas Uterwijk have established successful artistic practices around this approach. In major academic conferences with a focus on artificial neural networks, creative and artistic perspectives have found a place in workshops and art exhibitions. But while generative machine learning researchers focus their work on the very specific objective of distribution fitting, creative practitioners are happy to adopt lucky mistakes and embrace generative imperfection. In this chapter, we analyse some of the techniques used to create culturally valuable artefacts with generative models and define the general approach as follows:

> *Active divergence* using generative machine learning methods to intentionally model a new distribution that does not directly approximate a data distribution to generate novel artefacts.

We make the case for generative modelling approaches, in particular GANs, as a successful and useful technology for image generation. However, in their conventional formulation, they are inherently limited in their creative abilities by the objective of perfectly matching a given data distribution.

In the following section, we discuss how, despite its limitations, GANs have been used as artwork production engines and seen heavy customisation for this purpose. We then explore how CC research can contribute to further evolving such models into more autonomous creative systems, looking specifically at measures of novelty as a first step towards this goal.

### 3.1.1 LEARNING FOR PERFECTION

While the purpose of GANs, like all generative models, is to accurately capture the patterns in a dataset and model its underlying distribution, guaranteeing convergence for this particular method remains a challenge (Lucic et al., 2018). Theoretical analyses of the GAN training objective suggest that the models fall significantly short of learning the target distribution and may not have good generalisation properties (Arora et al., 2017). It has further been suggested that GANs in particular might be better suited for other purposes than distribution learning. Given their high-quality output and wide artistic acceptance, we argue for the adaptation of this generative approach for CC purposes.

Generative models are currently only good at producing 'more of the same': their objective is to approximate the distribution of a given training dataset as closely as possible. This highlights two sides of the same fundamental issue. First, in practice, it remains unclear whether models with millions of parameters simply memorise and re-produce training examples. Performance monitoring through a hold-out test set is rarely applied and overfitting in generative models is not widely studied. Second, and most importantly in this context, conceptually, such models are only of limited interest for creative applications if they produce artefacts that are insignificantly different from the examples used in training. Hence we further argue for an adaptation such that generative capabilities align with the objectives of CC: to take on creative responsibilities, to formulate their own intentions, and to assess their output independently (Colton & Wiggins, 2012).

### 3.1.2 ACTIVELY DIVERGING FROM PERFECTION

In order to produce artefacts in a creative setting, GANs still require expert knowledge and major interventions. Artists use a variety of techniques to explore, break and tweak, or otherwise intervene in the generative process. In the following, we present three case studies as illustrative examples of *acive divergence* techniques. A more comprehensive survey of the state of the art at the time of publication can be found in Broad et al. (2021). From a pure machine learning perspective, these exploits and accidents only produce sub-optimal results, since their objective is different from perfectly modelling the training data distribution. Actively diverging from local likelihood maxima in a generator's internal representation is necessary to find those regions that hold sub-optimal, but culturally valuable artefacts that would otherwise rarely be sampled.

*Cross-domain training* blends two (or more) training sets of the same modality, such that a model is first fit to the images from one type (e. g. human faces) and then fine-tuned to another (e. g. beetles). The resulting output combines features of both into cross-over images (Figure 3.1). Finding the right moment to stop fine-tuning is crucial and human supervision in this process is indispensable.

*Loss hacking* intervenes at the training stage of a model where the generator's loss function is manipulated in a way that diverts it towards sub-optimal (with respect to the traditional GAN training objective) but interesting results. Given a model that generates human faces, for example, the loss function can be negated in a fine-tuning process such that it produces faces that the discriminator believes are fake (Figure 3.2; Broad, Leymarie & Grierson, 2020). Again, human supervision and curation of the results are just as important as devising the initial loss manipulation.

Figure 3.1: Example for *cross-domain training*: StyleGAN trained on the FFHQ dataset (Karras, Laine & Aila, 2019), fine-tuned on a custom beetle dataset. Reproduced with permission from M. Mariansky.[1]

*Early stopping and rollbacks* are necessary whenever a model becomes too good at the task it is being optimised for. Akin to the pruning of decision trees as a regularisation method or focusing on sub-optimal (in terms of fitness functions) artefacts produced by evolutionary methods, rollbacks can improve generalisation, resulting in artefacts that are unexpected rather than perfect. As an example, Pinar Yanardag described the process of training a GAN to generate a 'little black dress' on Twitter[2]: "The algorithm works so well that ... we actually had to go to earlier iterations to find 'creative' designs (when the model was still 'learning' and making mistakes like weird color patches)."

All of the above techniques require manual interventions that rely on human action and personal judgement. There are no well-defined general criteria for how much to intervene and at which point or when to stop. It is central to an artistic practice to develop such standards, nurture their individuality and highlight the difference from other practices. A major theme in GAN art, however, and a commonality in the above non-standard uses, is the *active divergence* from the original objective of the tool of the trade, in pursuit of novel and possibly surprising artefacts of high cultural

1 Tweet by @mmariansky. https://twitter.com/mmariansky/status/1226756838613491713
2 Tweet by @PINguAR. https://twitter.com/pinguar/status/1109821860273967104

Figure 3.2: Samples from Broad, Leymarie and Grierson (2020) of StyleGAN fine-tuned with a negated loss function. In its state of 'peak uncanny' the model started to diverge but has not yet collapsed into a single unrecognisable output.

value. This dynamic appears to be, in contrast to other artistic disciplines, exceptionally pronounced due to the use of state-of-the-art technology that has yet to find its definite place and purpose and whose capabilities are open to be explored. We celebrate and support this endeavour and argue that CC can help by pushing generative models further, towards new objectives.

### 3.1.3 NEW OBJECTIVES FOR GENERATIVE MODELS

In a creative setting, two avenues of future applications for generative models come to mind: (1) creativity support tools and (2) autonomous creative systems, which we briefly discuss in the following, together with necessary improvements.

First, let us consider creativity support tools (Shneiderman, 2002), where a person is in charge of achieving an independently formulated creative goal and, to this end, draws on assistance from tools and technologies, such as generative DL. For generative models to be useful in this context and to facilitate the creative process rather than obstructing it, two main requirements need to be addressed. On the one hand, generative DL needs to be more accessible. The deployment of generative models in general, and the active


Figure 3.3: Series of image edits applied to three different GANs with the method from Härkönen et al. (2020)

divergence techniques presented here in particular, require highly technical knowledge and specialised computing resources of limited availability. On the other hand, models need to be more controllable in their generative process to allow for precise interventions and human-directed creation. Active research on disentangled representation learning has proposed interpretable controls for global image manipulation (Härkönen et al., 2020). Common dimensions of variance in the data are first identified by the model and later manually sighted and named. Interpretable controls allow for the manipulation of images in a single specific aspect, such as a person's age, the exposure of a photograph or the depicted time of day, while maintaining the others (Figure 3.3). Similarly, localised semantic image edits (Collins et al., 2020) transfer the appearance of a specific object part from a reference image to a target image, e. g. one person's nose onto another person's face.

Second, generative models can be augmented to be turned from creativity support tools into more autonomous creative systems (Saunders, 2012). For this, some of the creative responsibilities conventionally held by the person in charge of the generative process need to be handed over to the system, such that it is, for example, able to formulate an intention or evaluate its own output. This may require endowing the system with additional, more sophisticated abilities and responsibilities. We argue, that as a start, gener-

ative modelling approaches need to be adapted to allow for the generation of novel artefacts, instead of reproducing the examples from a given dataset. Our framework for automating generative DL, presented in the following Section 3.2, provides specific targets for automation in the development and deployment of generative models.

While creativity is arguably an essentially contested concept (Jordanous & Keller, 2016) and there exist a variety of individual definitions, many of those include the notions of novelty, surprise and some form of value (e. g. usefulness or significance) (Boden, 2004; Runco & Jaeger, 2012; Jordanous, 2013). Our analysis of GAN artists' work highlights a commonality in their practices: divergence from the standard machine learning practice to produce novel, perhaps surprising, outputs. The early focus of this thesis research was novelty, which eventually shifted to diversity. Yet, there is a connection between the two concepts which we discuss in the introduction (1), together with their connection to similarity as a foundational relation. The novelty of an artefact can be interpreted as the amount of diversity that it adds to a collection. A model that produces diverse output, possibly more diverse than a reference dataset, will thus also generate novel artefacts. Hence, the following section focuses on the aspect of novelty and how the output of generative models could be assessed in regard to novelty.

### 3.1.4 EVALUATING NOVELTY

As many evaluation schemes for creativity include notions of novelty, an exhaustive review of the literature is beyond the scope of this chapter and thesis. We focus here on explicit measures of novelty, in particular in the context of generative models. Currently, novelty can be achieved by tuning the stochasticity of a generative process whenever it is conditioned on a distribution of probabilities. In GANs, the latent code truncation trick clips values drawn from a normal distribution to fall within a limited range (Brock, Donahue & Simonyan, 2019). On the other end, a temperature parameter can be applied to scale a network's softmax output (Feinman &

Lake, 2020). Both improve the quality of individual artefacts at the cost of sample diversity. While the original intention is to decrease randomness to obtain artefacts closer to the mean, they may also be able to achieve the inverse. Neural network-based methods have been proposed for the generation of novel artefacts, e. g. CAN (Elgammal et al., 2017), Combinets (Guzdial & Riedl, 2019), as well as several measures for the evaluation of GANs, e. g. the Inception Score (Salimans et al., 2016) and FID (Heusel et al., 2017), discussed in more detail in Section 2.4. However, none of these measures can be used to estimate novelty or to compare the extent to which DL methods are capable of producing it. For a measure that might fill this gap, we can draw from work in CC.

Ritchie (2007) proposes a formal framework of empirical criteria for the evaluation of a computer program's creativity, advocating for a post-hoc assessment based on a system's output and independent of its process. A definition of creativity focuses on novelty, quality and typicality, where the latter refers to whether an artefact matches the intended class (e. g. when generating jokes, whether it has the formal structure of a joke). Quality (also denoted as value) and typicality are expressed as ratings. Novelty is seen as the relationship between the input and output of a program and is formalised in a collection of proportions in set-theoretic terms.

Most interesting for our purposes is Ritchie's concept of an 'inspiring set', which could be treated as the knowledge base but, in the context of learning algorithms, does not have to be equivalent to the training set. Representing the examples that the author of a generative system hopes to achieve, it would be too trivial to allow a learning algorithm a glimpse at such examples. Rather, an inspiring set can inform about the necessary choices in the design process of a generative system that might evoke the desired output. Current discussions around the inductive biases of the fundamental building blocks in DL pose similar questions. Recent work has tried to leverage the specific choice of structure in hybrid neuro-symbolic models (Feinman & Lake, 2020). This idea leaves room for the question of how the concept of an inspiring set could be integrated into the training and sampling schemes of a generative model.

In work on curious agents, Saunders et al. (2004, 2010) use Self Organising Maps (SOMs) (Kohonen, 1988), to measure the novelty of an input through a distance metric in vector space and in comparison to all other examples stored in the SOM. 'Interestingness' is estimated through an approximation of the Wundt curve (Berlyne, 1960) (the sum of two sigmoids), to the effect that the score peaks at moderate values of novelty and rapidly drops thereafter. This model is based on the understanding that for new stimuli to be arousing, they have to be sufficiently different but not too dissimilar from known observations.

Pease, Winterstein and Colton (2001) discuss novelty in relation to complexity, archetypes and surprise, and propose specific metrics for these aspects. First, an item is deemed more novel the more complex it is. Complexity is defined in terms of the size of a given domain and how unusual and complicated the generation of an item is, which attempts to capture how many rules and how much knowledge was necessary in the process. Second, responding to Ritchie's typicality, novelty is defined as the distance of an item to a domain's given archetypes. This approach is similar to Saunders et al. (2004, 2010) in that it compares items to a knowledge base and computes distances in vector space. Third, the authors argue that 'fundamental' novelty evokes surprise as a reaction. However, a metric for surprise cannot be used to prove novelty, it only shows the absence of 'fundamental' novelty through the lack of surprise.

At the beginning of this section, we pointed towards two applications of generative models in creative settings: creativity support tools and autonomous creative systems. We discussed approaches for the evaluation of novelty, primarily from the CC literature, for the benefit of artistic practices, in particular when using creativity support tools where human supervision is essential. In the following section, we focus on advancing autonomous creative systems by endowing them with creative responsibilities to increase the system's creative autonomy. For this, we adopt automation techniques from AutoML.

## 3.2 AUTOMATING GENERATIVE DEEP LEARNING FOR ARTISTIC PURPOSES

The increasing demand in industry and academia for off-the-shelf machine learning methods has generated a high interest in automating the many tasks involved in the development and deployment of machine learning models. Such automated machine learning (AutoML) can make machine learning more widely accessible to non-experts, and decrease the workload in establishing machine learning pipelines, amongst other benefits. Examples of the different task areas that AutoML techniques address include data preparation, feature engineering, neural architecture search (NAS), hyper-parameter optimisation and model selection. AutoML is a very active area of research. The progress to date has been documented in several surveys (e. g. Truong et al., 2019; Tuggener et al., 2019; Chauhan et al., 2020; He, Zhao & Chu, 2021) and a book (Hutter, Kotthoff & Vanschoren, 2019). The AutoML challenges (Guyon et al., 2019) and the workshop at the International Conference on Machine Learning (ICML) have evolved into a dedicated AutoML conference (Guyon et al., 2022; Faust et al., 2023; Eggensperger et al., 2024). Crucially though, at the time our contribution was published (Berns et al., 2021), the automation of generative modelling pipelines had received little attention.

In the evolution of DL as a subdomain of machine learning, some of the early advances consisted of incorporating previously manual pre-processing steps into an automated optimisation pipeline. While some machine learning algorithms require the extraction and construction of features by hand and image filters have to be selected depending on the task, modern DL techniques have no issues with handling raw data, and image filters can easily be learned end-to-end. This brings significant advantages, allowing us to scale up the purely computational processes. This has contributed to the consistent success of the field. As the complexity of DL and its impact on society have increased, it has become more pressing and difficult to

solve remaining manual tasks and decisions, such as the choice of network architecture and the tuning of hyper-parameters.

While AutoML is concerned with automating solutions for classification and regression, methods in generative DL deal with the task of distribution fitting, i. e. matching a model's probability distribution to the (unknown) distribution of the data (see Section 2.1). NAS, an important topic of research in AutoML, has been extended to GANs (Gong et al., 2019; Y. Fu et al., 2020; C. Gao et al., 2020; M. Li et al., 2020) and diffusion models (L. Li et al., 2023). Moreover, evolutionary approaches have been applied to optimising the GAN training objective (C. Wang et al., 2019) and other training parameters (Costa et al., 2019). Even though certain aspects of the GAN training scheme have been automated, we highlight three gaps in existing research: 1) there exists no unified automation framework for generative DL more generally; 2) existing work does not address the use of generative DL for creative applications; 3) researchers have not sought to motivate the automation of DL systems to endow artificial systems with creative autonomy.

We propose a framework for the automation of generative DL that, on the one hand, adopts core concepts from AutoML, and on the other hand, is informed by the theory and practice of CC research (Colton & Wiggins, 2012). The framework has two goals. First, to highlight opportunities for automation in the generative DL pipeline for artistic purposes. Second, by automating some parts of this pipeline, to endow a computational system with more *creative responsibilities* (Colton, 2009), i. e. the ability to make decisions that have a high impact on the outcome of a creative process. These individual decisions can be understood as *targets* for automation when framing the design of a generative DL pipeline as a form of *co-creativity* (Kantosalo et al., 2014). Under this interpretation, we inform the automation of generative DL more specifically with well-established, generic CC strategies to equip computational systems with creative responsibilities. Our framework differs from AutoML not only in its stronger focus on generative models but also in the assumed goals of the generative DL pipeline. More specifically, we identify targets for automation based on the wide and successful

application of generative DL in artistic work. In contrast to standard applications, people in an artistic setting prefer to produce artefacts of high cultural value over perfectly generalised reproductions of the training data. In this sense, they aim to *actively diverge* from a given data distribution (see previous Section 3.1).

In connecting the two research fields, AutoML and CC, our framework benefits the long-term goal of artificial intelligence to develop autonomous systems that can devise novel concepts, strategies and artefacts. Both automation and creativity play a vital role in developing autonomous systems and in enabling open-ended innovation. For automation, we can leverage the techniques from AutoML. However, these techniques often optimise for a specific, clearly defined objective and in this sense are narrowly-focused solutions. Per definition, novel artefacts cannot be specifically defined a priori. They are *unknown unknowns* (Lehman et al., 2025). By incorporating insights from CC, the framework extends beyond task-specific automation towards a collaborative creative process between human and computer.

Our main contribution is to gather, standardise and highlight opportunities to automate generative DL for artistic applications. We identify commonalities of DL pipelines in artistic projects and bring them together in a common framework. This provides a starting point for handing over creative responsibilities in a range of applications, not only artistic. We concentrate our efforts on generative DL, rather than generative machine learning more generally. While we assume the majority of applications to be built on DL approaches, we do not rule out that other generative machine learning methods might be used within the framework. Our contribution does not consist of a formal solution to a singular automation problem. Instead, we aim to provide a big-picture view of all automation tasks and their associated opportunities and challenges, to be solved in future work.

To leverage insights from CC in the development of our framework, we first clarify the relationship between automating generative DL and endowing artificial systems with creative responsibility. We then outline a standard non-automated pipeline for the development and deployment of generative DL models, and show how applications in artistic settings differ

from this standard pipeline. Drawing from these two sources, we lay out the automated generative DL pipeline, describe several targets for automation therein and suggest ways in which automation could be achieved. We continue with an illustrative example to demonstrate how our framework can give inspiration and guidance in the process of gradually handing over creative responsibility to a generative system. We analyse the relationship between automation and creative autonomy in the context of our framework. We conclude the chapter by discussing the limitations of our framework and suggest directions for future work.

### 3.2.1 AUTOMATED, ARTISTIC DEEP LEARNING AS CO-CREATION

We believe that the development of a framework for automated generative DL can benefit from the insights gathered over more than two decades of CC research because the automation of targets in generative DL can be considered a specific instance of the grand CC goal to give computational systems responsibility over decisions in a creative process.

With each creative responsibility that is handed over to the system, i. e. with each target that is being automated, we increase the computational system's *creative autonomy* (Jennings, 2010; Guckelsberger, Salge & Colton, 2017; McCormack, Gifford & Hutchings, 2019), i. e. its capacity to operate independently of a human instructor, allowing for it to be ultimately considered a creator in its own right (Colton, 2008b). Crucially though, the users of automated generative DL typically want to retain some control over the automation and its outcome. In developing our framework, we must thus decide which responsibilities should be retained to sustain certain modes of interaction between the artistic users and the generative DL system.

To this end, it is useful to frame this interaction in the process of automation as a *co-creative* act. We adopt Kantosalo et al.'s (2014) working definition of *human-computer co-creativity* as 'collaborative creativity where both the human and the computer take creative responsibility for the generation of

a creative artefact'. To qualify as a collaborative activity, both human and system must achieve *shared goals* (Kantosalo et al., 2014, drawing on Terveen, 1995).

Different automation strategies can enable two coarse forms of interaction. First, the user and system could engage in *task-divided co-creativity*, in which 'co-creative partners take specific roles within the co-creative process, producing new concepts satisfying the requirements of one party' (Kantosalo & Toivonen, 2016). Second, they could engage in *alternating co-creativity*, where both partners 'take turns in creating a new concept satisfying the requirements of both parties' (Kantosalo & Toivonen, 2016).

Alternating co-creativity requires the computational system to not only exhibit creative responsibility for either the *generation* or *evaluation* of artefacts but for both. Crucially, even a non-automated generative DL system can be considered creative in a minimal sense, in that it (despite the name) not only 'merely *generates*' (Ventura, 2016) new samples or artefacts, but also *evaluates* their proximity to the training set via its loss function. This is accomplished either explicitly, through likelihood estimation, or implicitly, with the help of a critic in an adversarial setting. The system thus produces artefacts that are *novel* and *valuable*, realising both requirements of the two-component standard definition of creativity (Runco & Jaeger, 2012). We write 'creative in a minimal sense' because the novelty of artefacts will decline, while their value increases, the better the system approximates the (unknown) distribution from which the training data was drawn.

The definition of the training set and loss function by the user satisfies that both partners interact towards shared goals. Through different ways to automate the machine learning pipeline, we can free the human partner from certain manual work, while retaining specific creative responsibilities.

We believe that providing the computational system with creative responsibility in the form of automating certain targets does not constrain, but rather expands the shared creative process. The *person* or *producer* has, due to their personality and cognitive characteristics, a strong impact on the creative *process*, *product*, and the creative environment, i. e. the *press* (Rhodes, 1961; Jordanous, 2016). However, human creativity is also limited, e. g. due to

our bounded rationality (Simon, 1990). A computational system can complement human shortcomings, e. g. via its higher information processing or memory capacity, enabling creativity on larger search spaces (Boden, 2004; Wiggins, 2006b).

### 3.2.2 THE STANDARD GENERATIVE PIPELINE AND ARTISTIC DEVIATIONS

We outline the various steps in the process of building and deploying a generative DL model for standard non-automated usage and contrast it with the particular differences that arise when using a model in different artistic contexts. Additionally, we provide a brief overview of post-training modifications that aim for active divergence (Berns & Colton, 2020), allowing us to manipulate a model into producing artefacts that do not exactly resemble the training data. A more detailed survey of such techniques can be found in Broad et al. (2021). Our goal is to highlight the many choices that have to be taken in the construction of a generative DL pipeline and identify those tasks which pose an opportunity for automation in our framework.

#### DATA ACQUISITION

The first step towards developing generative models is data acquisition. We distinguish two cases: (A) using pre-existing datasets and (B) creating new ones. It should be noted, that generative machine learning is also applied in privacy-sensitive areas such as medicine, and in the augmentation of small datasets, as it can produce synthetic data to replace an entire dataset or supplement it with additional samples. The augmentation by way of a generative model can be necessary whenever a dataset is too small to train another model (e. g. a classifier) with a high number of parameters (i. e. weights and biases in a neural network). However, when the generative model itself requires a large amount of training data, other pre-training data augmentation steps through graphic manipulations can help to do so effectively (Karras, Aittala et al., 2020).

*Using Existing Datasets* In a research setting, it is most common to use standard benchmark datasets or subsets thereof, for training and evaluating generative models. It is generally best practice in machine learning to split the data into training, test and validation subsets. However, generative models are sometimes trained on the entire dataset and alternative methods of evaluation are used.

*Creating a New Dataset* When creating a dataset from scratch, the goal is normally to fully represent the subject or category that is being modelled. Therefore, as much data as possible will be collected to maximise variation in the dataset and to represent all modes as evenly as possible, i. e. the variety of artefacts that are statistically significantly different from one another. Creating varied, high-quality datasets with the large amounts of data required for training generative models can be very labour-intensive and usually the purview of a select few academic and industry laboratories. This is often performed in a distributed fashion, where many workers are involved in collecting, evaluating and labelling data samples.

In contrast to datasets created for industrial and research applications, datasets for artistic purposes are often composed with very different goals. It may not be important to accurately and fully represent a subject matter or domain, as long as the end goal produces interesting results. Datasets are often much smaller, and the considerations for the desired aesthetic characteristics in the results are much more important in deciding which examples should and which should not be included in the dataset. A lot of effort will go into sourcing material and the resulting datasets are much more likely to reflect an artist's individual style and (visual) language. In some cases, the entire dataset will come from an artist's personal archive (Ridler, 2017).

## TRAINING

The objective of training a generative model is to learn a mapping function from an easily controllable and well-understood distribution, e. g. a

standard Gaussian, to a distribution of much higher complexity and dimensionality, e. g. that of natural colour images. There are a number of different training schemes, which apply to different architectures. They are commonly categorised by their formulation of the training objective. Methods maximise the likelihood of the data either explicitly (such as auto-regressive and flow-based models), approximately (e. g. VAEs), or implicitly (GANs). When using a method that explicitly models the data, training will be performed until a desired likelihood score is reached. With VAEs, the goal of training is to maximise the log-likelihood of the dataset. In the adversarial setup, the decision of when to stop training is less clear. Training is often run for a pre-specified period and the results are evaluated qualitatively. A fully trained model ideally represents the entire training data distribution and can be sampled randomly to produce good results. Another desirable quality is that interpolation between two input vectors is matched in the outputs.

Generalisation is a goal of almost all machine learning systems and applications. A model should be able to generalise to unseen data, while not underfitting or overfitting the training data. In an artistic setting, however, this is often less important, and if it produces interesting results, artists may often embrace the aesthetic qualities of an underfit (Shane, 2018) or overfit model (Broad & Grierson, 2017).

### EVALUATION

The general performance of a model is measured in terms of the distance of the learned distribution to the target distribution. A model further ideally covers all modes in the input dataset. For generative methods that explicitly model a probability distribution over the data, the (log) likelihood can be measured and evaluated directly. Implicit methods, such as GANs, have to be assessed with other metrics such as the Inception Score (Salimans et al., 2016) and the FID (Heusel et al., 2017). As these metrics are only a simplified standard for evaluation and have some shortcomings, additional qualitative checks might be needed to ensure the fidelity of the output.

While in some artistic settings, good quantitative performance might matter, it can be ignored entirely in others, and a qualitative assessment of the output is usually much more important. Quality, diversity and accuracy may not be the only considerations (and may even be actively avoided), whereas novelty, interesting misrepresentations of the data and other aesthetic qualities may be desired. Due to the variety of qualities that an artist might look for in a model's output, there is no unique or widely used standard metric for evaluation. This is rooted in the highly individualistic nature of artistic work and linked to the additional strategies for iterative improvements and curation of the output which we discuss in the following subsections.

### ITERATIVE IMPROVEMENTS OF OUTPUTS

Here we look at the diverging strategies for the gradual improvement of a system's output in research and development versus an artistic setting.

*Iterating on the Model* In the research and development of generative models, the dataset often remains fixed, while various aspects of the network architecture and training regime will be altered. For instance, various optimisation hyper-parameters will be evaluated, such as learning rate, momentum or batch size; or network configurations: number of layers, type of activation functions, etc. Different training regimes may also be experimented with, such as optimisation algorithms, loss functions, and methods for regularisation and sampling.

*Iterating on the Dataset* In artistic contexts, it is much more common to iterate on the dataset and keep other parameters fixed, before possibly making iterative improvements to the network and model parameters. Data that appears to be producing unwanted results, or skewing the model in certain directions may be removed. Revisiting the composition of samples (such as cropping), and the removal and addition of samples to refine the dataset may be undertaken (Schultz, 2020).

### DEPLOYMENT

Generative models are used differently in standard and artistic settings in accordance with their respective goals. We here differentiate between standard sampling and output curation.

*Standard Sampling* Generative models are trained with the goal that they can be sampled randomly and every generated output will be of value and high typicality (Ritchie, 2007). Therefore, in most standard applications models are simply sampled randomly with no additional filtering taking place. When filtering is performed, it is often done with the goal of quality evaluation, such as using the discriminator for evaluation quality (Azadi et al., 2019), or using the contrastive language-image pre-training (CLIP) model (Radford et al., 2021), as was the case in evaluating and ranking the generated outputs of the discrete VAE model in the DALL-E image generation project (Ramesh et al., 2021).

*Output Curation* Rather than sampling randomly from a model, artists will often spend a lot of time curating a model's output. The goal of building a model in an artistic setting is not necessarily to generate only samples of high value, but to produce some interesting or novel results, which can then be hand-selected. This can be through filtering samples or searching and exploring the latent space. In some cases, such as combining language-image models with latent space search for text-to-image generation, e. g. Murdock (2021), much effort goes into prompt engineering to find a specific latent vector that produces interesting results. These examples can be seen as a particular cases of *explorative creativity* (Section 2.2).

### POST-TRAINING MODIFICATIONS

Having looked previously at the curation of a model's output in an artistic setting, i. e. the act of identifying the few artefacts of interest in a large set of output samples, we now turn to active divergence techniques (Berns & Colton, 2020) which aim at consistently producing results that diverge

from the training data. These strategies, specifically developed in creative contexts for the purpose of art production, include hacks, tricks and modifications to the model parameters, as well as the daisy-chaining of several models.

One approach is to find a set of parameters where the generated artefacts blend the characteristics of multiple datasets. For this, a pre-trained model can be fine-tuned on a second dataset, different from the original data. As soon as the results present an optimal blend between the two data domains, the fine-tuning can be stopped. This mixture of datasets can also be achieved by blending the weights of two models. Either interpolating on the weight parameters of the two models or swapping layers between models, so that the new model contains higher-level characteristics of one model and lower-level characteristics of another. Another method consists of chaining multiple models together. This allows artists to explore and combine the characteristics of different datasets. Unconditional generative models will often be chained together with domain-translation models, e. g. CycleGAN (Zhu et al., 2017) for sketch-to-image translation, or style transfer algorithms (Gatys, Ecker & Bethge, 2016). Such pipelines aim to produce artefacts that reflect the complex combination of characteristics from many datasets.

Other approaches make modifications to the model to have artefacts completely diverge from any training data. An existing pre-trained model can be fine-tuned using a loss function that maximises the likelihood over the training data (Broad, Leymarie & Grierson, 2020). Other techniques intelligently combine learned features across various models (Guzdial & Riedl, 2019), or rewrite the weights of the model (Bau et al., 2020), re-configuring them to represent novel data categories or semantic relationships. In contrast, *network bending* does not require any changes to the weights of the model (Broad, Leymarie & Grierson, 2021). An analysis of the model is performed to determine which features are responsible for generating different semantic properties in the generated output. Deterministically controlled filters are then inserted as new layers into a model and applied to the activation maps of features.

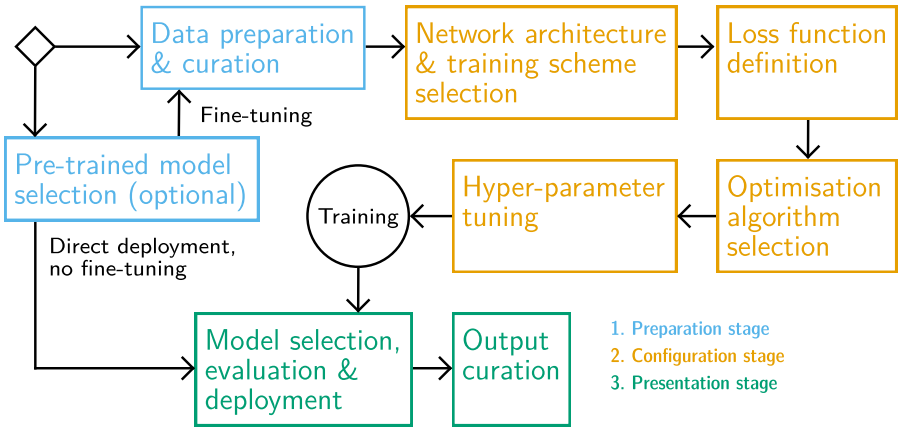



Figure 3.4: Automated generative deep learning (DL) framework in three stages: preparation (blue), configuration (yellow) and presentation (green). The flow starts in the top left and follows the arrows. Individual steps illustrate *targets for automation* (rectangular boxes).

### 3.2.3 THE AUTOMATION FRAMEWORK

We build our framework drawing on the standard generative DL pipeline and its artistic deviations, as previously described. The important difference between the pipeline for a standard generative model and its use in a creative setting is the overall objective of the project. While the former focuses on the model and its performance in terms of fitting the data distribution as closely as possible, a good model in the context of the latter can reliably produce an output of high cultural value. We therefore do not simply extend core concepts of AutoML, but join them with practices observed in the wild and re-frame them within the paradigms of CC. We first discuss automation as a search problem, connecting it to the idea of *transformative creativity*, and highlight some techniques as potential approaches. We then go on to list the targets for automation in a generative DL pipeline for artistic purposes. Some decisions have implications for other targets further down the line, e. g. the number and type of hyper-parameters depend in part on the kind of network architecture and optimisation algorithm. While the process is presented as a sequence of consecutive steps from input to output, it should be understood that all steps are optional and flexibility is required. Improving a system's output works best as an iterative loop in which we might go back and adjust or intervene at any given prior step.

We define the terminology of our framework as follows. With *automation*, we refer to the act of addressing with computational means those decisions in a generative DL pipeline that normally would be taken by a person. A *target* is defined as one such decision which provides an opportunity for automated instead of manual tuning.

### AUTOMATION AS A SEARCH PROBLEM

A generative pipeline is automated by assigning responsibilities over individual targets to either the user or the system. While those retained by a person will have to be tuned manually, all other targets require the system to determine a configuration independently. This problem is analogous to the search problem over hyper-parameters in AutoML. The possible values of each automated target effectively construct a search space over possible system configurations. The number of total permutations, and the resulting search space, can grow rapidly with every independent target added. Search is similarly connected to *transformative creativity* (Wiggins, 2006b), where the properties of a conceptual space are changed such that different artefacts can be reached (Section 2.2).

Limiting continuous parameter values to a reduced range or a set of discrete values, as per grid search for machine learning hyper-parameters, can help make the problem more feasible. The formulation as a search problem is the standard way to tackle automation in AutoML. However, extensive search over meta-parameters can be computationally expensive, time-consuming, cause high energy consumption and consequently have a considerable environmental impact.

The extensive work on search problems provides numerous approaches to constrain this search (Russell & Norvig, 2021). Strategies range from complete, to informed, to random methods. While an exhaustive search can yield an optimal solution, it can be impractical and often infeasible for large search spaces. Random sampling, on the other extreme, can be a surprisingly effective strategy at a low cost and with potentially surprising results. While Jennings (2010) requires a system to meet the *non-randomness* criterion to be

considered creatively autonomous, this definition does not rule out all uses of randomness and allows for testing random perturbations to a system's standards. AI-based search methods can benefit from meaningful heuristics and leverage both exploration and exploitation (e. g. evolutionary search). Gradient-based methods have seen a lot of progress in recent years. Other approaches include rule-based selection and expert systems, with drawbacks including that they require manual construction and expert knowledge.

Finally, machine learning itself can be used to choose values through a pre-trained model. Indeed, practitioners in generative DL tend to go directly to automation via DL. In particular, recent advances in contrastive language-image pre-training (Radford et al., 2021) allow for computing similarities between text and images. Such a model could take over the responsibility of assessing whether an image looks like a text description, or vice versa, at any point in the pipeline where a human artist would do the same task. All of the above approaches can be applied iteratively over subsets of the search space, gradually limiting the range of possible values.

### AUTOMATION VERSUS AUTONOMY

While we have primarily focused on increasing a system's creative autonomy through automation, our framework does not grant a system as much autonomy as to enable it to act entirely independently in response to its own motivations (Guckelsberger, Salge & Colton, 2017). Instead, a system within our framework would remain inactive until engaged. Such engagement can range from a stimulus through available sensors, e. g. cameras, microphones or heat sensors, to a text or image prompt or an entire inspiring set (Ritchie, 2007), to more precise and detailed instructions. In any case, this choice of input channel and sensibility has to be taken by a human and is not a target in our framework.

We further assume the choice of generated media (image, audio, text, video, etc.) to be made by a person prior to building a system. Naturally, it is not difficult to imagine a setup in which this choice, too, becomes part of the pipeline. Going one step further in autonomous automation, our

framework and its targets make it possible to devise a generative system which produces automated generative pipelines. In fact, it might be possible for a generative system to generate itself, much like a general-purpose compiler that compiles its own source code. This self-referential generation has similarly been proposed in work on automated process invention (Charnley, Colton & Llano, 2014).

## TARGETS FOR AUTOMATION

Below we define and discuss the many tasks and decisions that are part of a generative DL pipeline in an artistic setting and which can be automated within our framework. Wherever applicable, we explain how a target relates to concepts of AutoML and CC.

The following subsections identify individual targets for automation. The complete process is illustrated as a sequence of steps in Figure 3.4. As per this diagram, we organise the steps into three stages: 1) a *preparation* stage to gather relevant materials, 2) a *configuration* stage, where the models, training regimes and parameters are tuned to produce valuable output, and 3) a *presentation* stage where the user deploys a final model and curates the output. The first target (selecting a pre-trained model) is optional and can be skipped to start from scratch instead. In this case, we begin with data preparation and curation.

*Pre-trained model (optional)* It might not be necessary to train a network from scratch if an appropriate pre-trained model is available, especially when a quick system setup is desired. A list of pre-trained models, tagged with keywords associated to their generative domain, could provide a knowledge base for a system to select, download and deploy a model. This can either be directly put to use, in which case the system could immediately skip to evaluating the model, or it can be fine-tuned on a smaller set of data. Such additional fine-tuning could be dependent on the outcome of the pre-trained model's evaluation. Only if the pre-trained model's output is not satisfactory would it have to be further optimised or de-optimised. Working

with a pre-trained model has implications for the subsequent choices of the network architecture, training scheme and loss function.

*Data preparation and curation* This preparation step includes the acquisition, cleaning, augmentation and transformation of data samples, akin to data preparation in AutoML. Starting with the data collection task, we consider different data sources from which a system could select. Drawing on existing datasets, such as an artist's private data collection, can introduce important desirable biases and ensure high-quality output. In contrast, scraping samples from the internet could contribute to the generation of surprising results. Additional pre-trained generative models can provide a source for synthesised data in large quantities.

An important addition to the pre-processing is data curation, in contrast to simple cleaning. Rather than filtering out noisy samples, for artistic purposes, it can be desirable to add 'noise'. To this end, it is not uncommon in an artistic context to mix multiple datasets. In this additional step, the system thus further refines the dataset, similar to an artist adding or removing individual samples, which can influence the qualities of the system's final output. This is an opportunity for iterative improvements and for *alternating co-creativity* (Kantosalo & Toivonen, 2016), given that the system both generates and evaluates. Automation in the cleaning and curation tasks can be achieved, e. g. in the image domain, by employing other computer vision or contrastive language-image models.

*Network architecture and training scheme* This target for automation defines the choice of possible architectures (e. g. GAN, VAE, Transformer), which could include non-neural methods. Neural architecture search (NAS) in AutoML is concerned with finding optimal combinations of basic building blocks of artificial neural networks in terms of performance on a classification or regression task, an immensely difficult optimisation problem. We recommend in our framework to instead select from tried-and-tested architectures, only altering parts of the architecture with a direct influence on

the output, e. g. the number of upsampling convolutions which determine the final output image size.

The training scheme is largely influenced by the choice of architecture. In the case of GANs, the training scheme includes the choice of whether to train the discriminator and generator networks in parallel or consecutively and how many individual optimisation steps to perform for either.

*Loss function* The formulation of the basic loss term is highly dependent on a model's training scheme and constitutes the minimum requirement for successful training. However, additional loss terms can change or supplement the basic term for further refinement of the training objective. For example, a novelty loss term could be added by leveraging measures of novelty (briefly discussed above in Section 3.1.4). As a central part of guiding the model parameter optimisation process, any modification to the loss terms will strongly impact the modelled distribution and consequently the system's output. In other contexts, methods have been proposed for the automatic invention of objective functions (Colton, 2008a). These could provide a starting point for adapting the approach to the constraints of loss functions in generative DL.

*Optimisation algorithm* The selected algorithm will be responsible for adjusting a model's parameters through error correction informed by the gradient of the loss function. This choice can potentially have an influence on the system's output, as it is responsible for finding one of the potentially many local minima in the loss landscape. As it determines whether convergence can be reached at all, this decision can ultimately make or break the success of the training process. It can further largely influence convergence speed and be critical in time-sensitive setups. The choice of optimisation algorithms might be limited by the previous selection of network architecture and corresponding training scheme.

*Hyper-parameter tuning* Optimisation of batch size, learning rate, momentum, etc. can be achieved via AutoML methods, and there is much active research in this area.

*Model selection and evaluation* From all the possible models, the best one has to be selected in accordance with the given criteria relevant to the task at hand. As the training process is essentially a succession of gradual changes of model parameters over time, this task is equivalent to identifying the right moment to stop training. Additionally, and in order not to lose previous training states, model checkpoints can be saved along the way as training progresses and whenever model evaluation satisfies given criteria. After training is finished, the best model has to be selected from all candidate checkpoints. In standard ML projects, this would normally be done with respect to the primary concern of predictive accuracy. But in generative projects, other considerations may include how surprising the outputs are, synthesis speed (for tool or real-time uses) and coherence of the results. Such criteria could be employed in a weighted sum of metrics, where the system can give more or less emphasis to individual terms. This would allow the combination of standard metrics like FID in the image domain for general output fidelity with a measure for sample similarity compared to a reference sample(s), inspiring set or text prompt via a contrastive language-image model.

*Output curation* Having obtained a successfully trained model, we want a system to reliably produce high-quality output. While efforts in previous steps were aimed at refining the model which is at the core of the generative process, this final automation target aims to raise the system's overall output quality. Two approaches come to mind: filtering and search. In the former, a system could select those samples from a large batch of model outputs that rank highest against a given metric. In the latter, the system could search for vectors directly in a model's latent space via one of the various methods we have outlined in the section above on approaches to search problems. The evaluation measure, as before, could be the similarity of

samples compared to a set of reference samples, an inspiring set or a text prompt via a contrastive language-image model.

### 3.2.4 AN ILLUSTRATIVE EXAMPLE

In early 2021, a generative DL Colab notebook (Bisong, 2019) called *The Big Sleep* was shared online (Murdock, 2021). It allows for text-to-image generation (Agnese et al., 2020), effectively visualising a user-given text prompt, often with innovative content and design choices, as per the example in Figure 3.5. This is an instance of an artistic deviation from the standard pipeline, where CLIP (Radford et al., 2021) is used to evaluate a generated image with respect to a given text, driving a gradient-based search for latent vector inputs to a generative model called BigGAN (Brock, Donahue & Simonyan, 2019). In more recent text-to-image approaches, instead of searching the latent space of a model, the image generation is directly conditioned on the text prompt (Rombach et al., 2022). We use this setup as an example to identify the following places where automation could be introduced, based on our framework. We highlight concrete techniques and references for automation from the literature. While the notebook comprises a fairly simple text-to-image system, the same ideas of automation are potentially applicable to larger commercial systems, like OpenAI's ChatGPT and Google Gemini. These systems likely already implement similar strategies for autonomous decision-making, albeit not for creative responsibilities, but rather in content moderation and the selection of specialised synthesis models.

In terms of *pre-trained model selection*, numerous people have substituted BigGAN with other GANs generators. This creative responsibility could be automated, with the system choosing from a database of models and installing new ones into the notebook. In terms of *data preparation and curation*, users often choose imaginative text prompts, as the notebook often produces high-quality, surprising results for these. This could be substituted, for example, with automated fictional ideation techniques (Llano et al.,

Figure 3.5: Image generated by the *Big Sleep* Colab notebook for the prompt "The Melbourne skyline in pastel colours". Note the appropriate presentation of content and style, and additional pastel strokes in the sky as an unprompted innovation.

2016). The author of the Colab notebook, Murdock (2021), innovated in *loss function definition*, employing patches from generated images rather than the entire image to evaluate its fit to the prompt. Various image manipulation routines could be automatically tested within loss function calculations from a library, with the system automatically altering the notebook at the code level. As described in Colton et al. (2021), in some circumstances where multiple images are being generated simultaneously, increasing the learning rate can help searches fail quickly. Such *hyper-parameter tuning* could be automated using standard AutoML techniques, guided by requirements on acceptable search successes and output image quality. In terms of *model selection and deployment*, we can imagine models being used as creative web services (Veale, 2013), with higher-level CC systems accessing text-to-image generators in a variety of projects. While using Colab notebooks like the *Big Sleep*, people cherry-pick results for posting on social networks and in blogs, effectively doing *output curation*. This would be an ideal target for automation with systems using CLIP and other techniques to evaluate images, also possibly inventing new aesthetic measures (Colton, 2008a).

## 3.3 DISCUSSION

In this chapter, we introduced the term *active divergence* to describe a common theme in the artistic uses of generative deep learning (DL). Artists often consciously break, tweak or otherwise intervene in data-driven generative processes in order to produce artefacts, that are culturally valuable, but sub-optimal from a pure modelling perspective. For illustration, we presented an introductory overview of some active divergence techniques, many of which require human supervision of important tasks and decisions. We identified two avenues for the use of generative models in an artistic setting: creativity support tools and autonomous creative systems. In the latter, the extent of human supervision and decision-making is reduced and the system is given more creative responsibilities and capabilities for automation.

Following this idea, we presented a framework for the specific purpose of automating manual tasks in a generative DL pipeline for artistic projects. For this, we adopt core concepts of AutoML and adjust them in two ways. First, we focus on generative DL which differs in the type of learning task, in that it is concerned with modelling the distribution of a training set, rather than classification or regression. Second, we address the artistic usage of generative DL, where more emphasis is given to the qualities of the generated output over the qualities of the model. The specialisation of our framework inversely limits its generalisability in the same ways. On the one hand, there might be artefact-driven applications of generative DL within or outside CC that we have not considered. On the other hand, our framework is not generally applicable to generative approaches in DL due to its special emphasis on artistic uses. Its focus on generative DL further limits its validity for other generative modelling methods.

We have previously analysed the close relationship between the *automation* of generative DL systems and the central CC goal to increase a system's *creative autonomy* (Jennings, 2010; Guckelsberger, Salge & Colton, 2017; McCormack, Gifford & Hutchings, 2019) by granting it more *creative responsibilities* (Colton, 2008b). Here, we complement the earlier analysis with

knowledge of our concrete automation pipeline. The aim is to understand to which extent our proposed pipeline already enables facets of creative autonomy, and how CC insights on creative autonomy could be used to advance it in future work.

Automation is necessary for creative autonomy, but the opposite does not hold: while a fully automated generative DL system might still exactly follow user-prescribed goals, an autonomously creative system has the 'freedom to pursue a course independent of its programmer's or operator's intentions' (Jennings, 2010). This firstly requires the system to autonomously *evaluate* its creations, which is satisfied by any system that can be considered *creative* (Ventura, 2016). In addition, an *autonomously creative* system must be capable of autonomous *change*, i. e. initiating and guiding 'changes to its standards without being explicitly directed when and how to do so' (Jennings, 2010). To prevent trivial implementations of these capabilities, Jennings requires them to not exclusively rely on random decisions.

To assess how much our pipeline realises creative autonomy, we can draw on various CC approaches to enhancing autonomy in computational systems. For instance, Colton (2009) proposes 'repeatedly asking ourselves: what am I using the software for now? Once we identify why we are using the software, we can [...] write code that allows the software to use itself for the same purpose. If we can repeatedly ask, answer and code for these questions, the software will eventually [...] create autonomously for a purpose, with no human involvement'. Our framework provides various candidate targets to perform such a gradual elevation of a generative DL system.

For the evaluation of a concrete system built under our framework, we consider the FACE model (Colton, Charnley & Pease, 2011; Pease & Colton, 2011) an adequate evaluation tool. In this evaluation model, systems are described in terms of the creative acts they perform. Such an analysis allows for the identification of newly added creative responsibilities through automation.

Linkola et al. (2017) follow a more constrained approach and, as part of a larger agenda to realise meta-creativity in CC, propose that creative autonomy requires *artefact-awareness*, *goal-awareness* and potentially *generator-*

*awareness*, realised through operators of *(self-) reflection* and *(self-) control* which closely match Jennings' (2010) requirements for evaluation and change. Whether a system built within our framework satisfies these definitions depends on the extent to which it is granted responsibilities in the form of automated decision-making for targets identified in the framework. We demonstrate this based on extensions to a non-automated generative DL system. Such a system can be considered to have some generator-awareness due to the role of its loss function (self-reflection), and the adjustments of its own parameters through error correction methods like back-propagation (weak self-control). A system's control over changes to its generator can be increased from weak to strong within our framework, through the automated manipulation of network architecture or the selection of a pre-trained model. Further putting a system in charge of its loss function within our framework (strong control) affords it goal-awareness, such that it can be considered autonomously creative if it is capable of modifying the loss function in response to its evaluation of generated output.

Crucially, more radical forms of creative autonomy do not eliminate co-creation, i. e. cut ties with the system user entirely, but facilitate different forms of interaction. To really become independent of its designer, a system must not be isolated but interact with critics and creators that shape its evaluation and changes (Jennings, 2010). A fully creatively autonomous system might refuse the will of its interaction partner (Jennings, 2010; Guckelsberger, Salge & Colton, 2017), but we believe that this holds a promise for innovative artistic collaborations between people and computational systems, connecting artistic practices in generative DL with the philosophy and goals of CC.

Having discussed the potential and benefits of generative models for creative and artistic work, in the next chapter, we focus on how conventional modelling approaches are limited for such applications as a generator's expressivity depends on the dataset it was trained on.

Chapter 4

# LIMITATIONS OF CONVENTIONAL GENERATIVE MODELLING

When a generative model is used to produce artefacts, the variety of its output is bound by the expressivity of the model. We posit that this holds both for simple random sampling, as well as searching a model's latent space, e. g. with a quality diversity (QD) algorithm. In any case, the possibility to find artefacts of both high *fidelity* and *diversity* is determined by the properties of the learned latent space, which in turn is conditioned on the training dataset. We hypothesise that the expressivity of generative models is limited by their statistical nature.

In this chapter, we address RQ 2 (*How are the conventional generative modelling approaches limited in terms of output diversity?*) through a principled study of the expressivity of a variational auto-encoder (VAE). We define *expressivity* as the variety of different outputs that can be produced by a model, quantified by a diversity measure. The objective of our study is to analyse the generative capabilities of VAEs and give empirical evidence for its limitations. For this, we define a parametric design space for a simple two-dimensional shape generation task. We then compare the output diversity of QD search in the parametric space as a baseline against the search in a model's latent space. From our findings, we derive practical recommendations for the use of VAEs in generative systems, in particular in combination with QD search algorithms.

The work in this chapter was presented at GECCO 2021:
Hagg, A., Berns, S., Asteroth, A., Colton, S., & Bäck, T. (2021). Expressivity of Parameterized and Data-driven Representations in Quality Diversity Search. *Proceedings of the Genetic and Evolutionary Computation Conference*.

CONTENTS

## 4.1 INTRODUCTION

While engineering-driven design optimisation looks for solutions to technical problems, artistic practices are usually more concerned with generating culturally valuable artefacts than optimising for a specific objective. For example, let us consider shape optimisation. An engineering-driven design process might seek to optimise the shape of an aeroplane wing in terms of aerodynamics. Whereas a type designer, in addition to functional aspects, might also be concerned with the stylistic aspects of a typeface and its glyphs. An artist, however, might be completely unconcerned with function. Yet, these two approaches are more similar than the seeming differences in disciplines and objectives would suggest. Architects and engineers often use the output of a design optimisation tool at the beginning of the design process to survey the space of possibilities, where underlying parameters can have complicated correlations (Bradner, Iorio & Davis, 2014). Candidate solutions are then expanded or contracted upon in an iterative design loop. Similarly, artists might set up an evolutionary system to find initial inspiration and continue to tweak their system towards a desired outcome through the iterative adjustment of the fitness function. In both workflows, the diversity of the generated population is key to illustrating the range of possibilities. We propose that initial diversity is the basis for the potential of later discoveries. Focusing on only one optimal individual too early would limit the chances of encountering unexpected candidate solutions. Evolutionary multi-solution approaches such as quality diversity (QD) algorithms have been developed for this purpose of divergent search (Lehman & Stanley, 2011).

Generative models such as variational auto-encoders (VAEs) (Kingma & Welling, 2014) find application in this context for their ability to extract patterns from raw data, learn representations for data examples that preserve semantic relations and reliably produce more samples with similar properties. Disentangled representation learning can furthermore equip a model's latent space with linearly separated factors of variation (Burgess et

al., 2017), revealing the underlying parameters of a generative process. The resulting feature compression network provides meaningful descriptors or encodings to be used in QD algorithms (Cully, 2019; Gaier, Asteroth & Mouret, 2020; Hagg, Preuss et al., 2020). Defining similar descriptors by hand is a non-trivial task which requires expertise and intuition and, depending on the domain, often cannot compete with an automated solution (Hagg, Preuss et al., 2020). While the advantage of learning from data lies in the recognition of complex patterns, we hypothesise that the expressivity of the resulting generative model is entirely dependent on the quality and *representativeness* of the data samples provided. That is to say, how well do data examples represent the range of variations in a given domain? This is especially critical when relying on such a model to produce novel examples and diverse sets of outputs. In fact, artists who employ generative adversarial networks (GANs) in their work often use a variety of strategies to *actively diverge* from the intended purpose of these models and to produce outputs significantly different from the original data (see Chapter 3 for a detailed discussion). To the best of our knowledge, there does not exist sufficient evidence on the capabilities and limitations of QD search in learned latent spaces. We address this gap with a principled study of a simple generative system, aiming to derive from our findings specific recommendations for the use of learned latent spaces in QD search, as well as evidence for the limitations of statistical generative modelling approaches in terms of output diversity.

We compare the performance of multi-solution evolutionary search in the parameter space of a generative system with the search in the learned latent space of a VAE that was trained with examples from the same system. For this, we use a simple point-symmetric shape generation task and define a parametric design space of black-and-white two-dimensional shapes. To generate shapes, we build a generative system combining a variational autoencoder (VAE) with Voronoi Elites (VE) search, a modification of the MAP-Elites algorithm. An example of the resulting artefact sets (see Section 4.4.2) produced by the two search methods is depicted in Figure 4.11. While the latent space is built from a limited dataset, the parameter space represents

the full range of the system's possible output. The purpose of this work is to understand how expressive either of the two search spaces is and, from this knowledge, to derive recommendations for their usage. As an application domain for our study, we choose to generate two-dimensional point-symmetric black-and-white shapes, because of their simplicity and ease of visualisation. While more complex domains might be closer to actual applications, they would make the presentation of our results less accessible. Shape is an important basic design element in art, architecture, engineering, as well as graphic and industrial design. On the one hand, shapes can carry semantic meaning (e. g. letters of a font) and on the other hand, define the properties and visualise the form of a physical object in engineering-driven design (e. g. the cross-section of a wing optimised for aerodynamic flow). Other possible but more complex domains (e. g. sequences of movements of a robotic arm) would make the interpretation of our results more difficult.

Our work is relevant in two scenarios: (A) when the generative search space is manually defined but a VAE is used to compare artefacts (e. g. distance/similarity estimation), and (B) when only data is available and its underlying generative patterns are unknown or too difficult to extract manually and thus have to be approximated by latent variables to obtain a searchable generative space.

Our study makes the following contributions:

1. In the context of the first scenario (A), we give informed recommendations of how to use a VAE to its full capacity in combination with a QD search algorithm. We test whether the latent space is suitable for evaluating artefact similarity and also for searching artefacts or whether the two steps should be performed in separate spaces.

2. For both scenarios (A and B), we give evidence for the limitations of VAEs in their ability to represent and generate examples beyond the original training data and, as a result, the diversity of their output.

In the following, we describe techniques for latent space search for artefact generation, quality diversity (QD) search in general and VE in particular. We then lay out our methodology and study setup, including details on the

shape generation task, the generative system and the evaluation procedure. We then present four benchmark tasks in two different experiments. We follow with a discussion of our findings and close with conclusions.

## 4.2 ARTEFACT GENERATION VIA LATENT SPACE SEARCH

Generative machine learning algorithms model a given data distribution to reliably synthesise data examples with high fidelity. For this, an artificial neural network is conventionally used to map from a latent space, typically a simple and well-understood distribution, to the complex feature space of the data domain. The latent variables ideally encode the factors of variation that underlie the process which generated the data. Such latent spaces can be leveraged for artefact production in several ways. While random sampling is the simplest and computationally inexpensive approach, it does not allow for any control over the output. When specific criteria exist, searching the latent space of a generative model can be more effective. For this, there exist several search strategies, most of which are designed to optimise single solutions, focusing on *quality* only. While the goal of global optimisation is to find a single global optimum in the parameter (or genotypic) space, multimodal optimisation (Deb et al., 2002; Mouret, 2011; Deb & Saha, 2012) provides a set of (potentially only locally) optimal solutions. In contrast, quality diversity (QD) algorithms typically operate in the feature space, comparing phenotypic representations, and aim to cover the entire search space by finding the highest-performing solutions in all local neighbourhoods, even if they are not of optimal fitness globally.

QD search approaches are particularly relevant for artistic and creative applications, where the goal is not necessarily to find a single optimal artefact. Given a variety of several different high-quality proposals, the human in the loop can make an informed selection of artefacts for further design iterations. For some applications, practitioners are therefore not interested in full automation but use optimisation tools to assist in the design process. In this context, QD has been shown to produce more diverse sets of artefacts than

multi-criterion and multi-modal optimisation techniques (Hagg, Preuss et al., 2020; Hagg, 2021).

In this section, we describe the conceptual approach of quality diversity (QD) search in general and the Voronoi Elites (VE) search algorithm used in this work. We follow the conventional evolutionary terminology in the context of artefact generation, where *genotype* (*genomes* or *chromosomes*) and *phenotype* refer to the parametric and visual representations of an artefact, respectively. The *population* of all candidate artefacts is made up of *individuals* that change over various optimisation steps. The *archive* is the collection of selected artefacts. In multi-solution evolutionary search, the search space is divided into *niches* for local competition.

### 4.2.1 QUALITY DIVERSITY

Just as other evolutionary computation methods, quality diversity (QD) (Pugh et al., 2015; Cully & Demiris, 2018b) takes inspiration from natural evolution and its dynamics of competition and adaptation. These dynamics can be used to tackle complex optimisation problems. But instead of promoting competition between all individuals in a population, QD formulates the evolutionary process as a divergent search. In nature, not all species compete with each other for the same resources and many can simply co-exist. QD therefore proposes for individuals to compete locally within niches of specialisation. This allows for the simultaneous optimisation and diversification of artefacts. QD thus focuses both on *quality* and *diversity*.

QD search is performed in parameter space (Figure 4.1), but individuals are evaluated based on their phenotypic representation, i. e. artefacts. Specific measures can be defined to determine the features of individual artefacts, e. g. to capture some aspects of behaviour or morphology which are important for a task. Examples of such features include the proportion of time that each leg is in contact with the ground in a walking robot's gait (Cully et al., 2015), the turbulence in the airflow around an aerodynamic shape or its surface area (Hagg, Wilde et al., 2020). An important decision

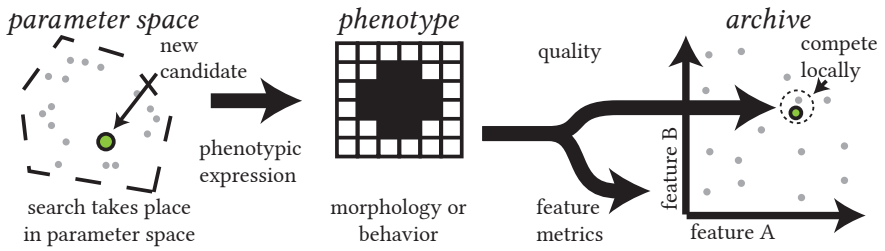



Figure 4.1: Local competition in QD. Search is performed in parameter space. Candidate solutions are converted from their genetic into their phenotypic representation, i. e. from parametric descriptors into artefacts. Candidates compete locally in feature space and are only added to the archive if they improve the quality score compared to their immediate neighbourhood of individuals.

is the definition of local neighbourhoods, a process called *niching*. In MAP-Elites (Mouret & Clune, 2015), it is common to divide the search space into a fixed regular grid of niches. Competition between individuals only takes place within the same niche, the local neighbourhood of artefacts. An individual is added to the archive if it survives local competition. New individuals are created by selecting surviving candidates from the archive and modifying their genome, either through *mutation*, by adding small perturbations, or *crossover* with the genome of another individual.

### 4.2.2 VORONOI ELITES

In this work, we use Voronoi Elites (VE) (Hagg, Preuss et al., 2020), a modification of the MAP-Elites algorithm (Mouret & Clune, 2015). *Elites* is the common term for high-performing individuals (candidate solutions in a population). For the setting of our study, VE provides some advantages over other QD algorithms. In MAP-Elites, the search space is divided into a fixed grid of *niches*, phenotypic sub-spaces for local competition. With a growing number of phenotypic feature dimensions, this approach leads to an exponential growth of niches. CVT-Elites (Vassiliades, Chatzilygeroudis & Mouret, 2017) deals with this problem by pre-defining fixed niches using a Voronoi tessellation of the search space. However, due to the fixed number of individuals in an archive, both methods tend to reduce the variance of the population in the first iterations. On the one hand, random initial indi-

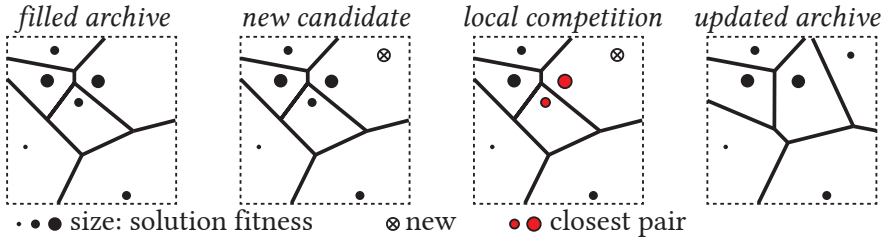



Figure 4.2: Updating the Voronoi archive. The size of the dots indicates fitness, new individuals are marked with a cross and pairs of closest individuals are marked red. The VE approach allows for a fixed archive size, independent of its dimensionality, making experiments more controllable. In this example, the maximum number of niches is set to six. When a new candidate individual is added to the archive, the pair of closest individuals is compared. The worse of the two is removed from the archive and the individual with higher fitness is kept in the archive. The borders between niches drawn here are for visualisation only, to illustrate the range of influence of individuals and how they are changed by archive updates.

viduals typically do not cover the entire search space. On the other hand, competition is higher in niches with a large number of initial individuals, leading to the early exclusion of many candidate solutions. In contrast, in VE niches are not pre-defined, and any new individual is added to the archive until a maximum number of niches is reached. Only then local competition is initiated, and the phenotypically closest elites are compared, removing the worst-performing individuals from the archive. VE is therefore more effective in maximising the number of available individuals to be used as training examples for the VAE.

The evolution of an VE archive is illustrated in Figure 4.2. Selection pressure is applied based on artefact similarity. VE effectively minimises the variation of distances between artefacts in an unbounded archive. The total number of niches is fixed, independent of the number of search space dimensions. As a result of local competition, the division of the search spaces into niches may change from one iteration to the next. Tournament selection is used to select individuals from the archive. New individuals are created by mutation, drawn from a normal distribution.

## 4.3 METHODOLOGY AND SETUP

This section outlines the details of our study's subject domain (generation of two-dimensional point-symmetric shapes), lists the general configurations of the VAE and VE algorithm (specific settings for experiments can be found in the experimental setups below) and explains how the two methods are combined to build two versions of a generative system which we compare in a principled way through a series of experiments.

### 4.3.1 SHAPE GENERATION TASK

For our study, we use a simple shape-generation task. We define a parametric design space of black-and-white two-dimensional shapes (Figure 4.3). This allows us to define specific benchmarking tasks in our experimental setup (Section 4.4). Shapes are generated by connecting eight control points which can be freely placed in a Euclidean plane with a polar coordinate system. Each control point is thus defined by two parameters: the radial ($dr$) and angular coordinates ($d\theta$). Following the standard evolutionary terminology, these overall 16 parameters act as genes, encoding the properties of the individuals. For evaluation, individuals need to be converted from their genetic representation into their phenotypic representation. To form a smooth outline, the points are connected by locally interpolating splines (Catmull & Rom, 1974). A discretisation step renders this smooth shape onto a square grid resulting in a bitmap of $64 \times 64$ pixels.

### 4.3.2 EVALUATION AND FITNESS

For the evaluation of shapes in the evolutionary fitness criterion, we use point symmetry. For our simple generative system, this objective is computationally inexpensive and easy to understand and interpret.

The symmetry error of an artefact is calculated in five steps. First, the shape boundary in the bitmap image is determined as a collection of $N$

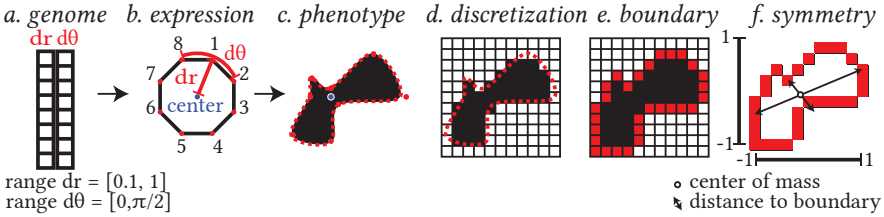



Figure 4.3: Shape encoding, representation, conversion and evaluation: (a) 16 genes define the position of (b) eight control points with polar coordinates in a Euclidean plane. (c) Smooth outlines are formed by locally interpolated splines. (d) A shape is converted from its genetic into its phenotypic representation through a discretisation step that renders the smooth shape onto a square grid of $64 \times 64$ pixels, producing a bitmap image representation. The quality of a shape is evaluated by first (e) determining its boundary and then (f) measuring its symmetry from the centre of mass.

pixels. Second, the coordinates of the boundary pixels are normalised to the range $[-1, +1]$ to remove any influence of the shape's scale. Third, the boundary's centre of mass is determined to be used as the centre of point symmetry. Fourth, we calculate the Euclidean distance between all $M = \frac{N}{2}$ pairs of pixels, that are opposite each other across the centre. Finally, we calculate the overall symmetry error $S$ as the sum of the distances of all opposing pixel pairs. For a perfectly symmetric shape, this sum equals zero. The fitness function thus $F$ is calculated as follows:

$$F(\mathbf{x}) = \frac{1}{1 + S(\mathbf{x})} \qquad S(\mathbf{x}) = \sum_{i=1}^{M} \|\mathbf{x}_i - \mathbf{x}_{i+M}\| \tag{4.1}$$

### 4.3.3 GENERATIVE SYSTEM DESIGN

For the generation of two-dimensional point-symmetric black-and-white shapes, we build a generative system where a VAE is trained from scratch and its latent space is searched with the Voronoi Elites (VE) algorithm. This generative system is similar to AutoVE (Hagg, Preuss et al., 2020), except that it uses a VAE instead of a conventional auto-encoder. VAEs provide several advantages: they distribute training examples more evenly in latent space, can learn disentangled latent dimensions and allow to generate

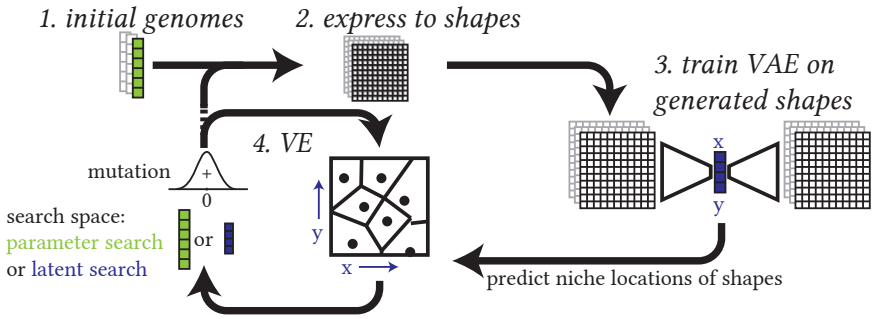



Figure 4.4: We combine a VAE and VE into a generative system in two phases. First, initialisation: (1) an initial set of genomes is generated and (2) converted into shape bitmaps which are used to (3) train a VAE. We compare two initialisation scenarios: starting from scratch with random initialisation ($R$) and continuation ($C$) where the system starts with a pre-determined set of candidates, e. g. from a previous run. Second, optimisation loop: (4) VE iteratively updates the archive of candidates. We compare two setups of this loop: the VE performs search either in parameter space ($PS$) or in the VAE latent space ($LS$).

new samples via interpolation. The full generative system, illustrated in Figure 4.4, can be separated into two phases: (1) initialisation and (2) an evolutionary optimisation loop.

At initialisation time, a set of genomes (parameter configurations) are set randomly and converted into their phenotypic counterpart (bitmap images). The VAE is trained until convergence on this bitmap data. The learned latent space is then used in the following evolutionary process and the model's encoder and decoder networks serve as mapping functions between the phenotypic bitmap representations and the model's latent representations and vice versa. We compare two initialisation scenarios: starting from scratch with random initialisation ($R$) and continuation ($C$) where the system starts with a pre-determined set of candidates, e. g. from a previous run.

In the evolutionary optimisation loop, the VE algorithm iteratively updates the archive of artefacts, increasing its diversity as well as the quality of artefacts in individual niches through local competition. For this, two candidates are compared to each other in the VAE's low-dimensional latent representation space (artefact genome), which preserves semantically meaningful artefact relations.

For the central comparison of our study, we perform this optimisation process in two different search spaces: (1) parameter space (the explicit genome encoding) and (2) the VAE's latent space (the learned representation). This way, we can evaluate the expressivity of the VAE latent space and its capability to generate a diverse set of artefacts in comparison to the full space of possibilities which is reflected by the 16 predefined genetic parameters. The performance of the two approaches is measured in terms of the diversity of the produced set (see the following Section 4.3.4). This setup allows us to study the limitations of the VAE latent space and compare it to the baseline diversity when searching for candidate artefacts over the complete parametric search space.

### VAE CONFIGURATION

Throughout this work, we employ VAEs with a beta-annealing loss term (Burgess et al., 2017) that regularises the latent distribution to be a Gaussian with independent dimensions that capture linearly separable factors of variation in the data.

$$\mathcal{L} = \mathop{\mathbb{E}}_{q_\phi(z|x)} \left[\log p_\theta(x|z)\right] - \beta \left| D_{\mathrm{KL}}\big(q_\phi(z|x) \,\|\, p(z)\big) - \gamma \right| \qquad \beta \geq 1 \qquad \gamma \geq 0 \tag{4.2}$$

The scaling factor of the regularisation loss (second right-hand-side term) was set to $\beta = 4$. The annealing factor $\gamma$ controls the capacity of the latent space information bottleneck. Throughout the training, it increased from 0 to 5. As the capacity is increased, the encoder learns to encode latent dimensions in the order of decreasing returns to the log-likelihood over the data. The most important information for reconstruction is thus encoded first. This results in better reconstruction quality compared to standard Beta-VAE (Higgins et al., 2016) while achieving similar levels of disentanglement.

We use a model's encoder as a mapping from phenotype bitmaps to genetic latent representations. The encoder network is made up of four downscaling blocks, each consisting of a convolution layer (8, 16, 32 and 64 filters

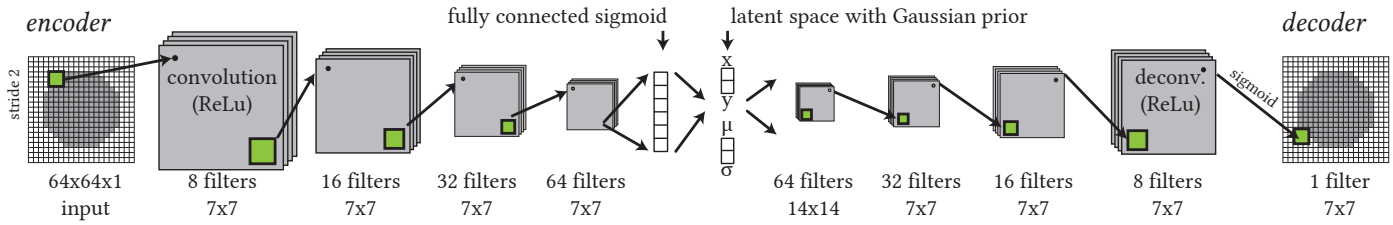



Figure 4.5: Architecture of a convolutional variational autoencoder.

respectively; kernel size $7 \times 7$; stride 2) followed by a ReLU activation function. The set of blocks is followed by a final fully-connected layer. The decoder network inversely maps from the genetic latent space to phenotype bitmaps through five transposed convolution layers, which have 64, 32, 16, 8 and 1 filter respectively, kernel size $7 \times 7$ and stride 2, except for the first layer which has a kernel size of $14 \times 14$. The last layer is responsible for outputting the correct size ($64 \times 64$ pixels). The weights of both networks are initialised with the Glorot scheme (Glorot & Bengio, 2010). Each model was optimised with the Adam optimiser (Kingma & Ba, 2015) with a learning rate $\mu = 0.001$ and a batch size of 128.

### VORONOI ELITES CONFIGURATION

We configure VE to start with an initial set of samples, generated from a Sobol sequence (Sobol, 1967) in parameter space. Sobol sequences are quasi-random and evenly space-filling. They decrease the variance in the experiments but ensure that the sampling is similar to a uniform random distribution and easily reproducible. In all experiments, VE runs for 1,024 generations with a population size of 32 individuals per generation, which remains constant over the entire experiment. We perform random mutation, but no crossover of genes. To produce children via mutation for the next generation, parents are randomly selected from the population. A small random perturbation is then added to the parent's genes. Mutation vectors for perturbation are drawn from a normal distribution $\mathcal{N}(0, 0.01)$ centred around zero with small scale.

### 4.3.4 DIVERSITY MEASURE

As a measure of diversity, in this work, we use Pure Diversity (PD) (H. Wang, Jin & Yao, 2017). Originally proposed as a measure of biological diversity (Solow & Polasky, 1994), it has since been applied to multi-objective optimisation (Ulrich, Bader & Thiele, 2010) and the evaluation of high-dimensional phenotypes (Hagg, Preuss et al., 2020). PD computes the diversity of a set from the pairwise distances between its items and thus does not depend on any additional information. In contrast, other measures of diversity are often domain-dependent or require taking one of the QD algorithms as a baseline (Hagg, Preuss et al., 2020). See Section 2.5 for an overview of approaches to measuring diversity. Archive-dependent measures, like the QD-score, do not generalise well and, by splitting the search space, introduce biases. We therefore rely on a distance-based diversity measure that is calculated on the expressed shapes directly. Here, we thus measure diversity directly, by calculating the PD of sets of bitmaps with binary pixel values, independent of their representation in parameter space or the VAE latent space. The PD score of a set $A$ is calculated recursively and is equal to the maximum of the sum of its value on all but one of the members and the minimum distance of that member to the set. The algorithm is effectively a sub-graph search for the longest distance between subsets of artefacts and their nearest neighbours.

$$\mathrm{PD}(A) = \max_{a \in A} \left( \mathrm{PD}(A \setminus \{a\}) + \mathrm{d}(a, A \setminus \{a\}) \right) \tag{4.3}$$

As a distance measure between an individual artefact and a collection, we use the Hamming distance with $L^{0.1}$-norm, recommended for high-dimensional cases (H. Wang, Jin & Yao, 2017), to find the minimum dissimilarity between an individual item and the items in a set $X$.

$$\mathrm{d}(y, X) = \min_{x \in X} L^{0.1}(x, y) \tag{4.4}$$

## 4.4 EXPERIMENTS

It is commonly assumed that generative models, such as VAEs, have good interpolation and reasonable extrapolation capabilities. Their latent spaces are thus potentially appealing search spaces for the synthesis of novel artefacts. Yet, to the best of our knowledge, there does not exist sufficient evidence on the capabilities and limitations of learned latent spaces, in particular in terms of output diversity when used as QD search spaces. We aim to alleviate this gap and gain insight into two research questions:

1. How accurately can a VAE represent a variety of artefacts? That is to say, how useful are its latent representations for artefact comparison?

2. How well can a VAE generate unseen shapes? Can their latent space be used to reliably find novel artefacts?

Our experimental setup, as described in detail above, consists of a two-dimensional shape generation task. We define a parametric encoding which serves as a genetic representation of artefacts. As baseline performance, we measure the output diversity of QD search in this parametric space ($PS$). The capabilities of the VAE are evaluated by training a model on samples covering the parametric space and performing QD search in its latent space. For this, we define one baseline and four benchmark tasks with corresponding datasets that we evaluate in two experiments (Figure 4.6). The datasets in the first experiment (tasks b–d) consist of samples which have been produced by varying two generating factors: scale and rotation. All experiments were performed with five different base shapes. Below we present the baseline dataset and four benchmark tasks.

(a) With a *baseline dataset*, the complete set of samples, we evaluate the standard reconstruction error of the model to determine the baseline quality of latent representations.

(b) In the *recombination task*, we leave out a subset of artefacts in the centre of the ranges of values of both generating factors, leaving sufficient examples at either end of the value ranges.

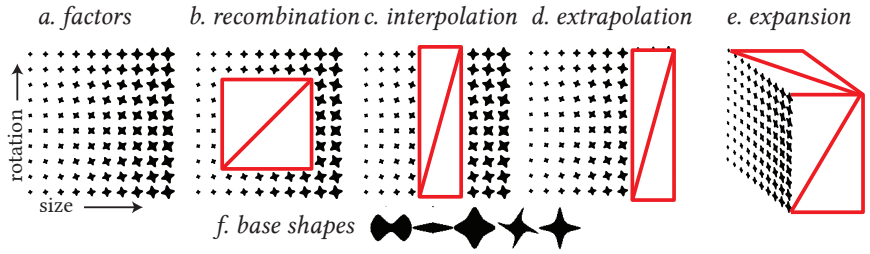



Figure 4.6: (a) Generative factors used to create datasets in this work. (b–e) Four tasks on which we compare the performance of latent space search with parameter search, the red rectangles indicate artefacts that either have been left out of a dataset (b, c, d) or are not available (e). (f) All base shapes used in this work. For illustration, visualisations here only show 100 of the 256 shapes.

(c) In the *interpolation task*, the left-out subset of artefacts covers the complete range of one of the two generating factors, while some examples remain at both ends of the other factor's range of values.

(d) The *extrapolation task* consists of omitted samples at one end of values of one factor of variation, which affects the complete range of the other factor.

(e) The *expansion task* focuses on generating artefacts beyond the two given factors of variation in the complete dataset.

We expect the VAE to perform reasonably well in recombining (b), interpolating between (c) and extrapolating beyond (d) the available variations to reproduce the samples missing from the training datasets. In the expansion task (e), we expect the VAE latent space to only produce artefacts of low fidelity.

### 4.4.1 EXPERIMENT 1: RECOMBINATION, INTERPOLATION AND EXTRAPOLATION

We define five different base shapes (Figure 4.6, f). For each base shape, we build a dataset covering two factors of variation. By scaling the base shape by a factor from 0.1 to 1 and rotating from 0 to $\frac{1}{2}\pi$ degrees in 16 steps each, we obtain a dataset of 256 total shapes. The complete dataset is our baseline for comparison (Figure 4.6, a). For the other tasks, we create

separate datasets by omitting a subset of examples from the complete dataset (Figure 4.6, b–e). We then train four separate VAE, one model per task on the corresponding dataset: (a) baseline model, (b) recombination, (c) interpolation, and (d) extrapolation. The models are trained for 3,000 epochs, after which we choose the models with the lowest validation error (calculated on 10% of the input data).

To determine whether the VAE can correctly reproduce, and thus properly represent, the given shape, we measure the models' reconstruction errors. For the baseline model, this is done over the complete dataset. For the task models (b–d), the reconstruction error is calculated only on the held-out examples. We define the reconstruction error as the Hamming distance between an input bitmap and a generated bitmap, normalised by the total number of pixels. As bitmap pixels are either black or white, i. e. 0 or 1, Hamming distance is the appropriate choice of distance measure. A high reconstruction error indicates that the model cannot properly generate a shape, and consequently that its latent space does not provide an adequate search space for VE. Generating shapes to which there are no corresponding training examples, the reconstruction errors of unseen shapes that can be created with recombination and interpolation (b and c) are expected to be lower than for extrapolation (d).

To determine the resolution of the models, we measure the distances in the latent space between the training examples for the baseline model and between the training and the unseen examples for the task models (b–d). If the latter are of a similar order of magnitude as the first, the models can distinguish unseen shapes from the training examples and each other. This would indicate that the model's resolution is high enough to provide features of sufficient quality to perform a VE search.

This experiment was performed separately on each of the five base shapes (Figure 4.6f) and for three different sizes of the VAE latent space (4, 8, and 16 dimensions), as we assumed that the model would not able to perfectly learn the two generating factors. The results are reported as statistics over the total 15 runs.

### 4.4.2 EXPERIMENT 2: EXPANSION

The last task, expansion (e), cannot be treated as per the previous experiment, because we cannot properly define an a priori ground truth shape set 'outside' of latent space. Instead, we compare the two search spaces (parameter: *PS*, and latent: *LS*) using the framework proposed in Figure 4.4. We measure which one of the two search spaces produces the most diverse set of artefacts using the PD score (Section 4.3.4). The experiment is split up into two configurations: random initialisation (*R*) and continuation (*C*).

In the first configuration (*R*), both of the compared search approaches start from the same random initial set of genomes, which is common in many optimisation problems. We increased the size of the archive to 512, as this experiment poses a more difficult optimisation problem. As in the first experiment (Section 4.4.1), the genomes are translated into bitmaps, which serve as the training data for a VAE model. VE is then performed in both search spaces to fill two separate archives of 512 shapes each. The resulting shape sets are compared with respect to their diversity and average fitness, which are often in conflict with each other. As the translation from genome to bitmap always produces a contiguous shape, it is reasonable to expect that a VAE would learn to produce shapes, and not only random noise, even when starting with a randomly generated set of examples.

Oftentimes, however, a generative system does not start with a random set of data, but rather a pre-defined set of examples. For example, these can be artefacts observed in the real world, manually designed or the output of a previous search run. We thus define a second configuration, continuation (*C*), to test such a scenario. This way, we can compare whether the overall output diversity improves when a VAE is trained on a set of high-quality artefacts from a previous VE search process. We use the archive of shapes produced by parameter search (*PS*) from the random configuration (*R*) as training data for a new VAE model. Both *PS* and *LS* are then performed again with this improved model.

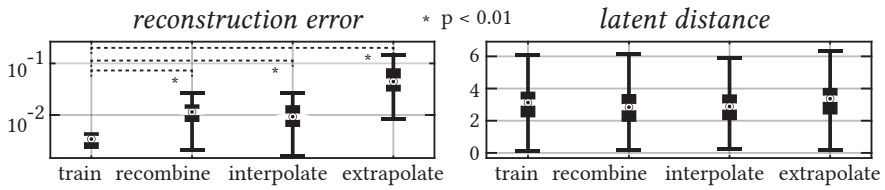



Figure 4.7: Reconstruction errors (log scale) and latent distances (linear scale) for tasks (a) through (d) over all models across all five base shapes and three different latent space sizes (4, 8, 16 dimensions). Box plots show median values, 25th and 75th percentiles and whiskers indicating minimum and maximum values. All tested differences were statistically significant (two-sample t-test, $p < 0.01$) and are marked with an asterisk.

It is expected that *LS* will interpolate between training samples, but not be able to expand beyond the generative factors in the data, except through modelling errors. Since *PS* is performed in the encoding's parameter space, this search approach should be able to produce a higher diversity of artefacts in both configurations *R* and *C*.

The number of latent dimensions of the VAE has been set to 8, 16 and 32 to analyse the influence of the degrees of freedom in latent space when it is lower than, equal to, or higher than the number of parameters of the genome representation. A higher number of degrees of freedom gives an advantage to latent space search, a lower number would give it a disadvantage. When using 16 latent dimensions, VE deals with the same dimensionality in *PS* and *LS*. For this task, the number of filters in the VAE is quadrupled to give the model a better chance at learning the higher number of variations.

This experiment has been repeated 10 times for each of the four configurations: (1) random initialisation *R* in *PS*, (2) continuation *C* in *PS*, (3) *R* in *LS* and (4) *C* in *LS*. We report results as statistics over 10 repetitions for each latent space size (Figure 4.10).

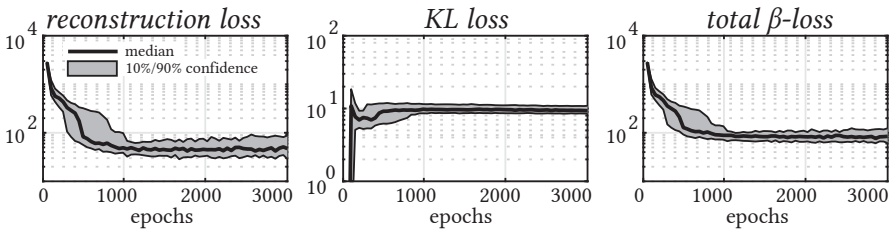



Figure 4.8: VAE validation losses during training on 10 % held-out validation data. Curves show median values and the 10/90 % confidence intervals.

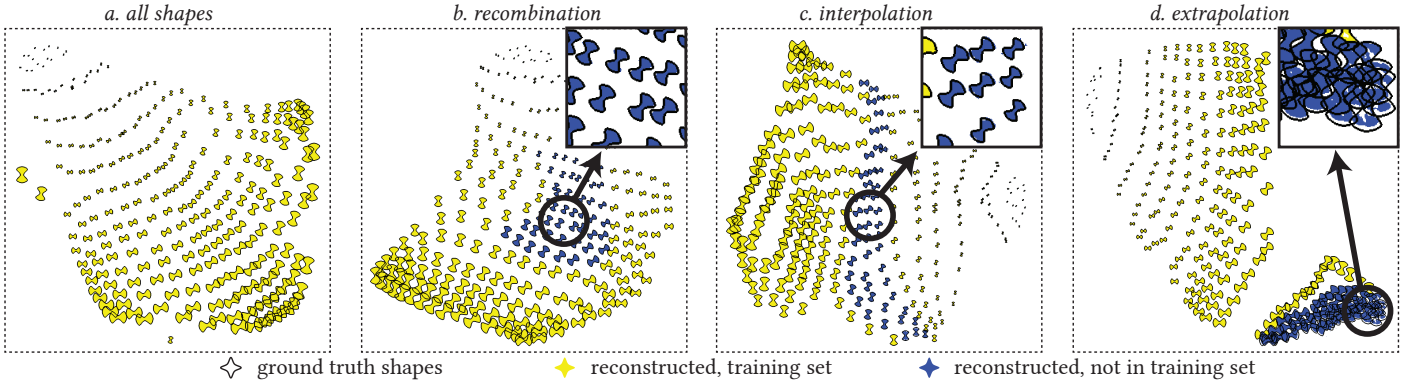



Figure 4.9: Visualisation of the VAE latent spaces (eight dimensions projected down to two with t-SNE). Shapes in yellow represent training examples, while blue ones are from the task's hold-out set. All shapes were reconstructed by the model. Black outlines show the ground truth shapes and coloured fills the reconstructed shapes. Differences between the outlines and fillings correspond to reconstruction errors.

## 4.5 RESULTS

For the first experiments, covering the recombination, interpolation and extrapolation tasks, we report the reconstruction loss, KL loss and total $\beta$-loss on the validation data, during training of the models (Figure 4.8). The training does not need much more than 1,000 epochs until convergence. We further report the reconstruction errors and latent distances between examples for all tasks (Figure 4.7). The reconstruction error on the full training dataset is lower than when reproducing the held-out recombination and interpolation examples. On average, 1 % of pixels (approximately 40 out of 4,096) were incorrectly reconstructed in the recombination and interpolation tasks. As expected, the error in the extrapolation task is the highest. All differences between the reconstruction error on the whole dataset and the hold-out sets are statistically significant (two-sample t-test, $p < 0.01$). The latent distances between the examples have similar distributions across all tasks. Four exemplary latent spaces corresponding to the different datasets and tasks are shown in Figure 4.9 from models with a latent space size of eight dimensions. For visualisation, the artefact positions were projected to two dimensions using the dimensionality reduction method t-SNE (van der Maaten & Hinton, 2008). The disentangled lat-

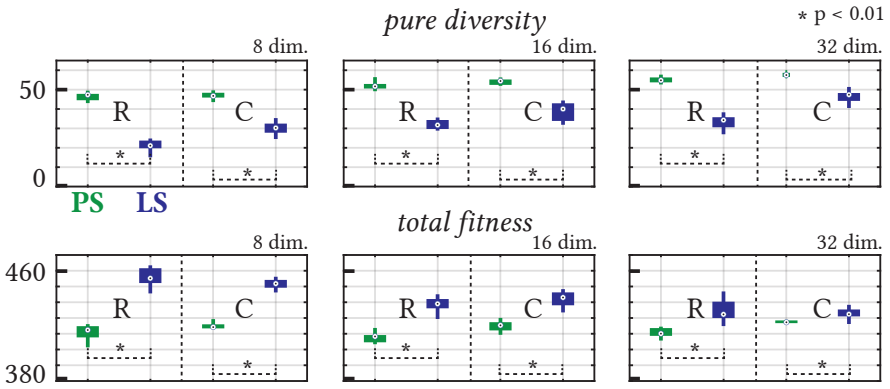



Figure 4.10: Pure diversity (top) and total sum of fitness (bottom) of artefact sets of parameter search (*PS*, green) and latent space search (*LS*, blue). VAEs were separately trained with 8, 16 and 32 latent dimensions (subplots). In every subplot, the two left-hand bars correspond to random initialisation (*R*) and the two right-hand to the continuation (*C*) configurations of the experiments. Box plots show median values, 25th and 75th percentiles and whiskers indicating minimum and maximum values. All tested differences were statistically significant (two-sample t-test, $p < 0.01$) and are marked with an asterisk.

ent spaces accurately capture the factors of variation in the data, arranging shapes by size and rotation. Shapes from the hold-out sets in the recombination and interpolation tasks (b and c) are correctly placed in a gap between training examples according to their size and rotation. In comparison, in the extrapolation task (d), hold-out shapes are not as neatly arranged and show significant reconstruction errors. Detailed reconstruction errors are shown in Figure 4.13.

For the expansion task in the second experiment, we compare the Pure Diversity (PD) and total fitness of the generated artefact sets from parameter search (*PS*) and latent space search (*LS*) (Figure 4.10). We can observe some interesting patterns:

1. The pure diversity of *PS* is significantly higher than *LS* across all configurations. In turn, *LS* produces artefacts with higher levels of fitness (point symmetry of shapes).

2. While diversity increases with bigger latent spaces, total fitness only decreases for latent space search (*LS*) and remains relatively stable for parameter search (*PS*). A higher dimensional latent space makes latent space search more difficult.

3. The performance gap between *PS* and *LS* decreases as the latent space size increases and when continuing the search from an updated model (configuration *C*). This observation provides us with two important insights:
   a) In *LS*, there is a trade-off between fitness and diversity which can be controlled via the latent space size.
   b) The continuation scenario provides a model with a more diverse training dataset, which leads to higher PD.
4. In all configurations, the tested differences between random initialisation (*R*) and continuation (*C*) are statistically significant (two-sample t-test, $p < 0.01$).

An example of the output shapes from *PS* and *LS* are shown in Figure 4.11 to illustrate the effective difference in pure diversity and point symmetry. Analogous to our previous hypothesis (Section 4.4.2), we interpret a shape's reconstruction error as its distance to the model's latent space manifold. We visualise the expansion away from the latent surface of shapes found by *PS* in Figure 4.12. For this, the *PS* and *LS* artefact positions in the 16-dimensional latent space are reduced to two dimensions using t-SNE.

## 4.6 DISCUSSION

For the interpretation of our results, we will discuss two separate relevant aspects: in particular, (1) the use of learned latent spaces in QD search approaches and, more generally, (2) the limitations of generative models trained on raw data.

The models tested in our first experiment reliably reproduce previously unseen shapes through recombination and interpolation. As expected, the more difficult task, extrapolation beyond the given training examples, results in higher reconstruction errors. The latent distances of all four task datasets have similar distributions. This suggests that, even when VAEs are not able to fully reproduce the extrapolated shapes, they can still distinguish

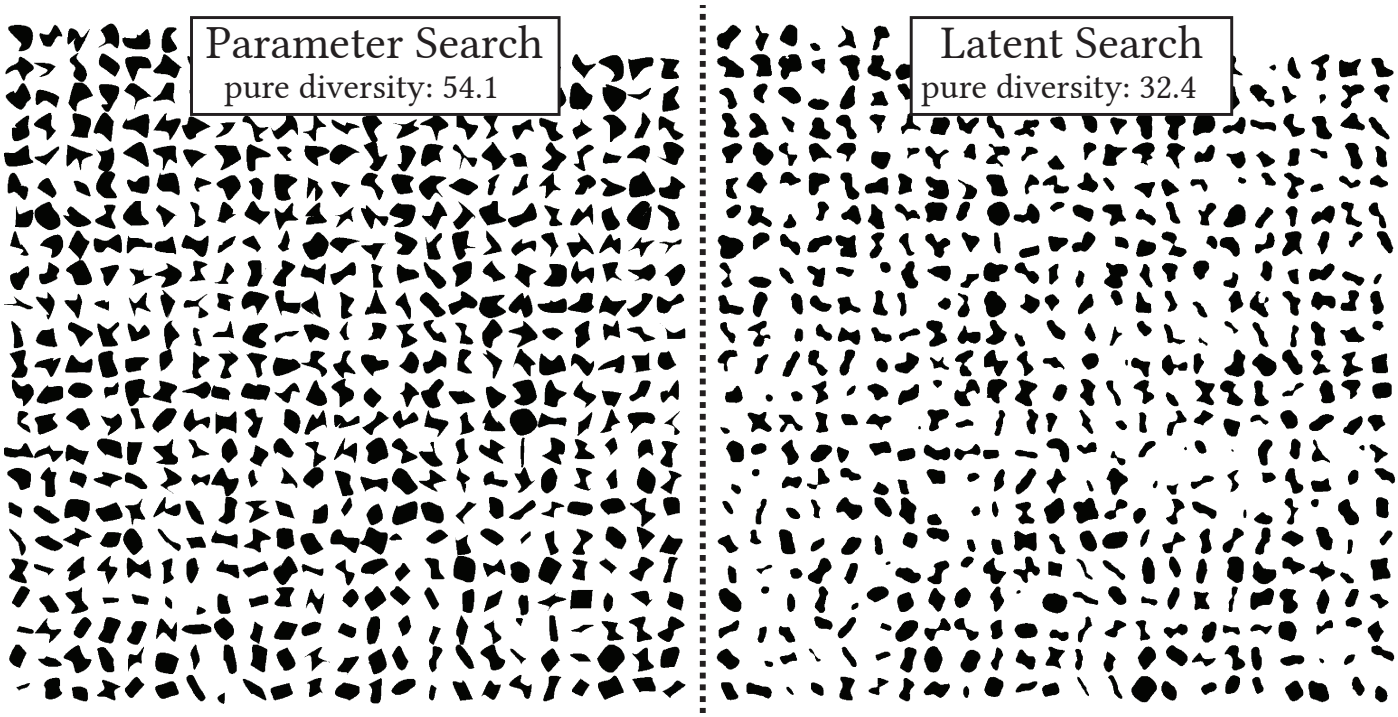



Figure 4.11: Searching the parameter space produces a more diverse set of artefacts than searching the VAE latent space. In both cases, the same VAE latent dimensions were used as niching dimensions of the QD algorithm. Artefacts shown here (512 total) represent the complete VE archive from a single run with one of the base shapes.

them from the training data and each other. That is to say, the position of shapes in the latent space reflects their semantic relations, as visualised in Figure 4.9. While shapes are not perfectly reconstructed in the extrapolation task, they are nonetheless positioned in a well-structured relation to others. Based on these observations, we recommend using a VAE latent space as an approximate measure of similarity.

Our second experiment addresses the most difficult task, expansion beyond the generative factors in the dataset. The ability of a VAE to find new shapes is indirectly measured by comparing the pure diversity of the artefact sets created by a search in parameter space and latent space (*PS* and *LS* respectively). The diversity of *PS* is significantly higher than that of *LS* (Figure 4.10). This holds as the number of latent dimensions is increased beyond the number of degrees of freedom in the original encoding, and when the VAE is updated after a first VE run (*C*). Although a trade-off between diversity and fitness is expected, it becomes less pronounced in the

32-dimensional model. Given this evidence, we conclude that parameter search finds a more diverse set of artefacts than latent search (Figure 4.11). Our findings suggest, that manually-defined parametric encodings are more expressive than learned latent representations, and should therefore be preferred as QD search spaces, whenever available.

Our work, however, is limited to a simple generative task with a continuous but range-restricted search space of 16 dimensions. It is possible that with increasing complexity the benefits of search in parameter space, which covers the full generative space, are outweighed by the curse of dimensionality. That is to say, parameter search is only useful in settings where it can perform well. The authors of related work have come to the conflicting conclusion that search in the latent space of a GAN trained on simplified levels from the video game Overcooked yields more diverse output than MAP-Elites search on the manually-defined tile-based level encodings (Fontaine, Hsu et al., 2021). The parametric search space differs from ours in two important aspects. First, it is a combinatorial search problem. Second, its size is much larger with 8 possible tile types in $15 \times 10 = 150$ tile locations. We hypothesise that the size of the search space in particular limits the performance of parametric search in this setting. While the number of parameters can grow infinitely and parametric search can quickly become infeasible, a learned latent search space for the corresponding type of artefacts would maintain a constant size and remain comparatively small. The regularisation through compression, and potentially disentanglement, in the generative model's learned mapping from feature to latent space, might provide an important advantage to a search algorithm. The biggest question for future work is thus, whether latent search spaces can outperform parametric search, and at what point on the scale of increasing parametric complexity this happens. We describe possible follow-up experiments in Section 8.1.

Limitations

We further acknowledge the following limitations to our work. The present study is only meaningful to problem settings which allow for multiple solutions because they are most relevant to artistic and creative applications. Our findings are limited to multi-modal continuous domain

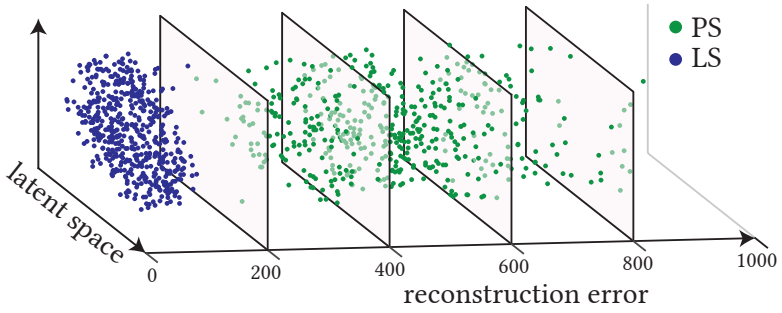



Figure 4.12: Expansion in a 16-dimensional latent model (projected to two dimensions with t-SNE). We interpret the reconstruction error of a shape as its distance from the latent surface. Samples from parameter search (*PS*, green) tend to extrapolate away from the latent distribution (*LS*, blue).

optimisation, as we focus on latent space search. Extensions to other optimisation problems, e. g. combinatorial search, are not included in this study and are left for future work. The presented problem setting, two-dimensional shape generation, is kept simple for ease of visualisation and interpretation. Some application domains are much more complex and more work is needed to confirm whether our findings hold. Yet, our approach allows us to evaluate a generative model and quantify its limitations through specific recombination, interpolation, extrapolation and expansion tasks. This would not be as straightforward in more complex settings.

Summary

In summary, in this chapter, we have presented a principled study on the capabilities and limitations of generative models, in particular when using their learned latent spaces as a base for divergent search methods like the VE algorithm. Our findings quantify the ability of VAEs to generate samples through recombination, interpolation and extrapolation within and expansion beyond the distribution of a given dataset. We compare the diversity of generated artefacts when VE is run either in a parametric encoding space or a learned latent space. Our findings show that the pure diversity of artefact sets generated by latent space search is significantly lower than that of parameter space search.

While we hypothesise, given the conflicting account from related work, that these observations might not hold with the increasing complexity of the parametric search space, we believe that the limitations of VAEs continue to hold at scale. In very high-dimensional domains for which we can collect

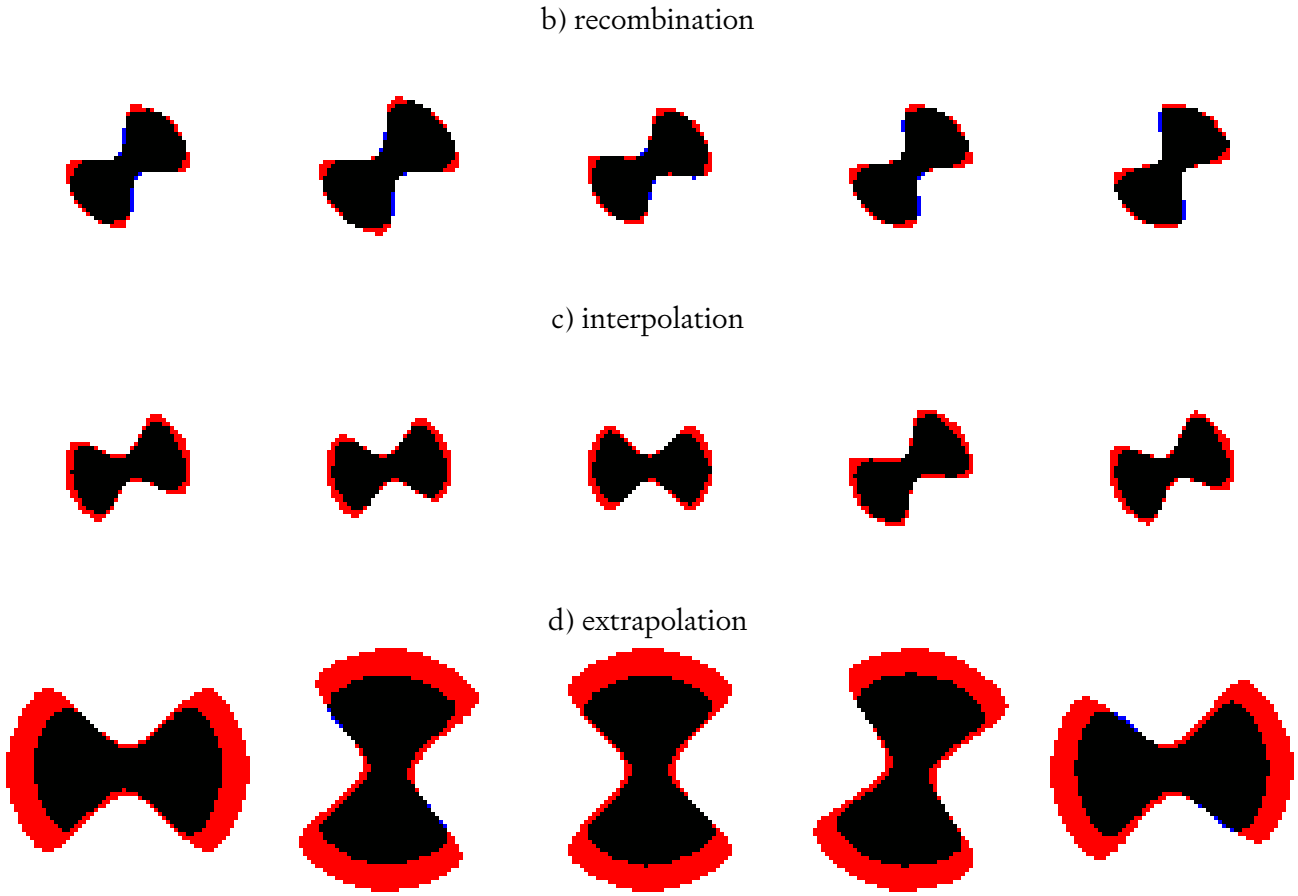



Figure 4.13: Five worst model reconstructions (blue) of left-out shapes (red) from each task b–d (top to bottom rows). Overlaps (black) indicate pixels that were correctly reconstructed. Reconstruction errors are shown as red (not reconstructed) and blue pixels (erroneously generated)

large datasets, like image and video data, latent space search might already afford a vast space of possibilities — likely sufficient for most applications. Yet, on a fundamental level, generative models remain limited. Due to their statistical nature, generative models adopt, integrate and propagate any issues that stem from a given dataset, be it in terms of biases, under-representation or limitations of data coverage. We have to assume that generative models are always limited by a given training dataset, which biases models towards the most prominent features therein.

In this chapter, we presented evidence for the limitation of generative models through a principled study of the expressiveness of a VAE's latent space. In the following chapter, we argue that this shortcoming in the conventional formulation of generative models is limiting to artistic applications and ef-

forts of diversity, equity, and inclusion (DEI). We present *diversity weights*, a method to increase the baseline output diversity of a generative model through *mode balancing*.

Chapter 5

# INCREASING THE OUTPUT DIVERSITY OF GENERATIVE MODELS

In this chapter, we turn to improving generative modelling methods towards higher model output diversity, addressing RQ 3. We first review the conventional modelling objective, *mode coverage*, and contrast it with the failure state *mode collapse*. We then propose *mode balancing* as an alternative objective for generative machine learning in artistic and creative applications. We present our approach to mode balancing, *diversity weights*: a diversity-weighted sampling scheme for model training. The effect of diversity-weighted training on model performance is demonstrated in a proof-of-concept study on handwritten digits, which highlights the trade-off between artefact diversity and typicality.

We motivate this work by bringing together the conclusions from previous chapters. We highlighted the common goal in relevant applications of generative models to produce a diverse range of output (Chapter 3), in particular for artistic and creative work. We provided evidence for the limited expressivity of generative models in terms of diversity (Chapter 4). We aim to address this limitation by increasing the output diversity of generative models, and in order to cater to specific artistic and creative applications. We further connect this goal with efforts in diversity, equity, and inclusion (DEI) as data biases often negatively affect under-represented groups. We identify and address the specific data imbalance bias in unsupervised learning.

The work in this chapter was presented at ICCC 2023:
Berns, S., Colton, S., & Guckelsberger, C. (2023). Towards Mode Balancing of Generative Models via Diversity Weights. *Proceedings of the 14th International Conference on Computational Creativity (ICCC)*.

Code available at https://github.com/sebastianberns/diversity-weights.

CONTENTS

## 5.1 INTRODUCTION

Large image generation models (LIGMs), in particular as part of text-to-image generation systems (Ramesh et al., 2021; Saharia et al., 2022), have been widely adopted by visual artists to support their creative work in art production, ideation, and visualisation (Ko et al., 2023; Vimpari et al., 2023). While providing vast possibility spaces, large models like LIGMs, trained on huge image datasets scraped from the internet, not only adopt but often exacerbate data biases, as observed in word embedding and captioning models (Bolukbasi et al., 2016; Zhao et al., 2017; Hendricks et al., 2018). The tendency to emphasise majority features and to primarily reproduce the predominant types of data examples can be limiting for many CC applications that use machine learning-based generators (Loughran & O'Neill, 2017). Learned models are often used to illuminate a possibility space and to produce artefacts for further design iterations. Examples range from artistic creativity, like the production of video game assets (Liapis, Yannakakis & Togelius, 2014; Volz et al., 2018), to constrained creativity, e. g. industrial design and architecture (Bradner, Iorio & Davis, 2014), and to scientific creativity, such as drug discovery (Madani et al., 2023). Many of these and similar applications would benefit from higher diversity in model output. Given that novelty, which underlies diversity, is considered one of the essential aspects of creativity (Boden, 2004; Runco & Jaeger, 2012), we expect that a stronger focus on diversity can also foster creativity (Stanley & Lehman, 2015).

Most common modelling techniques, however, follow a distribution-fitting paradigm and do not accommodate the goal of higher diversity. Within this paradigm, one of the primary generative modelling objectives is *mode coverage* (Zhong et al., 2019), i. e. the capability of a model to generate all prominent types of examples present in a dataset. While a conventionally trained model can in principle produce many types of artefacts, it does not do so reliably or evenly. A model's probability mass is assigned in accordance with the prevalence of a type of example or feature in a dataset. Common ex-

amples or features have a higher likelihood under the model than rare ones. As a consequence, samples with minority features are not only less likely to be obtained by randomly sampling a model, but they are also of lower fidelity, e. g. in terms of image quality. Related studies on Transformer-based language models (Kandpal, Wallace & Raffel, 2022; Razeghi et al., 2022) have identified a "superlinear" relationship: while training examples with multiple duplicates are generated "dramatically more frequently", examples that only appear once in the dataset are rarely reproduced by the model.

In this work, we argue for an adjustment of modelling techniques from mode coverage to *mode balancing* to enrich CC with higher output diversity. Our approach allows us to train models that cover all types of training examples and can generate them with even probability and fidelity. We present a two-step training scheme designed to reliably increase output diversity. Our technical contributions are:

- *Diversity weights*, a training scheme to increase a generative model's output diversity by taking into account the relative contribution of individual training examples to overall diversity.
- *Weighted Fréchet Inception Distance (wFID)*, an adaptation of the FID measure to estimate the distance between a model distribution and a target distribution modified by weights over individual training examples.
- A proof-of-concept study, demonstrating the capability of our method to increase diversity, examining the trade-off between artefact diversity and typicality.

In the following sections, we first introduce the objective of *mode balancing* and highlight its importance for CC based on existing frameworks and theories. Then, we present our *diversity weights* method in detail, as well as our formulation of *Weighted FID*. Following this, we present the experimental setup and methodology of our study and evaluate its results. In the discussion section, we contribute to the debate on issues of DEI in generative machine learning more generally, and CC specifically, by explaining how our method could be beneficial in addressing data imbalance bias.

## 5.2 MODE BALANCING

Generative deep learning models now form an integral part of CC systems (Berns et al., 2021). A lot of work on such models is concerned with *mode coverage*: to match a data distribution as closely as possible by accurately modelling all types of examples in a dataset (Figure 5.1). In the specific case of GANs, great effort is put into preventing *mode collapse* (Arjovsky, Chintala & Bottou, 2017), a training failure state in which a model disregards important modes and is only able to produce a few types of training examples. Mode coverage is captured formally in common evaluation measures such as FID and PR. Crucially, this is always done in reference to the training set statistics or data manifold. In this context, diversity is often arguably misused to refer to mode coverage. While mode coverage describes the fraction of modes in a dataset that are represented by a model, the diversity of a model's output, if understood more generally and intuitively, can theoretically be higher than that of the dataset.

Mode coverage is conceptually similar to the notion of *typicality* (Ritchie, 2007). Defined as the extent to which a produced output is "an example of the artefact class in question", a model which only generates outputs with high typicality, if sampled at random, has to provide the most support to those training set examples with the highest density of features characteristic of that artefact, i. e. to maximise mode coverage. Crucially, sampling from the model would resemble going along the most well-trodden paths in the possibility space defined by the dataset and, as Ritchie (2007) already suggests, counteract novelty as a core component of creativity (Boden, 2004; Runco & Jaeger, 2012).

Crucially, mode balancing breaks with the convention of viewing the dataset as 'ground truth'. Instead, we consider the dataset to provide useful domain information and the characteristics of *typical* examples (Ritchie, 2007). But a data distribution does not have to be matched exactly. Particularly in artistic applications, creators often strive to *actively diverge* from the typical examples in a dataset (Berns & Colton, 2020; Broad et al., 2021). To

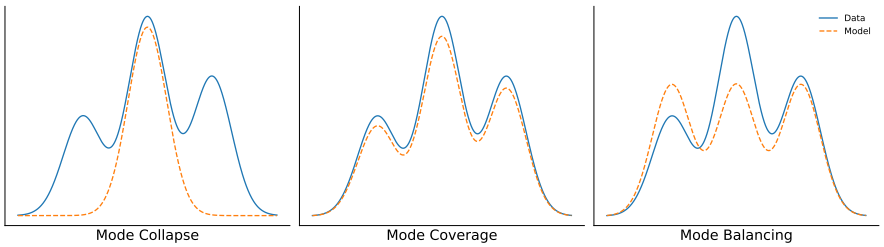



Figure 5.1: *Mode collapse*: the model does not cover all modes in the data distribution. *Mode coverage*: the data distribution's modes are modelled as closely as possible w.r.t. their likelihood. *Mode balancing*: the model covers all modes but with equal likelihood.

stay with our metaphor, borrowed from Veale, Cardoso and Pérez y Pérez (2019), *mode balancing* allows us to walk more along the less trodden paths and thus especially support exploratory and transformational creativity (Boden, 2004; Stanley & Lehman, 2015). In contrast to the mode coverage paradigm, in mode balancing, diversity is measured independently of the training data distribution. In the theoretical case of a balanced dataset of absolutely dissimilar examples, i. e. multiple equally likely modes, our method would assign uniform weights to all examples and thus be identical to standard training schemes with random sampling.

## 5.3 THE VENDI SCORE

We adopt the Vendi Score (VS) as a measure of diversity to evaluate datasets and compare model performance (Friedman & Dieng, 2023). Here, we first describe its general form and then introduce its probability-weighted formulation which we employ in our work.

Given a set of artefacts $\{x_1, x_2, \ldots, x_N \mid x_i \in \mathcal{X}\}$, the VS is defined as the exponential of the Shannon entropy of the eigenvalues of the normalised pairwise similarity matrix over all artefacts:

$$\mathrm{VS}(\mathbf{K}) = \exp\Big(-\sum_{i=1}^{N} \lambda_i \log \lambda_i\Big) \tag{5.1}$$

Where $\mathbf{K}$ is a positive semi-definite similarity matrix ($N \times N$) between pairs of artefacts such that $\mathbf{K}_{ii} = 1$, and $\lambda_1, \lambda_2, \ldots, \lambda_N$ are the eigenvalues

of the normalised similarity matrix $\mathbf{K}/N$. The eigenvalues can be obtained via the eigendecomposition $\mathbf{K}/N = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$ as the diagonal elements of the diagonal matrix $\lambda_i = \mathbf{\Lambda}_{ii}$.

The similarity matrix $\mathbf{K}$ is obtained by computing pairwise distances over all artefacts with a similarity function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. While the next Chapter 6 covers human-aligned similarity estimation, in the present chapter we obtain pairwise similarities differently for two reasons. First, the work presented here (Berns, Colton & Guckelsberger, 2023) predates the work on similarity estimation (Berns et al., 2024). Second, due to the difference in the generative domains of the two works: our study on similarity estimation was performed with tile-based video game levels, whereas the present work focuses on image generation, in particular the synthesis of hand-written digits.

Exponential entropy, also known as *perplexity*, can be used to measure how well a probability model predicts a sample. Low perplexity indicates good prediction performance. Consequently, the more diverse a sample, the more difficult its prediction, the higher the perplexity and its VS. This can be shown with a simple coin flip example. A coin has two sides, thus allowing for one out of two outcomes: heads or tails. Typically, we deal with fair coins such that both outcomes have equal probability: 1/2. In this case perplexity is 2, equal to the number of possible outcomes. This is extendable to N-sided dice, where its perplexity is equal to N if all sides are equally likely. Outcomes that are sampled from a uniform probability distribution are the most difficult to predict, consequently both entropy and perplexity are maximised. If any of the outcomes has higher probability, entropy and perplexity decrease. VS, in this sense, can be considered an effective number, as it quantifies the number of absolutely dissimilar examples in a dataset.

### 5.3.1 PROBABILITY-WEIGHTED VENDI SCORE

For our work, we use the probability-weighted formulation of the VS to define a probability distribution $\mathbf{p}$ over all artefacts. Here, the similarity

matrix is normalised by the probability distribution instead of the number of artefacts: $\mathbf{K_p} = \text{diag}(\sqrt{\mathbf{p}})\,\mathbf{K}\,\text{diag}(\sqrt{\mathbf{p}})$. The VS is then calculated as previously defined from the eigenvalues of the probability-weighted similarity matrix (Equation 5.1).

We employ this probability-weighted formulation to determine the contribution of each artefact to overall diversity (Section 5.4). While a dataset may contain many artefacts, its diversity is low if the artefacts are similar to each other. Diversity can be increased by composing a dataset of as many artefacts as possible that are as dissimilar from each other as possible. There is thus an inverse relationship between the relative abundance of a type of artefacts in a dataset and its contribution to diversity. The more similar artefacts there are, the less each of them adds to the overall diversity. We illustrate this relationship with an example in the following section.

### 5.3.2 ILLUSTRATIVE EXAMPLE

The probability distribution $\mathbf{p}$ represents the relative abundances of individual artefacts in a dataset. Instead of repeating identical artefacts in a set, their prevalence can be expressed with higher probability. For illustration, we present an example of four artefacts, of which three are absolutely similar to each other and one is absolutely dissimilar to all others. All have equal probability.

$$\mathbf{K}^a = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \quad \mathbf{p}^a = \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix} \tag{5.2}$$

The same information can be reduced to two absolutely dissimilar artefacts and the corresponding probabilities $\mathbf{p}^b$.

$$\mathbf{K}^b = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{p}^b = \begin{pmatrix} 0.25 \\ 0.75 \end{pmatrix} \tag{5.3}$$

Both representations yield the same VS, which reflects the imbalanced set of two absolutely dissimilar artefacts. $\text{VS}(\mathbf{K}^a_\mathbf{p}) = \text{VS}(\mathbf{K}^b_\mathbf{p}) = 1.755\ldots$

The imbalance in our example set negatively affects its diversity. If all items in the set are given equal importance, one artefact is under-represented. Instead, each of the two absolutely dissimilar artefacts in the set should thus be assigned equal weight $p = 0.5$. In the case of repetitions, this weight has to be divided across the repeated artefacts.

$$\mathbf{K}^c = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \quad \mathbf{p}^c = \begin{pmatrix} 0.5 \\ 0.166... \\ 0.166... \\ 0.166... \end{pmatrix} \tag{5.4}$$

This maximises VS to reflect the effective number of absolutely dissimilar artefacts $\mathrm{VS}(\mathbf{K}^c_{\mathbf{p}}) = 2$.

## 5.4 DIVERSITY WEIGHTS

If artefacts in a set are repeated, i. e. their relative abundance is increased, their individual contribution to the overall diversity of the set decreases. Yet, with uniform weighting, all artefacts contribute to the model distribution equally (Equation 5.2). Instead, we aim to adjust the weight of individual artefacts in a set in accordance with their contribution to overall diversity.

We formulate an optimisation problem to find the optimal weight for each artefact in a set, such that its diversity, as measured by VS, is maximised.

$$\begin{aligned} &\max\ \mathrm{VS}(\mathbf{K}_{\mathbf{p}}) = \max \exp\Big(-\sum^{n} \lambda_i \log \lambda_i\Big) \\ \text{s.t. } &0 \le p_i \le 1 \qquad \sum^{n} p_i = 1 \\ \text{where } &\mathbf{p} = (p_i, \ldots, p_n), \quad p_i \in \mathbb{R}^{[0,1]} \\ &\mathbf{K} \in \mathbb{R}^{n \times n}, \quad \mathbf{K}_{ii} = 1 \\ &\mathbf{K}_{\mathbf{p}} = \mathrm{diag}(\sqrt{\mathbf{p}})\, \mathbf{K}\, \mathrm{diag}(\sqrt{\mathbf{p}}) = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{-1} \\ &\boldsymbol{\lambda} = \mathrm{diag}(\boldsymbol{\Lambda}) = (\lambda_1, \ldots, \lambda_n) \end{aligned} \tag{5.5}$$

### 5.4.1 WEIGHTED SAMPLING

Conventionally, training examples are drawn from a dataset with uniform probability. In our method, examples are instead chosen according to their

**Algorithm 1** Vendi Score Diversity Weight Optimisation

**Input**: Pairwise similarity matrix $\mathbf{K}$ over $N$ artefacts
**Parameter**: Loss term balance $\gamma$, num iterations $I$, learning rate $\alpha$, Adam hyperparams $\beta_1, \beta_2$

Initialise $\mathbf{w} = (w_1, \ldots, w_N)$, where $w_i = 1$
**for** $i = 0$ **to** $I$ **do**
  $\mathbf{p} \leftarrow \mathbf{w} / \sum w_i$
  $g \leftarrow -\nabla_{\mathbf{p}} \gamma \mathrm{VS}(\mathbf{K}, \mathbf{p}) - (\gamma - 1)\,\mathrm{H}(\mathbf{p})$
  $\mathbf{w} \leftarrow \mathrm{Adam}(\mathbf{w}, g, \alpha, \beta_1, \beta_2)$
**end for**

**Output**: Weight vector $\mathbf{w}$

contribution to an unknown target distribution. The weight of training examples is determined by their individual contribution to the overall dataset diversity as quantified by the optimised probability distribution $p$ (see example above, Section 5.3.2). We aim to increase the output diversity of a model. For this, during model training, we replace the basic data sampling operation by a diversity-weighted sampling scheme.

### 5.4.2 OPTIMISATION ALGORITHM

We compute an approximate solution to the optimisation problem via gradient descent (Algorithm 1). The objective function consists of two terms: a diversity loss and an entropy loss. The diversity loss is defined as the negative probability-weighted VS of the collection of artefacts, given the similarity matrix between artefacts and the corresponding probability distribution over artefacts (Section 5.3.1). Instead of optimising the artefact probabilities directly, we optimise a weight vector $\mathbf{w}$. The probability vector $\mathbf{p}$ is obtained by dividing the $\mathbf{w}$ by the sum of its values, which guarantees the second axiom. To satisfy the first axiom, we implement a fully differentiable version of VS in log space. Optimising in log space enforces weights above zero, since the logarithm $\log x$ is only defined for $x > 0$ and tends to negative infinity as $x$ approaches zero. However, if the weights have no upper limit, values can grow unbounded. A heavy-tailed weight distribution negatively affects the diversity-weighted sampling step of our method during training, as batches can become saturated with the highest-weighted training examples, causing

overfitting. We therefore add an entropy loss term $\mathrm{H}(\mathbf{p}) = -\sum p_i \log(p_i)$ to be maximised in conjunction with the diversity loss. The entropy loss acts as a regularisation term over the weight vector, such that its distribution is kept as close to uniform as possible. The emphasis on the two loss terms is balanced by the hyperparameter $\gamma \in [0, 1]$.

$$\mathcal{L} = -\gamma \mathrm{VS}(\mathbf{K}, \mathbf{p}) - (\gamma - 1)\,\mathrm{H}(\mathbf{p}), \qquad \mathbf{p} = \frac{\mathbf{w}}{||\mathbf{w}||_1} \tag{5.6}$$

Given a normalised data matrix $X$ where rows are examples and columns are features, we obtain the similarity matrix $\mathbf{K}$ by computing the Gram matrix $K = X \cdot X^\mathsf{T}$. The weight vector $\mathbf{w}$ is initialised with uniform weights $w_i = \log(1) = 0$. The probability vector $\mathbf{p}$ is obtained by dividing the weight vector $\mathbf{w}$ by the sum of its values. We choose the Adam optimiser (Kingma & Ba, 2015) with default hyper-parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We experimentally determine the best initial learning rate $\alpha = 0.1$ with a decay every $5$ iterations by a factor of $0.99$. An implementation of the optimisation algorithm is available at:
https://github.com/sebastianberns/diversity-weights

### 5.4.3 WEIGHTED FID

The performance of generative models, in particular that of implicit models like GANs, is conventionally evaluated with the FID (Heusel et al., 2017). Raw pixel images are embedded into a representation space, typically of an artificial neural network. Assuming multi-variate normality of the embeddings, FID then estimates the distance between the model distribution and the data distributions from their sample means and covariance matrices.

In our proposed method, however, the learned distribution is modelled on a weighted version of the dataset. Moreover, referring to the standard statistics of the original dataset is no longer applicable, as the weighted sampling scheme changes the target distribution. We therefore adjust the measure such that it becomes the Weighted Fréchet Inception Distance (wFID), where the standard mean and covariances to calculate the dataset

statistics are substituted by the weighted sample mean $\bar{x}$ and the weighted sample covariance matrix $\mathbf{C}$.

$$\begin{aligned} \bar{x} &= \frac{1}{\sum w_i} \left( \sum w_i \, \mathbf{x}_i \right) \\ \mathbf{C} &= \frac{1}{\sum w_i} \left( \sum w_i (\mathbf{x}_i - \bar{x})^\mathsf{T} (\mathbf{x}_i - \bar{x}) \right) \end{aligned} \tag{5.7}$$

Note that the statistics of the model distribution need to be calculated without weights as the model should have learned the diversity-weighted target distribution.

## 5.5 PROOF-OF-CONCEPT STUDY ON HAND-WRITTEN DIGITS

We show the effect of the proposed method in an illustrative study on pairs of handwritten digits (Lecun et al., 1998; LeCun, Cortes & Burges, 2010). While artistically not particularly challenging, digit pairs have several benefits over other exemplary datasets. First, the pairings of digits create a controlled setting with two known types of artefacts. Second, hand-written digits present a simple modelling task, in which the quality and diversity of a model's output is easy to visually assess. And third, generating digits is fairly uncontroversial. While, for example, generating human faces is more relevant for the subject of diversity, it is also a highly complex and potentially emotive domain.

### 5.5.1 METHODOLOGY

For individual pairs of digits, we quantitatively and qualitatively evaluate the results of GAN training with diversity weights and compare it against standard training. Experiments are repeated five times with different random seeds.

*Digit Pairs* From the ten classes of the MNIST training set, we select three digit pairs: 0-1, 3-8, and 4-9, which represent examples of similar and

Table 5.1: Vendi Score (VS) of digit pair datasets (mean ± std dev) with uniform and diversity weights with different loss balances $\gamma$

| VS weights | MNIST digit pairs | | |
|---|---|---|---|
| | Pair 0-1 | Pair 3-8 | Pair 4-9 |
| Uniform weights | 1.77±0.003 | 1.96±0.004 | 2.07±0.004 |
| DivW ($\gamma = 0.6$) | 2.13±0.020 | 2.64±0.016 | 2.65±0.010 |
| DivW ($\gamma = 0.8$) | 2.79±0.052 | 3.45±0.027 | 3.38±0.025 |
| DivW ($\gamma = 1.0$) | 3.08±0.046 | 3.67±0.023 | 3.60±0.023 |

dissimilar pairings. For example, images of hand-written zeros and ones are easy to distinguish, as they are either written as circles or straight lines. In contrast, threes and eights are both composed of similar circular elements.

*Balanced Datasets* For each pair of digits, we create five balanced datasets (with different random seeds) of 6,000 samples each. Each dataset consists of 3,000 samples of either digit, randomly selected from the MNIST training set. We compute features by embedding all images using the CLIP ViT-L/14 model. To optimise the corresponding diversity weights, we obtain pairwise similarities between images by calculating the Gram matrix of features.

*Diversity Weights* For each dataset (5 random draws per digit pair), we optimise the diversity weights for 100 iterations. We fine-tune the loss term balance hyperparameter and determine its optimal value $\gamma = 0.8$, where the weights converge to a stable distribution, while reaching a diversity loss as close to the maximum as possible. Without the entropy loss term ($\gamma = 1.0$) the weights yield the highest VS, but reach both very high and very low values. Large differences in weight values negatively affect the diversity-weighted sampling step of our method during training, as batches can become saturated with the highest-weighted training examples. In contrast, a bigger emphasis on the entropy loss ($\gamma = 0.6$) results in the weights distribution being closer to uniform, but does not maximise diversity. The hyperparameter $\gamma$ provides control over the trade-off between diversity and *typicality*, i. e. the extent to which a generated artefact is a typical training

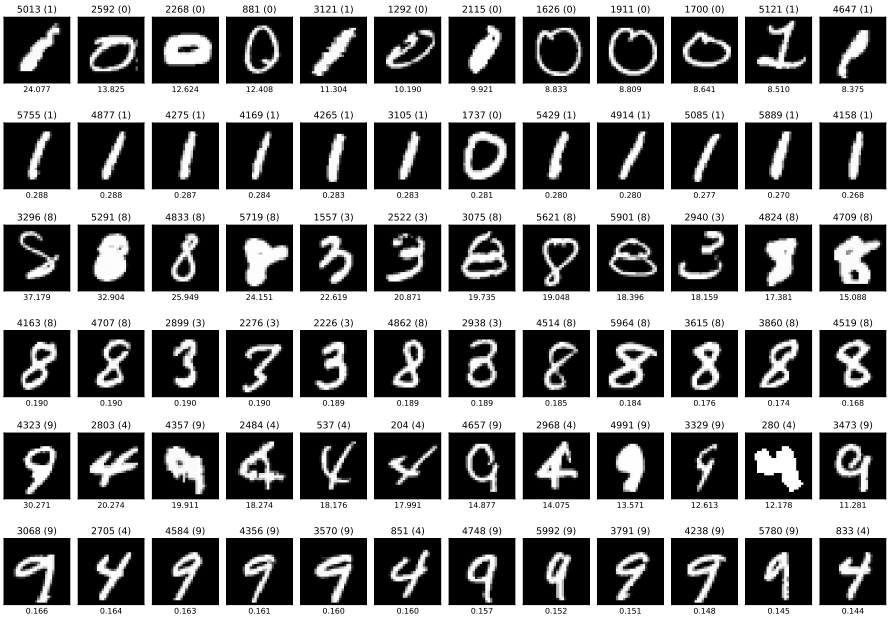



Figure 5.2: Digits ordered by diversity weight (index above with label in brackets, weight below). First two rows: pair 0-1, two middle rows: pair 3-8, last two rows: pair 4-9. Odd rows: twelve highest weighted, even row: twelve lowest weighted.

example (Ritchie, 2007). The VS of the digit datasets when measured with and without diversity weights at different loss term balances are presented in Table 5.1.

The resulting diversity weight for each of the 6,000 samples corresponds to their individual contributions to the overall diversity of the dataset. We give an overview of the highest and lowest weighted data samples in Figure 5.2. Low-weighted samples are prototypical examples of the MNIST dataset: e. g. round zeros and simple straight ones, all of similar line width. High-weighted samples show a much greater diversity: thin and thick lines, imperfect circles as zeros, ones with nose and foot line.

*Training* For each digit dataset, we compare two training schemes: 1) a baseline model with the standard training scheme, and 2) three models trained with our diversity weights (DivW) method and different loss term balances ($\gamma$), where training examples are drawn according to the corresponding diversity weights. The compared loss term balances are $\gamma = 0.6$, $\gamma = 0.8$, and $\gamma = 1.0$. All models have identical architectures (Wasserstein

Table 5.2: Architecture of generator and critic networks. Upsampling convolutional layers (ConvTranspose) have kernel size $4 \times 4$, stride 2, padding 1, dilation 1. Convolutional layers (Conv) have kernel size $5 \times 5$, stride 2, padding 2.

| WGAN-GP Generator | | |
|---|---|---|
| Layer | Output | Activation |
| Input $z$ | 64 | |
| Linear (FC) | 2,048 | ReLU |
| Reshape | $4 \times 4 \times 128$ | |
| ConvTranspose | $8 \times 8 \times 64$ | ReLU |
| Cut | $7 \times 7 \times 64$ | |
| ConvTranspose | $14 \times 14 \times 32$ | ReLU |
| ConvTranspose | $28 \times 28 \times 1$ | Sigmoid |

| WGAN-GP Critic | | |
|---|---|---|
| Layer | Output | Activation |
| Input | $28 \times 28 \times 1$ | |
| Conv | $14 \times 14 \times 32$ | LeakyReLU(0.2) |
| Conv | $7 \times 7 \times 64$ | LeakyReLU(0.2) |
| Conv | $4 \times 4 \times 128$ | LeakyReLU(0.2) |
| Reshape | 2,048 | |
| Linear (FC) | 1 | |

GAN with gradient penalty; Gulrajani et al., 2017) and hyperparameters and are optimised for 6,000 steps (see Tables 5.2 and 5.3 for details).

To allow our method to develop its full potential, we increase the batch size to 6,000 samples, the size of the dataset. Training examples are drawn according to diversity weights *with* replacement, i. e. the same example can be included in a batch more than once. Small batches in turn would be dominated by the highest-weighted examples, causing overfitting and ultimately mode collapse.

*Evaluation* We evaluate individual model performance on six measures, using some common measures for generative models, as well as measures specifically relevant to our method. From each model, we obtain 6,000 random samples, the same size as the digit datasets. Inception Score (IS), Fréchet Inception Distance (FID), and Precision–Recall (PR) with k-NN parameter $k = 3$ quantify sample fidelity and mode coverage with respect to the biased training data distribution. We employ our Weighted Fréchet Inception Distance (wFID) to account for the change in target distribution, induced by our method through diversity-weighted sampling (see Section 5.4.3 for details). We follow the recommendations on anti-aliasing re-scaling in image embedding models (Parmar, Zhang & Zhu, 2022). We

Table 5.3: Training hyperparameters

| Hyperparameter | Value |
|---|---|
| Num steps | 6,000 |
| Num critic steps | 5 |
| Batch size | 6,000 |
| GP weight | 10.0 |
| LR generator | 0.0001 |
| LR critic | 0.0001 |
| Adam $\beta_1$ | 0.5 |
| Adam $\beta_2$ | 0.9 |

use CLIP ViT-L/14 (Radford et al., 2021) as the image embedding model in our feature extraction and measurement pipelines (except for IS), thus evading the ImageNet data biases and unreliable measurements that do not agree with human assessment (Kynkäänniemi et al., 2023). Note that, while trained on a much larger proprietary dataset and better suited as an embedding model, CLIP still has its own biases. Diversity is estimated with the Vendi Score (VS), for which we obtain pairwise similarities between images by calculating the Gram matrix of features. Note that we follow the recommendations by Barratt and Sharma (2018) and calculate IS over the entire generated set of samples, removing the common split into subsets. We also remove the exponential, such that the score becomes interpretable in terms of mutual information. While not all reported scores are directly comparable to other works, our measurements are internally consistent and reliable.

To quantify the effective increase in diversity, we calculate how many random samples from the standard models are required to reach the diversity of a smaller random sample from the DivW models. We measure diversity with the VS. For this, we collect 6,000 random samples, the same size as the digit datasets, from the DivW $\gamma = 0.8$ models and measure their diversity. We then collect increasingly larger sets of random samples from the standard models (6,000, 30,000, 60,000, 300,000 and 600,000) and measure their diversity. These random sampling procedures are repeated with five

different random seeds for each of the five models of the respective training methods (DivW and standard GANs). As a baseline, we measure the diversity of the corresponding digit dataset.

### 5.5.2 RESULTS

An overview of our quantitative results is given in Figure 5.3. For three pairs of digits, we compare our diversity weights (DivW) method with three different loss term balances ($\gamma$) against a standard GAN. The balance of loss terms determines the emphasis on a uniform distribution of weights (lower $\gamma$) over higher diversity (higher $\gamma$). Accordingly, in the diversity weight optimisation, a balance of $\gamma = 1.0$ corresponds to a full emphasis on diversity and no entropy loss, while $\gamma = 0.5$ strikes an equal balance between the two.

Our results agree on almost all measures across all three digit pairs, except on IS which we discuss further below. As expected, the higher the emphasis on the diversity loss, the higher (and better) the VS (Figure 5.3, top left). This comes with a trade-off in sample fidelity and mode coverage, as quantified by PR (Figure 5.3, middle and bottom left) and FID (Figure 5.3, top right). However, when accounting for a weighted training dataset with our Weighted FID measure, the distance of our DivW model distribution to the target distribution is notably lower than or at least on par with the standard model (Figure 5.3, middle right).

Results on IS (Figure 5.3, bottom right) show the difficulty in distinguishing different pairs of digits. For the pairing 0-1, the standard model and the DivW $\gamma = 0.6$ model score notably higher than the other two DivW models ($\gamma = 0.8$ and $\gamma = 1.0$), while their scores are lower for the pairings 3-8 and 4-9. This suggests that, even for the standard model it is difficult to model two similar digits like 3-8 and 4-9.

For the digit pair 4-9, the conventional performance measures (Precision, Recall and FID) exhibit high variances in the standard model. Note the spread of the 95 % confidence interval and the outliers in Figure 5.3.

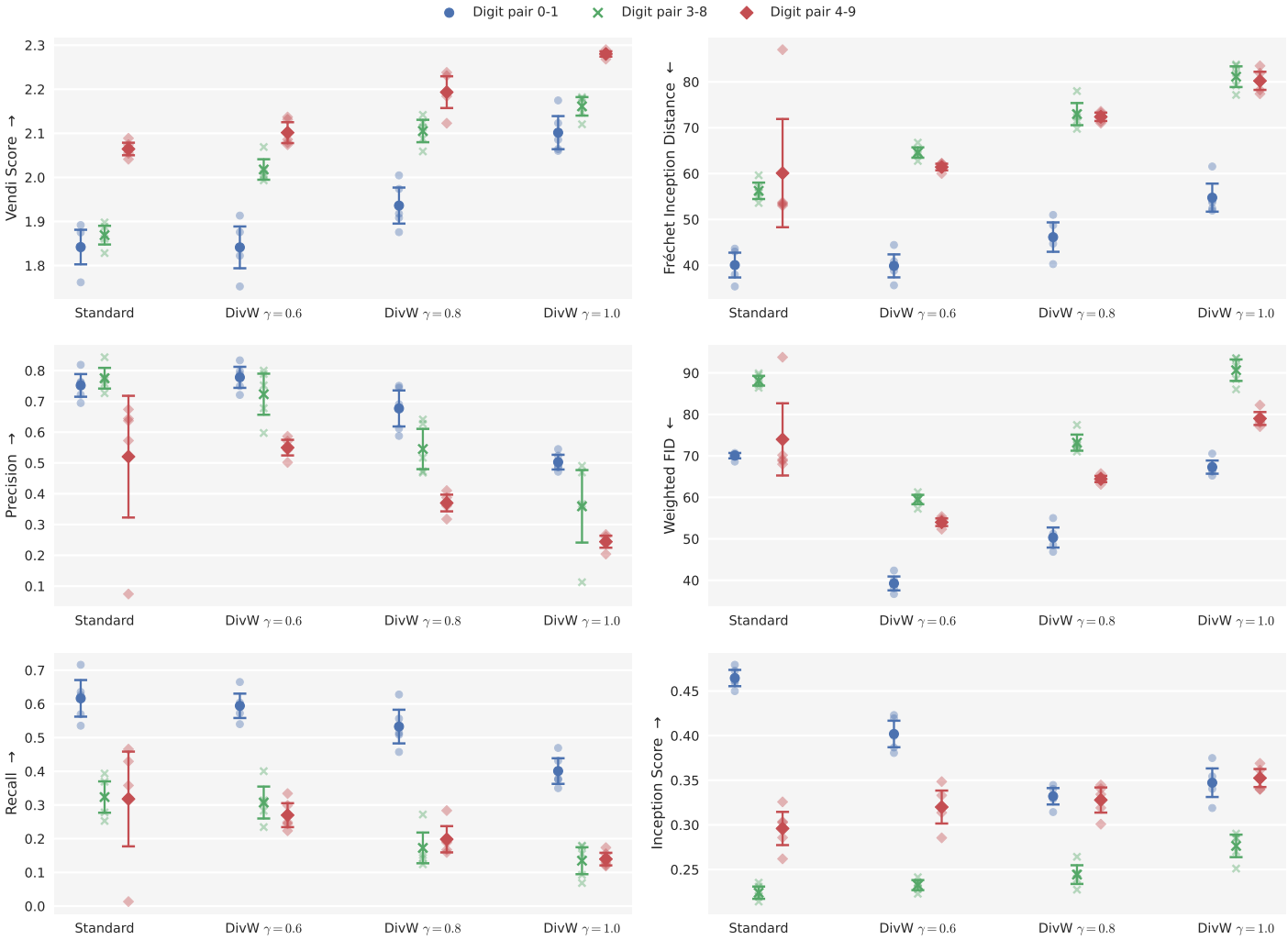



Figure 5.3: Performance comparison of our method (DivW) with different loss term balances ($\gamma$) against a standard GAN, trained on three digit pair datasets (blue circles: 0-1, green crosses: 3-8, red diamonds: 4-9) with six measures: VS, PR and IS (higher is better), as well as standard FID and weighted FID scores (lower is better). Means and 95 % confidence intervals over five random seeds. Individual datapoints show means over five random sampling repetitions. The hyperparameter $\gamma$ provides control over the trade-off between diversity and typicality.

For models trained with our DivW method, the variance is greatly reduced, leading to a more predictable model performance. In particular, when a small loss term balance ($\gamma$) is chosen, the performance is comparable with the standard model, however with lower variance.

For visual inspection and qualitative analysis, we provide random samples in Figure 5.4 for all digit pairs and models.

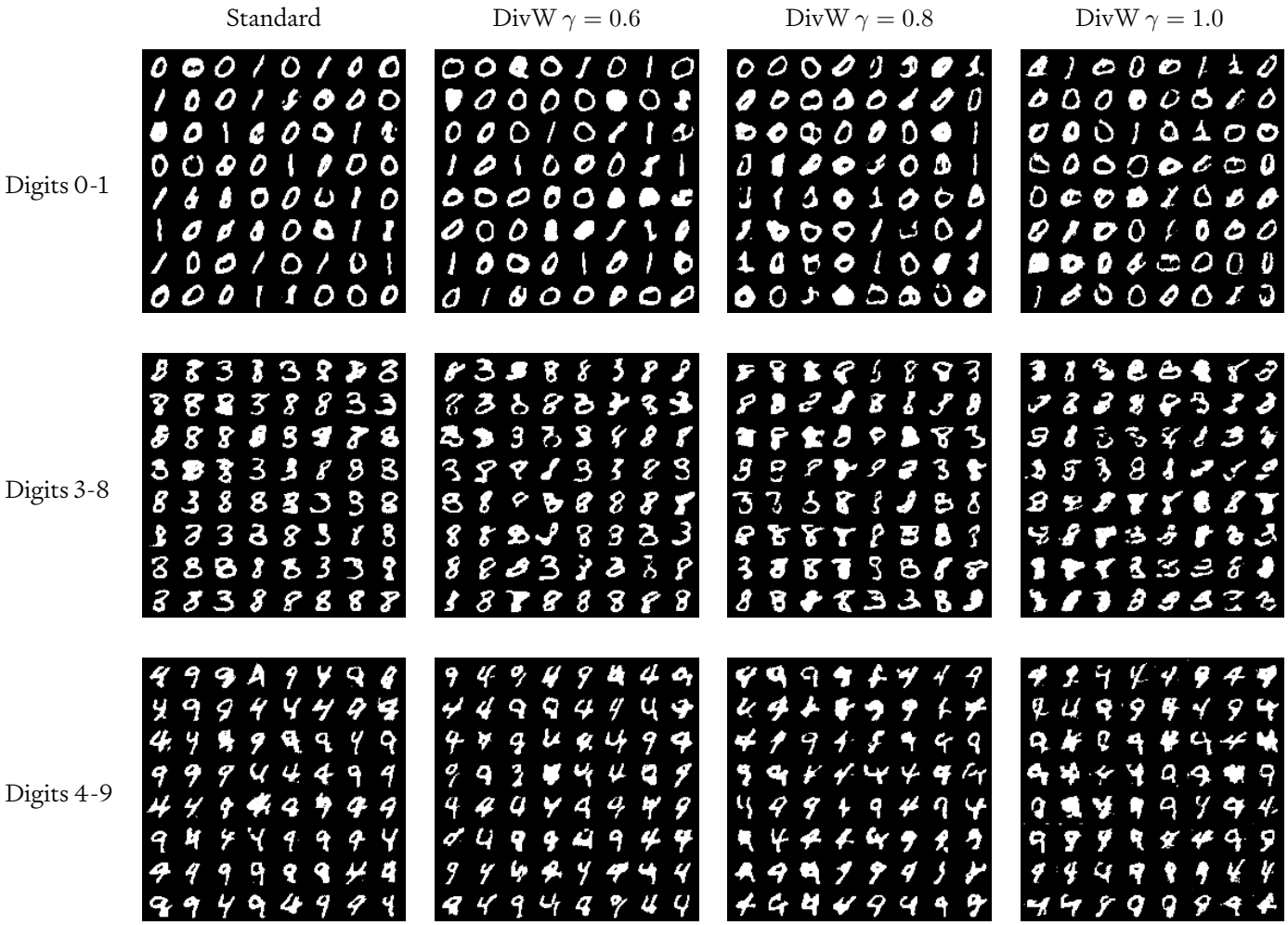



Figure 5.4: Random samples for all digit pairs (top row: 0-1, middle: 3-8, bottom: 4-9) from the standard models (left column) and our DivW models with different loss balances ($\gamma$). The hyperparameter $\gamma$ provides control over the trade-off between diversity and typicality.

To quantify the effective increase in diversity, we compare the output diversity of the models trained with our DivW method to the standard GANs and the digit datasets as the baseline. Results for different digit pairs are visualised in Figure 5.5. The diversity of the digit datasets is calculated over the complete set of 6,000 training examples. The output diversity of the standard and DivW models is calculated on sets of random samples of different sizes (y-axis). The diversity of the digit dataset sets the baseline for the level of diversity present in the training data. Neither the standard GANs nor our DivW method can match the level of dataset diversity. However, the output diversity of DivW models is considerably higher than that of the

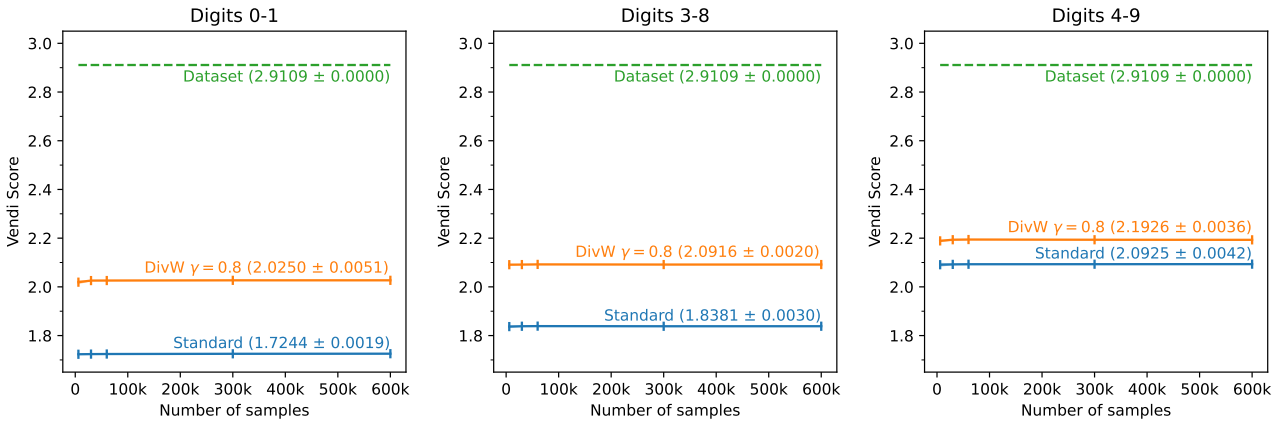



Figure 5.5: Comparison of the output diversity (y-axis) for different sample sizes (x-axis) of DivW models and standard GAN models against the diversity of the training dataset. Means and standard deviations over scores were computed for five random initialisations (dataset and models) and five random samples (models) for each initialisation.

| Digits | 0-1 | 3-8 | 4-9 |
|---|---|---|---|
| DivW ($\gamma = 0.8$) | +17.43 % | +13.79 % | +4.79 % |

Table 5.4: Relative increase in output diversity (VS) of models trained with our DivW method over standard GANs

standard models across all sample sizes. Furthermore, even when sampling a very large amount of samples from the standard models, their output diversity does not reach the diversity of a small set of samples from the DivW models. This suggests that the DivW training scheme enables models to capture modes from the dataset that standard GAN training is unable to include natively.

As the value of the VS is an effective number, it can be interpreted as the number of modes present in a sample. For the standard models, the VS is around 2, corresponding to the two digits in the datasets. Table 5.4 shows the relative increase in output diversity of the DivW models over the standard models as estimated by the VS.

## 5.6 DISCUSSION

In recent years, research communities have become better aware of data biases and their impact on society through the proliferation of data-driven technologies. Likewise, CC researchers have highlighted its potential implications for CC research and the importance of mitigation (Smith, 2017; Loughran, 2022). Real-world datasets are a limited sample of a complex world and should not be considered the 'ground truth', or as representing the 'true' distribution. This practical impossibility further motivates our proposal to shift away from the predominant mode coverage paradigm.

Ongoing debates have not yet resulted in a uniformly accepted way of dealing with data bias in generative machine learning more generally, and CC specifically. One way to address data bias is to gather more or better data. But this is not always possible or practical, since collecting, curating and pre-processing new data is notoriously laborious, costly, or subject to limited access. Another way is to instead adjust the methodology of learning from data, such that a known data bias is mitigated. In this work, we focus on the latter and propose the *diversity-weighted sampling* scheme to address the imbalance of representation between majority and minority features in a dataset.

Diversity weights address the specific bias of *data imbalance*, particularly in unsupervised learning. In contrast to supervised settings, where class labels provide a clear categorisation of training examples, here common features are often shared between various types of examples. This makes it difficult to find an appropriate balance of training examples. Diversity weights give an indication of which type of examples are under-represented from a diversity-maximisation perspective. We draw a connection to issues of DEI as data biases often negatively affect under-represented groups (Bolukbasi et al., 2016; Zhao et al., 2017; Hendricks et al., 2018; Stock & Cisse, 2018).

Combining image generation models with multi-modal embedding models, like CLIP, enables complex text-to-image generative systems which can be doubly affected by data bias through the use of two data-driven mod-

els: the image generator and the image-text embedding. The discussion on embedding models, and other methods that can guide the search for artefacts, is beyond the scope of this chapter and thesis. Our work focuses on the image generators powering these technologies. Yet, a conscious shift to *mode balancing*, in particular for the training of the underlying generative model, could support the mitigation of bias in text-to-image generation models, complementing existing efforts in prompt engineering after training (Colton, 2022).

It is worth noting, that our method also introduces bias, particularly emphasising under-represented features in the dataset. We do this explicitly and for a specific purpose. Other applications might differ in their perspective and objective and deem none or other biases less or more important. Since a dataset cannot maintain its status of 'ground truth', the responsibility of reviewing and potentially mitigating data biases falls onto researchers and practitioners.

Diversity–typicality trade-off

A possible explanation for the diversity–typicality trade-off is given by an information-theoretical view on compression. The architecture of image-generating artificial neural networks (ANNs) are typically set up as decoding networks, to learn a mapping from a latent representation space to a complex feature space (e. g. colour images). The prior distribution of the latent space is typically a standard normal or uniform distribution. Where possible, the latent space is often of lower-dimensionality than the feature space. For example, VAEs feature an encoding network and make use of an explicit information bottleneck to learn a compressed latent encoding of a given data distribution. In contrast, the input and output space of flow-based and diffusion probabilistic models (DPMs) have the same dimensionality, due to constraints of the modelling approach. Latent diffusion models (LDMs) extend DPMs by an initial perceptual image encoding step.

In this context, a latent space is effectively a lower-dimensional compression of data examples, that retains semantically meaningful relationships between examples. However, an encoding network only has limited resources and thus is a lossy compressor. Depending on the constraints imposed by the information bottleneck (e. g. degrees of freedom, disen-

tanglement, regularisation), the encoding network can assign more or less latent information to high-level semantic image features (e. g. in the case of human faces: skin tones, length and colour of hair). Since likelihood under the model is assigned in accordance to the frequency of types of examples in a dataset, it is to be expected that small image details and differences are ignored by the model, as they only appear in individual examples. With an increasing compression rate between feature space and latent space, e. g. due to a big difference in dimensionality, it becomes more difficult for a model to include all variation in a dataset. As a result, a model will focus on the most typical examples, while disregarding outliers. This phenomenon is likely to occur in *all models* to some degree, as the world is too complex to be efficiently represented with limited resources. Hence, a trade-off between diversity and typicality is inevitable.

Fidelity–coverage trade-off

A similar trade-off between sample fidelity and mode coverage appears in the generative process of several other methods. A recent diffusion model (Dhariwal & Nichol, 2021) introduces a hyper-parameter to control this trade-off. The parameter scales the influence of gradients from a separate classification model at every diffusion step. The classifier guidance improves sample fidelity at the cost of mode coverage. The truncation trick (Brock, Donahue & Simonyan, 2019) for GANs consists of sampling the latent vector, input to the generator network, from a normal distribution of limited value range, effectively tightening all values around the mean. As outlier values are no longer permitted, generated samples gain in fidelity, but lose in mode coverage. In most likelihood-based models, the temperature parameter (Ackley, Hinton & Sejnowski, 1985) likewise allows for emphasis on the modes of the training data distribution. In rejection sampling for GANs (Azadi et al., 2019) and likelihood-based models (Razavi, van den Oord & Vinyals, 2019) a classifier provides confidence scores of generated samples which are used to reject samples that do not meet a given probability threshold. Rejection sampling for VAEs (Bauer & Mnih, 2019) has been extended by a learned acceptance function to improve the model's prior distribution. Classifier scores can be seen as implicit likelihood estimates.

Chapter 6

# SIMILARITY ESTIMATION FOR THE EVALUATION OF DIVERSITY

In artificial intelligence, to automate processes as much as possible, human evaluation is often substituted by approximate computational measures, e. g. to quantify the similarity of two artefacts. When a measure substitutes human evaluation of similarity, it is paramount to employ a similarity measure that accurately captures relevant criteria and relations between artefacts. While there are many options for similarity measures, it is unclear how they correlate to human perception of similarity. There exists no empirical evidence that can support this decision.

We alleviate this gap in two human participant studies, in which we collect human similarity judgements and interpretations of the relevant visual criteria of the judgements. We focus on video game applications and procedural content generation (PCG), particularly tile-based video game levels. In a quantitative study, we collect the similarity judgements from a large group of participants ($N = 456$) and compare them against 7 similarity metrics with a total of 12 configurations to determine which existing metrics best approximate the human similarity perception of tile-based video game levels. In addition, we perform a qualitative study in which four focus groups with relevant experience ($N = 4 \times 2$) provide their interpretation of the dimensions underlying human similarity judgements. Our findings inform the selection of existing similarity metrics and highlight requirements for designing new metrics benefiting video game development and research.

The methodology presented in this chapter addresses RQ 4 and exemplifies *how computational measures can be aligned with the human perception they are supposed to substitute*. In the wider context of this thesis, this is important for a human-centric approach to increasing the output diversity of generative

models. The performance of a generative machine learning model and the qualities of a dataset are determined with specialised measures to compare different approaches and track their progress. Many of these measures rely on similarity estimation to quantify conventional performance indicators like sample fidelity and mode coverage, as well as more complex concepts such as novelty and diversity. In the Vendi Score (VS) family of diversity measures for machine learning (Friedman & Dieng, 2023; Pasarkar & Dieng, 2024), selecting a similarity function is an important choice for two reasons. First, to align automated evaluation and decision-making with human criteria. And second, to ensure a measure captures domain-relevant criteria. For example, images are typically compared on image semantics rather than absolute pixel values. Similarly, the functional qualities of molecules might be more important for their similarity than their structure.

The work in this chapter was presented at CHI 2024:
Berns, S., Volz, V., Tokarchuk, L., Snodgrass, S., & Guckelsberger, C. (2024). Not All the Same: Understanding and Informing Similarity Estimation in Tile-Based Video Games. *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*.

The collected study data and the implementation of our similarity metric test suite are available online:
https://github.com/sebastianberns/similarity-estimation-chi24.

CONTENTS

## 6.1 INTRODUCTION

For video games to be enjoyable, game designers must anticipate how their players will perceive, and consequently experience and react to, elements in the game. This however proves challenging when considering elements that only unfold dynamically at runtime, such as procedurally generated content or the behaviour of non-player characters (NPCs). To constrain such processes and meet players' expectations, designers can endow them with computational metrics that approximate player perception, experience, and behaviour (Yannakakis & Togelius, 2011; Canossa & Smith, 2015; Guckelsberger et al., 2017).

Here, we focus on an important family of such metrics to assess players' perception of visual similarity. Metrics of similarity are an integral part of many game AI applications such as procedural content generation (PCG), which we use as a motivating case. For example, procedural content generation via machine learning (PCGML) (Summerville et al., 2018) approaches rely on similarity metrics to generate artefacts such as game levels that resemble existing samples, or that are sufficiently distinct from previously played levels. In contrast to this runtime use-case, similarity metrics have also been used in design-time tools, e. g. to determine and modify the expressive range of content that a generator with a specific configuration can produce (Smith & Whitehead, 2010; Summerville, 2018; Cook et al., 2021).

There exist many similarity metrics to choose from, ranging from general-purpose ones to data-driven approaches to expert measures custom-tailored to a scenario. Typically, game designers and researchers select a similarity metric based on conventions, personal preferences, basic assumptions, or computational properties.

However, in both online and design-time PCG practice, it is unclear if the generated content is actually perceived as similar by players, i. e. how well the metric works as a surrogate of players' perception (Volz et al., 2020; Withington & Tokarchuk, 2023). If the selected metric is misaligned, the consequences can be detrimental to how a game is experienced, as exemplified

by the *Thousand Bowls of Oatmeal Problem*, a term coined by Kate Compton to describe the 'common antipattern of generating a set of artefacts which are technically distinct to the computer, but perceived by humans as uniform' (Compton & Mateas, 2015). Given its strong resonance with game development reality, it has quickly become one of the best-known idioms in the PCG community (Rabii & Cook, 2023). A classic example of this phenomenon is the lack of visual variety in the 18 quintillion possible planets of *No Man's Sky*, which are mathematically, but not perceptually, unique (Maiberg, 2016). In this context, perceptual uniqueness has been promoted as the 'real metric' required. Its characterisation as a 'darn tough' one, highlights the importance of the development and identification of visual similarity metrics that approximate human perception as a core research challenge in games.

We hold that there exists no empirical data in the context of games to adequately support designers and researchers in selecting the most appropriate similarity metric. At the same time, there exists a wide range of metrics to choose from, including general-purpose metrics with no psychometric claims, custom-made metrics from game AI and PCG, and models from computer vision (CV) research. While some hold that writing metrics that ought to approximate human perception in video games 'is a difficult skill that requires a deep understanding of the application domain' (Rabii & Cook, 2023), it is presently unclear whether such domain-specific metrics truly perform best. Image embedding models are of particular interest here. While CV has long been active in developing surrogate models for the human similarity judgement for very specific domains (e. g. Wills et al., 2009; Piovarči et al., 2018; Lagunas et al., 2019; Shi et al., 2021), we recently saw a surge in the publication of more generic models (e. g. Radford et al., 2021; S. Fu et al., 2023). This goes against the above claim for the necessity of domain knowledge. Moreover, one may assume that such CV models cannot approximate the human similarity judgement well on the synthetic and highly stylised levels of especially non-realistic video games, since they are trained on, and optimised primarily for natural images. But we do not know this for a fact. Moreover, metric development at present is based on designer intuition, but we are in the dark about which visual features of game levels

really determine players' similarity judgement. This is of high significance for the game industry and research, as the development of custom metrics is time-intensive and hence costly. More generally, choosing a sub-optimal metric could result in bad guidance at design time or unsatisfying player experiences when used in-game.

With the work in this chapter, we contribute to a better understanding of the human similarity judgement and its alignment with existing metrics for a specific sensory modality in a well-constrained but very popular non-realistic game genre. We address the more general RQ 4 and, by focusing on people's perception of visual similarity of tile-based video game levels, further address two more specific research questions.

RQ 4.1: Which existing metrics approximate the human similarity perception of grid-based video game levels best?

RQ 4.2: What are the dimensions of this space that govern players' similarity perception?

The second question serves as a direct response to the first, in that its answer can serve as a stepping stone to inform the development of better domain-specific metrics. Moreover, the gained insights can teach designers how players perceive their assets, e. g. to inform a more intuitive creative process at design time even if not relying on computational support tools. All in all, the methodology of this chapter, from the collection of human participant judgements to the labelling of, answers our initial RQ 4 and exemplifies *how computational measures can be aligned with the human perception they are supposed to substitute*.

We investigate these questions through two empirical studies. We collect data on the human similarity perception in a 2 × 2 factorial study, covering two very different titles (Candy Crush Saga; Legend of Zelda) in two visual representations (level screenshots; abstract colour patterns). In a mixed design, participants compared the similarity of level triplets for subsets of each factor combination. In each triplet comparison task, they are presented with a reference stimulus and choose the most similar stimulus from two options. Choices are forced, and participants cannot skip a task. Using

a variant of multi-dimensional scaling (MDS), we build domain-specific perceptual spaces, encoding similarity-relevant attributes for this specific scenario. We compare a selection of PCG, general purpose and computer vision metrics against these perceptual spaces, thus contributing to RQ 4.1. This is complemented by our second study, in which we asked focus groups *with relevant experience* to gather interpretations of the dimensions underlying these perceptual spaces, supporting RQ 4.2 and thus fostering designer insights and the future development of better metrics.

Our contributions are threefold:

1. A quantitative study comparing similarity judgements from participants ($N = 456$) against a total of 12 configurations of 7 existing metrics.

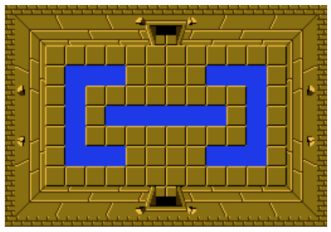
2. A qualitative interpretation study, in which four focus groups with experience relevant to video game design, development, and research ($N = 4 \times 2$) provide their interpretations of the dimensions underlying the human similarity judgements in this domain.

3. A public dataset of human similarity judgements in tile-based video game levels and implementation of the comparison test suite.[1]

We moreover critically reflect on the requirements of each group of metrics and provide recommendations for scenarios when not the best but a runner-up metric might be preferred.

Similar to Rabii and Cook (2023), we thus set out to put intuitions and internalised knowledge of game researchers and practitioners to the test in the hope of strengthening applications and inspiring new research. Our findings from (1) serve the game development and research communities by informing recommendations for which existing metrics should be preferred in different scenarios, resting on a strong empirical basis. Moreover, our findings from (2) inform the future development of more human-aligned similarity metrics in the video games domain. The publicly available code and data in (3) facilitate the evaluation of additional existing and newly developed similarity metrics, thus enabling game developers and researchers

1 https://github.com/sebastianberns/similarity-estimation-chi24



Consider the patterns contained in the images. Which of the two images below is **most similar** to the one above?

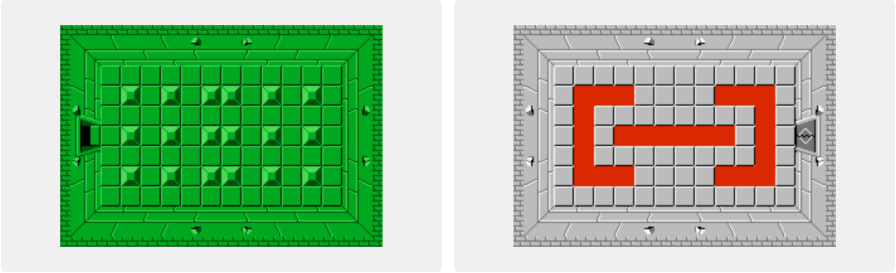


Figure 6.1: Triplet questions: two alternative forced choice (2AFC). Participants are presented with a reference stimulus (top) and have to choose between two options (bottom). Questions can not be skipped.

to easily build on our work. Focusing on levels of tile-based video games, we relate to a particularly big and popular game genre, and arguably the most common type of procedurally generated content. While using PCG as a prominent application domain to motivate this research, an evaluation of these metrics in concrete PCG algorithms is out of scope. Instead, we focus on the comparison of existing metrics for approximating the human similarity perception in a way that is agnostic w.r.t. the specific game AI application.

## 6.2 METHODOLOGY

### 6.2.1 DATA COLLECTION AND ANALYSIS

Our choices of data collection and analysis methods are well supported by other work with similar methodologies. Human similarity ratings are typically collected by presenting participants with triplet matching tasks (*two alternative forced choice*, 2AFC, Figure 6.1) (Wills et al., 2009; Piovarči

et al., 2018; Lagunas et al., 2019; Shi et al., 2021; S. Fu et al., 2023), which is the most robust judgement type (Demiralp, Bernstein & Heer, 2014). Collected triplet judgements are then converted into a perceptual space through multi-dimensional scaling (MDS) or related ordination techniques (see Section 6.2.3 for details). The resulting representation is often used to label and thus identify the dimensions underlying the human similarity judgement for the stimuli in question. Crucially, this is where most existing analyses stop; we adopt this common methodology for our study, but take it further by comparing the perceptual spaces derived from human judgement against those produced by computational metrics.

### 6.2.2 SIMILARITY METRICS FOR VIDEO GAMES

To calculate video game level similarity, game developers and researchers leverage methods from three different groups of measures and distances: 1) artificial neural network-based image embeddings trained on datasets of natural images for computer vision (CV), 2) domain-agnostic, general-purpose distance metrics (General), and 3) manually-designed measures based on expert knowledge, from the PCG literature (PCG). In Table 6.1, we list and describe all embeddings, distances, and measures used as metrics for comparison in this study. Our focus in this selection lies on measures specifically used in video games-related research and the game industry. In the following, we provide further detail about our choice of metrics.

We define a *measure* as a method to quantify the qualities of a video game level and a *metric* as the comparison of such qualities between two levels.[2] To build a working similarity metric, embeddings, distances, and measures need to be transformed and compared. We outline here how this applies to the aforementioned groups and our selection.

In computer vision (CV), it is common to use the embedding spaces of artificial neural networks to compare the perceptual similarity of images (R.

2 Note that we do not follow the stricter mathematical definition of a "metric" here, but instead use the term more colloquially as a way of more easily differentiating the methods that quantify qualities from those that compare them.

Table 6.1: Selection of image embeddings, metrics and measures (with optional configurations) compared in this work. Note that the image embeddings and measures require additional transformations to be used as similarity metrics (Section 6.2.2).

| Name | Group | Input | Output | Description |
|---|---|---|---|---|
| CLIP (Radford et al., 2021) | CV | Image | Vector | Image embedding trained on a huge dataset of image-text pairs scraped from the internet. |
| DreamSim (S. Fu et al., 2023) | CV | Image | Vector | Image embedding fine-tuned on human similarity judgements (two alternative forced choice). |
| Normalised Compression Distance (M. Li et al., 2004) | General | Tiles | Scalar | Using a compression algorithm (gzip), compares the joint compression length of two levels to their individual compression lengths. |
| Hamming Distance | General | Tiles | Scalar | Fraction of tiles that exactly match across two levels. |
| Tile Frequencies (Summerville et al., 2017) | PCG | Tiles | Distribution | Relative frequencies of tile types appearing in a level. |
| Tile Patterns (Lucas & Volz, 2019) | PCG | Tiles | Distribution | Relative frequencies of tile patterns appearing in a level. Configurations: size of patterns ($2 \times 2, 3 \times 3, 4 \times 4$). |
| Symmetry (Volz et al., 2020) | PCG | Tiles | Scalar | Fraction of tiles that match when mirroring half of a level across a corresponding axis (e. g. vertical symmetry: left and right halves compared across the centre). Configurations: axis of symmetry (Horizontal, Vertical, Diagonal Forward, Diagonal Backward). |

Zhang et al., 2018). Most recent embedding models (e. g. CLIP) have been specifically designed for the evaluation of two inputs via cosine similarity (Radford et al., 2021). We chose two state-of-the-art image embedding models: CLIP (ViT-L/14@336px) for its ubiquitous use and DreamSim (ensemble) for its specific alignment with human perception. Both take as input one square colour image (either a level screenshot or the corresponding colour pattern; see Section 6.3.1) and yield its corresponding embedding vector. To evaluate the similarity of any pair of images, we calculate the cosine similarity between their embedding vectors.

While little PCG research focuses on similarity estimation specifically, many works propose or use some measure to evaluate generative systems and their output. For example, in expressive range analysis (Smith & Whitehead, 2010) or to drive quality diversity search in video game asset produc-

tion (Fontaine, Liu et al., 2021). Researchers draw from expert knowledge to design specialised measures that capture relevant qualities. In contrast to CV embedding models, PCG measures always take as input a tile-based representation of a level (independent of experimental condition), where individual tile types are encoded as ASCII characters. Tile Frequencies is a popular baseline measure to characterise tile-based levels (Summerville et al., 2017). While it disregards the location of tiles and thus does not fully capture the composition of a level, the discrepancy of different tile types appearing in two levels might be enough to approximate the overall visual similarity between the two levels. This simple idea has been extended to larger Tile Patterns (Lucas & Volz, 2019). While Tile Frequencies only consider individual tiles ($1 \times 1$ patterns), Tile Patterns can be configured to calculate the occurrences of any $N \times M$ pattern in a level. Both Tile Frequencies and Tile Patterns take as input the tile-based representation of one level and yield the probability distribution over the tiles or patterns that appear in the level. We calculate the similarity between two levels by first calculating the Jensen-Shannon distance between the two tile or pattern distributions and then converting their distance into similarity by subtracting it from 1. We further included symmetry measures because research on patterns in Candy Crush Saga has shown that symmetric generated levels are considered more similar to original game levels by human expert judges (Volz et al., 2020). While symmetry by itself is probably not sufficient to fully describe level similarity, we hypothesise that it might be an important factor in the human perception of tile-based video game levels. Symmetry measures take as input one level in a tile-based representation and yield a scalar output that quantifies the level's symmetry on a given axis (horizontal, vertical, or either forward or backward diagonal). Two levels are compared in terms of their similarity by calculating the absolute difference between their symmetry scores.

As general and domain-agnostic metrics, we selected Hamming and Normalised Compression Distance (NCD). While these have been applied to PCG (Rodriguez Torrado et al., 2020; Edwards, Jiang & Togelius, 2021), they have not been specifically developed for video game applications. In-

stead, they stem from information-theoretic approaches to measuring distances between strings of text. Hamming Distance provides a simple baseline, is easily interpretable and finds many applications in video game research in its more general form as *edit distance* (Alvarez et al., 2018; Todd et al., 2023). NCD has been used as a metric for the structural similarity of video game levels, as it encodes both tile frequencies and positions (Shaker et al., 2012; Mariño, Reis & Lelis, 2015). Both general metrics take as input two levels in the tile-based representation and yield the distance between them, which is converted into their similarity by subtracting the distance from 1.

As motivated earlier , an intriguing question is how well state-of-the-art computer vision metrics, which were not developed specifically for use in games, can compete with more conventional or custom-made metrics already adopted in games. Crucially though, stimuli in studies on image similarity more generally, e. g. photographs (Rogowitz et al., 1998), are arguably far removed from the imagery that players experience in the tile-based video games under consideration. We include DreamSim (S. Fu et al., 2023) here both as a recent example to frame and compare our study setup against, as well as a metric in our study (Section 6.3). For the development of DreamSim, S. Fu et al. (2023) have curated a dataset of human judgements over pairs of synthetic images, following the same 2AFC triplet judgement task method described above and employed in our work. Crucially, their image triplets were iteratively selected for maximum participant agreement, effectively optimising for an easily solvable binary decision task. In contrast, we take into account participant disagreement and thus gather richer relational information between stimuli. Their work focuses on synthesised natural images and thus compares conventional CV metrics and state-of-the-art learned, i. e. data-driven, embeddings. We instead focus on metrics relevant to video game development and research but overlap with their work in comparing CLIP (Radford et al., 2021) as a popular image embedding. They finally use their dataset to fine-tune an ensemble model for measuring image similarity, which we leave for future work.

### 6.2.3 PERCEPTUAL EMBEDDING SPACES

Our work aims to compare various similarity metrics to the judgements collected from our study participants. To facilitate this, we apply the conventional methodology of constructing a *perceptual space* from the triplet judgements that embeds all stimuli in a Euclidean space where distances correspond to the perceived relations between triplets (Demiralp, Bernstein & Heer, 2014; Piovarči et al., 2016; 2018). We thus understand the distance in the Euclidean space to be the inverse of perceived similarity: the more similar two stimuli are, the closer they will be positioned to each other in the embedding.

More formally, in an exemplary triplet judgement task, let $A$ be the reference stimulus, and $B$ and $C$ be the two options participants can choose from (Figure 6.1). Suppose a participant decides that the reference $A$ is more similar to option $B$ than it is to option $C$. We can describe this relation as $d(A, B) < d(A, C)$, where $d$ is a distance metric in Euclidean space. Let us call this a *paired comparison* of the given triplet $A, B, C$. The embedding space is built by finding the vectors corresponding to all stimuli $\vec{a}, \vec{b}, \vec{c}$, such that $\|\vec{a} - \vec{b}\| < \|\vec{a} - \vec{c}\|$. Naturally, this relationship should hold for all collected triplet judgements, thus creating a set of constraints on the vectors. The construction of the perceptual embedding is conventionally formulated as a constrained optimisation problem.

A common method to obtain such an embedding is multi-dimensional scaling (MDS). A loss function (called *strain*) quantifies how well the embedding satisfies all constraints. Several versions of MDS exist, most notably metric and non-metric algorithms. However, most require a target distance matrix in which the pairwise similarities between stimuli are expressed as numerical distances. This is difficult to obtain from our study data, in particular, because not all participants judged every triplet. Generalised non-metric multi-dimensional scaling (GNMDS) (Agarwal et al., 2007) instead reformulate the loss function to primarily depend on information from the paired comparisons. Additional slack variables account for unsatis-

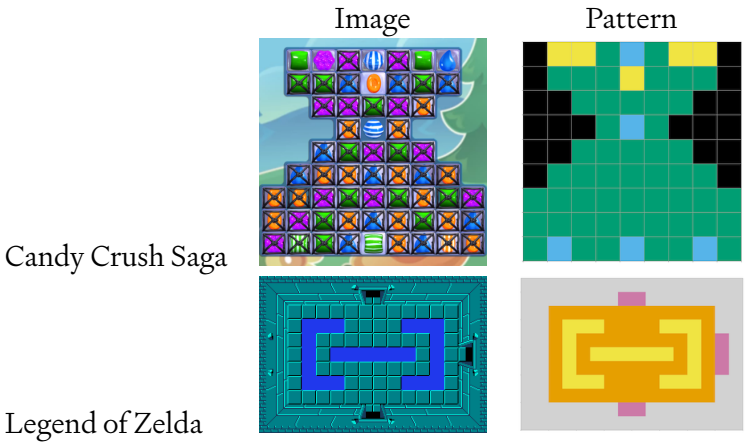



Figure 6.2

fied constraints. The optimisation objective aims to minimise the amount of slack. Yet, when there is high disagreement in the data between judgements from individual participants, it becomes difficult to satisfy all constraints at once. This results in large amounts of remaining slack.

In our work, we employ t-distributed stochastic triplet embedding (t-STE) (van der Maaten & Weinberger, 2012), which responds better to the naturally occurring noise in the judgement data by not trying to satisfy constraints that contradict the consensus. t-STE can thus deal best with two important characteristics of our collected judgement data: 1) missing data due to participants only judging a subset of triplets, and 2) high disagreement between individual participants due to the difficulty of the triplet judgement task.

## 6.3 STUDY 1: HUMAN VS. COMPUTATIONAL SIMILARITY EVALUATION

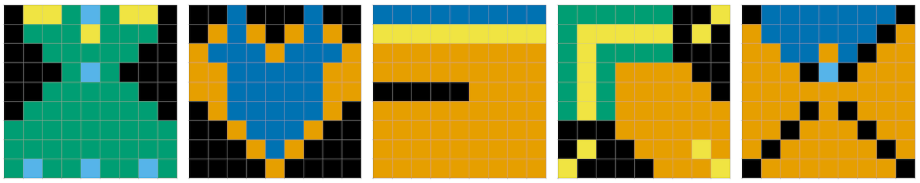
To compare the human evaluation of similarity with computational metrics, we collect data on people's evaluation of similarity in tile-based video games. To this end, we employ a full 2×2 factorial design with the first factor defining the video game *Title* (*ccs*: Candy Crush Saga; *loz*: Legend of Zelda) and the second the visual *Representation* of levels (*img*: level screenshots; *pat*: an abstract colour tile pattern of the level sprite layout). This yields a

total of four experimental conditions (Figure 6.2): *ccs-img*, *ccs-pat*, *loz-img*, *loz-pat*. We choose two *Representations* to cover different scenarios relevant to the application of similarity metrics in video games and PCG. The *img* representation provides direct insight into how people assess the similarity between levels as they appear in the given *Titles*. Through this, we aim to inform the selection of similarity metrics for application in these and other closely related video games. With the *pat* representation, we focus on more abstract pattern representations of level layouts as they are used in the level design process and by many PCG algorithms. Our goal is to provide practical recommendations for game designers, developers and researchers for the application of similarity metrics at design time and in conjunction with PCG and PCGML approaches. We are interested in answering RQ 4.1 individually for both of these scenarios (*Which existing metrics approximate the human similarity perception of grid-based video game levels best?*). Note that there is an important difference in the design of the similarity metrics. CV-based metrics (CLIP and DreamSim) take image input and can be applied to any image. In *img* conditions, they will be given level screenshots, whereas, in *pat* conditions, they will receive the colour patterns. In contrast, all other metrics receive levels in their tile-based representation and are given the same information in all conditions.

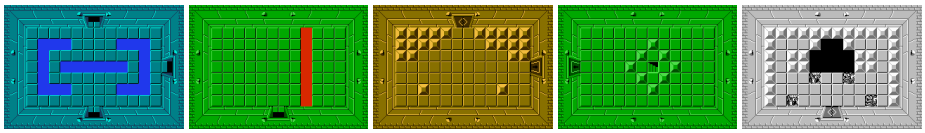
With the stimuli in each condition, we prepared a collection of triplet comparison tasks as two alternative forced choice (2AFC) questions. Given a reference stimulus, participants are asked to make a forced choice between two stimuli, selecting the option most similar to the reference. This design was shown to be the most robust data collection method and has been recommended for assessing perceptual similarities (*triplet ranking with matching*) (Demiralp, Bernstein & Heer, 2014). To prevent participant fatigue but still assess a high number of stimuli, we employ a mixed design where each participant judges a subset of triplets from each condition. The study was approved by the Queen Mary Ethics of Research Committee.[3]

3 Reference number: QMERC20.565.DSEECS23.030

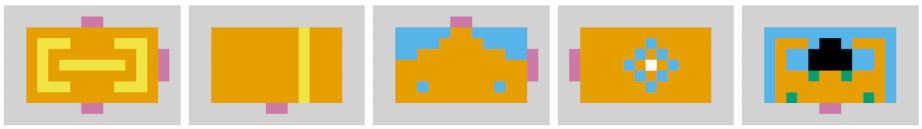
(a) Condition: *ccs-img*



(b) Condition: *ccs-pat*



(c) Condition: *loz-img*



(d) Condition: *loz-pat*

Figure 6.3: Five random example stimuli for each condition. The first two rows show levels from Candy Crush Saga and the last two levels from the Legend of Zelda, in the image and pattern representation, respectively. Each stimulus is randomly drawn from the respective subset identified through our three-stage selection procedure.

### 6.3.1 MATERIALS

As stimuli, we first select a subset of level images from both video games (*Title*). Video game levels in the *img* representation include some decorative elements, e. g. different colour sprites for the same game objects in *loz* and certain game objects, like candies, being represented by different sprites in *ccs*. We hypothesise that the *img* representation, essentially content shown in-game, evokes gameplay associations in the participants and obfuscates some similarity-relevant visual patterns.

To test this hypothesis, we leverage an abstract colour pattern representation (*pat*) for each *Title* that relies on existing mappings from level object to colour tile (Summerville et al., 2016 for *loz* and Volz et al., 2020 for *ccs*).

The purpose of the *pat* representation is to remove potentially distracting gameplay associations and emphasise the similarity-relevant characteristics of levels, e. g. shapes and patterns. These types of colour tile patterns are commonly used in PCG in research (Summerville et al., 2016; 2018; Sarkar & Cooper, 2022; Bhaumik et al., 2023), as well as in practice (Grinblat & Bucklew, 2010; Stålberg, Meredith & Kvale, 2018). In the following, we describe how the conversion from *img* to *pat* representation is performed and how it differs between the two *Titles*. However, to represent level objects, we apply the same colour-blindness-safe colour palette (Wong, 2011) to converted level representations from both *Titles*. For Legend of Zelda (*loz*), the colour tile mapping defined in the VGLC (Summerville et al., 2016) is straightforward, as it simply maps level elements with different functionality to distinct colour tiles (e. g. walls are different from floors are different from enemies are different from doors, and thus assigned different colours). In this abstraction, simplifications are limited to subsuming all enemies into a single colour tile and ignoring the different colour palettes of the various dungeon rooms. The colour tile mapping we use for Candy Crush Saga (*ccs*) is informed by Volz et al. (2020) and was devised in collaboration with the game's creator, King. Instead of a direct one-to-one mapping from each level object to a colour tile, this representation subsumes several level objects with similar in-game behaviour. For example, objects that look visually distinct (e. g. frosting and chocolate objects) but perform similar game functions (*blockers* impede moves, making gameplay more difficult) are mapped to the same colour tile. This is done for several functional level objects, such as *blockers*, *candies*, *power pieces*, and *locks*. Using the above *img* to *pat* mappings, we represent every *Title* in each *Representation*.

As datasets, we obtain *loz* levels from an open-source corpus of video game levels (Summerville et al., 2016) and scrape *ccs* level screenshots from a fan wiki[4]. Since our datasets (*ccs*: 2,792 levels, *loz*: 225 levels) were too large for a triplet comparison study, we selected a subset of stimuli informed by the expected amount of participants, a minimum of five comparisons per triplet, and a maximum amount of 100 comparisons per participant. To

4 https://candycrush.fandom.com

select a subset representative of the overall variety in levels, we employed a three-stage selection pipeline. We first obtained image embeddings from an artificial neural network (CLIP ViT-14/L@336px Radford et al., 2021). While using a metric assessed in the same study introduces a bias, the bias is explicit and can be accounted for. We discuss this and alternatives we considered in the limitations Section 6.5.1. We then reduced the dimensionality of the embeddings from 768 to 2 dimensions with t-SNE (van der Maaten & Hinton, 2008), to make the subsequent sampling step feasible. In our implementation, t-SNE uses cosine similarity, which is the most appropriate to calculate distances between CLIP embeddings. The origin of biases is thus limited to the choice of embedding model. Finally, we used conditioned Latin Hypercube Sampling (cLHS) (Minasny & McBratney, 2006) to find a subset for which items are maximally distant from each other in the low-dimensional embedding space. This is to ensure that 1) the samples cover a large part of the space of possible levels and 2) that we do not inadvertently draw conclusions from a non-representative subset of levels. To mitigate the influence of different tile colours in Legend of Zelda, we select levels based on their greyscale versions. We selected 17 stimuli for each of the four experimental conditions, yielding $\binom{17}{1} \times \binom{16}{2} = 2040$ triplets per condition, and 8160 triplet comparisons overall. Figure 6.3 shows a random selection of five levels from all subsets, each corresponding to one condition.

Participants were asked optional demographic questions about their self-described gender and age, and their experience with tile-based video games. The surveys were implemented in Qualtrics. Given a list of stimuli, we compute all triplet combinations and generate individual surveys for all conditions for upload to Qualtrics.

### 6.3.2 PARTICIPANTS

We recruited 460 participants from Prolific to complete a 15-minute survey paid at the equivalent of an hourly rate of £10. Funding was provided by modl.ai. We excluded four participants who did not complete the full survey

Table 6.2: Self-reported experience with tile-based video games of participants in study 1 (blue) and study 2 (red). Participants selected one option in each row, and percentages in each row add up to 100 %.

| | I do not know this type of game | I have heard of this type of game | I have played this type of game | I regularly play this type of game |
|---|---|---|---|---|
| Tile-matching games (like Candy Crush or Bejeweled) | 1.80% — | 14.7% 25% | 59.1% 75% | 24.4% — |
| Pacman or Ms Pacman | 2.90% — | 16.5% — | 73.6% 100% | 7% — |
| Retro dungeon crawlers (like Legend of Zelda) | 29% — | 42.0% 37.5% | 23.5% 37.5% | 5.50% 25% |
| Sokoban | 75.8% 62.5% | 15.6% 12.5% | 7.70% 25% | 0.9% — |
| Bomberman, Dyna Blaster, or similar | 48.1% 12.5% | 20.7% — | 29.9% 87.5% | 1.3% — |

and proceeded with the data from the remaining 456 participants. Out of these, 53.51 % reported their gender as female, 43.64 % as male, 1.75 % identified as non-binary or third-gendered, none chose to self-describe, 0.44 % preferred not to respond, 0.44 % left the question unanswered, and 0.22 % abandoned the survey before seeing the question. The median reported age is 28. Our sample is thus considerably more representative w.r.t. identified gender than common in studies related to video games. We summarise their self-reported experience with tile-based video games in Table 6.2.

### 6.3.3 PROCEDURE

We informed our final study procedure based on a pilot, involving seven stimuli in each condition. The goal of this pilot was to test the survey setup and identify average response times, suitability of validation questions, and baseline disagreement ratios on individual triplets. It was completed by 22 trusted participants from the authors' respective industry and academic institutions.

Our study follows the conventional methodology for collecting human similarity ratings with *two alternative forced choice* (2AFC) questions, one of the oldest methods of psychophysics (Fechner, 1860). We interchangeably

refer to this as triplet comparisons. Given a reference stimulus, participants are asked to make a forced choice between two stimuli, selecting the option most similar to the reference. For our study, out of the 8160 total triplets (Section 6.3.1), every participant was assigned a random subset of 25 from each of the four experimental conditions. In addition to these 100 triplet comparisons, participants were asked to judge three additional triplets as validation questions in each condition. The order between and within conditions was randomised for each participant, and colour patterns were shown before level images to not prime participants' perceptions. Participants provided informed consent at the start of the survey and answered optional questions on demographics and game experience after judging all triplets.

### 6.3.4 DATA ANALYSIS

To understand how the computational metrics correlate with our data on the human perception of similarity, we perform two complementary quantitative data analyses. First, we quantify how well the computational metrics can approximate the similarity matrices derived from our participant data. For this, we construct a perceptual space for each condition which embeds the stimuli in a low-dimensional Euclidean space. Second, we conduct pairwise comparisons between the judgements of individual human participants and the different computational metrics in an inter-rater agreement analysis. In addition, we provide a qualitative analysis of the features underlying the human similarity judgements in our second study (Section 6.4). All analyses are performed separately for each experimental condition.

#### 6.3.4.1 PERCEPTUAL EMBEDDING OF TILE-BASED LEVEL SIMILARITY

To determine the overall relationships between stimuli in terms of similarity, aggregated over all human responses, we construct a perceptual space from the collected triplet judgements, i. e. an embedding of stimuli in Euclidean space (here also called *perceptual embedding* or *embedding space*). Participants

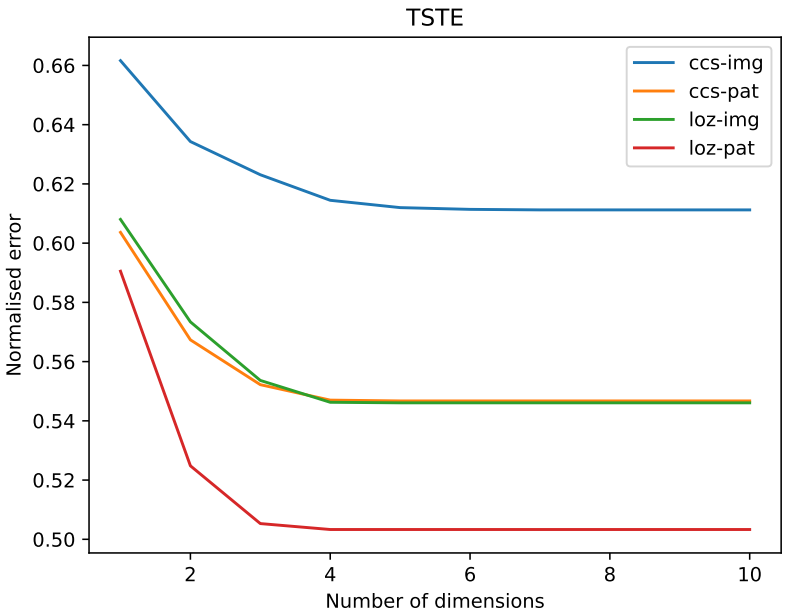



Figure 6.4: Elbow plots for t-STE goodness of fit in all conditions. We choose 4 as the number of dimensions (horizontal axis) for the embeddings based on the evaluation of overall normalised errors (vertical axis).

were asked for their subjective perception of similarity. Choices were forced, and participants did not have the option to skip a judgement task. It is natural that the triplet data is noisy and reflects some disagreement. Yet, this provides important information about the similarity-relations of stimuli and introduces constraints that need to be taken into account. For example, if many participants agree that reference stimulus $A$ is more similar to stimulus option $B$ than the other option $C$, in the embedding space $A$ needs to be positioned closer to $B$ than $C$. A perceptual embedding converts each individual piece of relationship information into an aggregated positional distance within the embedding while satisfying all constraints as best as possible. As noted in Section 2.6, this inclusive approach also distinguishes our work from related work.

We chose the embedding algorithm t-distributed stochastic triplet embedding (t-STE) (van der Maaten & Weinberger, 2012) as it provides several advantages over conventional multi-dimensional scaling (MDS) methods, in particular, the handling of missing data and noisy data (for background see Section 6.2.3). The former is necessary since not all parti-

cipants judge all triplets, and the latter as our data shows a lot of disagreement between participants. Elbow plots (Figure 6.4) indicate that four dimensions can adequately encode the most relevant attributes across experimental conditions while providing a close-to-optimal fit for the data.

To quantify the suitability of the embedding for subsequent comparisons with computational metrics, we analyse the robustness of the embedding to random initialisation over 10 runs with different random seeds. The goodness of fit to the raw data, number of required iterations, and number of constraints are almost identical across all runs. While the absolute positions of triplets in the embedding depend on the initialisation of the embedding and can differ significantly between random seeds, the variance of pairwise similarities between the embedded stimuli is much lower across all conditions (*ccs-img*: 0.0349, *ccs-pat*: 0.0464, *loz-img*: 0.0389, *loz-pat*: 0.0419; variance over 10 runs with random initialisation), indicating overall robustness of the resulting perceptual embeddings. For each condition, we select the embedding with the best fit to the data from these 10 candidates. The placements of the stimuli in the embedding dimensions are visualised in Figures 6.11 to 6.14.

#### 6.3.4.2 COMPARISON OF SIMILARITY MATRICES

To quantify the capabilities of the computational metrics to approximate the human similarity judgements, we calculate the error between similarity matrices derived from either source. The similarity matrix for human judgements is based on the previously described perceptual embeddings (Section 6.3.4.1). We first compute the pairwise Euclidean distances between all stimuli in the embedding, then normalise them by the maximum distance, and finally convert normalised distances into similarities by subtracting them from 1. The similarity matrix for a computational metric is constructed from the pairwise similarity between stimuli computed by a given metric as outlined and motivated in Section 6.2.2. Two similarity matrices are compared by calculating the mean squared error. Results are summarised in Section 6.3.5 and visualised in Figure 6.5.

This comparative analysis allows us to quantify a metric's prediction error of the similarity-relation between two stimuli by comparing it to the ground-truth human perception. However, this can only be done by way of constructing a perceptual embedding space from the collected judgement data, which itself only approximates the judgement data. We supplement this first analysis with the following inter-rater agreement analysis, as it allows for a more direct comparison to the judgement data given by our participants, without requiring an intermediate approximation.

#### 6.3.4.3 AGREEMENT BETWEEN PARTICIPANTS AND METRICS

We perform an inter-rater agreement analysis between human participants and computational metrics. Cohen's kappa ($\kappa$) is calculated for pairs of one participant and one metric as the two raters. For this, we first find the triplets judged by a given participant and then determine the judgements of the metric in question on the same triplet comparison tasks. This allows us to perform a direct inter-rater agreement analysis. This process is repeated for each combination of participant and metric in each condition. We remind the reader that in each triplet comparison, a participant is presented with a reference stimulus A and chooses the most similar stimulus from two options B and C. In standard inter-rater agreement terminology, we thus deal with two raters each judging 25 items on a two-category nominal scale (stimulus option B or C). We use Cohen's kappa over agreement percentage, as it takes into account the possibility of chance agreements, which is particularly important when dealing with only two categories. As not every participant has judged all triplets, the statistics only reflect agreement on each participant's random subset of 25 triplets from each condition. For each condition, we thus collect as many data points (kappas) as there are participants who completed this section of the survey.

There exist some potential limitations in the interpretation of Cohen's kappa statistic on its own as the range of agreement and disagreement between human participants and computational measures lacks a frame of reference. Due to the mixed design of our study, however, it is not pos-

sible to calculate a baseline agreement between human raters. To tackle the large number of triplets as efficiently as possible, each participant judges only a subset of triplets from each condition. Most participants thus did not rate the same triplets as any other participant. For each condition, the overlap of rated triplets between two participants is zero in over 89 % of the cases, one in over 9 % and two in over 1 %. Less than 1 % of participants rated three or more of the same triplets as any other participant. Since the vast majority of participants did not rate the same triplets, there is not enough information to calculate a meaningful statistic of the average agreement between human participants for each condition. In any case, such a single statistic would merely indicate the offset of Cohen's kappa from its neutral value of zero. A comparable, and potentially better adjustment of kappa's range of values is provided by the maximum value of kappa, as discussed below.

We thus perform additional analyses of the inter-rater agreements to support the interpretation of Cohen's kappa ($\kappa$). Different scales have been proposed to interpret the magnitude of kappa (e. g. poor, slight, fair, moderate, substantial, and almost perfect; for different intervals of kappa). Yet, choosing any such standard for the evaluation of the strength of agreement is inevitably arbitrary. Moreover, a potential scale would have to be adjusted to the maximum value kappa could attain for a given pair of ratings. While kappa is theoretically upper-bounded by 1, in practice its maximum value is often much lower, as kappa is highly sensitive to differences in allocation and quantities. Considering a $2 \times 2$ contingency table, maximum agreement is only possible if the marginal distributions are balanced. We assist the interpretation of kappa by calculating the maximum value of kappa across our pairwise comparisons (Sim & Wright, 2005), and visualise the difference between individual kappa and their respective maximum values $\kappa_{\max} - \kappa$ (*unachieved agreement*, lower is better) as raincloud plots in Figure 6.8. This provides a more realistic scale of comparison across metrics. We further report two easily interpretable coefficients, appropriate for evaluation of accuracy in prediction tasks, quantity disagreement and allocation disagreement (Pontius & Millones, 2011), visualised in Figures 6.9 and 6.10.

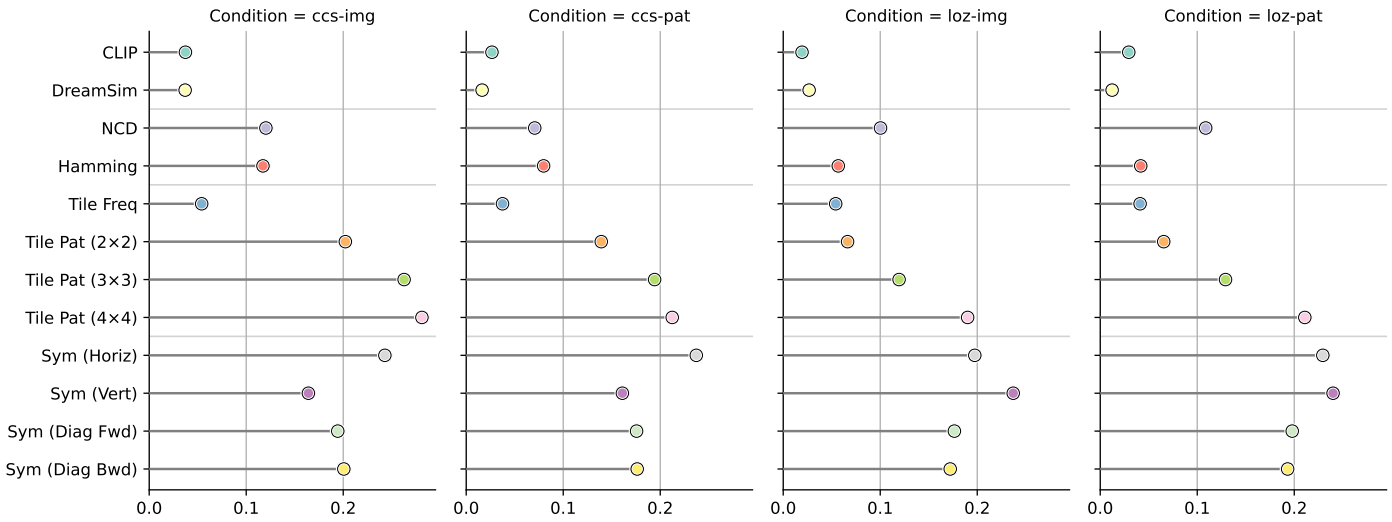



Figure 6.5: Mean squared errors (lower is better; horizontal axes) when comparing the pairwise similarity matrices of different candidate metrics (vertical axis) to those derived from the perceptual embeddings of the four experimental conditions (subplots).

This collection of inter-rater agreement statistics complements the initial comparison of similarity matrices, allowing for a direct comparison of a metric's binary prediction to a given participant's judgements. However, a binary choice between two stimuli options only gives a limited account of the complex similarity-relations between stimuli. In contrast, the initial comparison of similarity matrices can accommodate more fine-grained relations, expressed as distances in a Euclidean space. For readability, we present summarising box plots of the results for Cohen's kappa here and complete raincloud plots at the end of the chapter.

### 6.3.5 RESULTS

From the 456 participants, we collect a total of 11,400 judgements per condition, resulting in an average of 5.6 judgements per triplet comparison. We next present our results separately for each of the analysis steps outlined in Section 6.3.4. We give a summary of the results here, alongside visualisations.

In our main analysis, we compared the pairwise similarity matrices of different candidate metrics to those derived from the perceptual embeddings. We report results as mean squared error, where a lower score indicates a better approximation of the perceptual embeddings by a computational metric (Figure 6.5). Overall, the two CV metrics, CLIP and DreamSim, have the lowest errors across all experimental conditions. Looking at individual conditions, DreamSim exhibits the best approximation performance when using pattern-based representations (*ccs-pat, loz-pat*). While CLIP has slightly lower error for image-based Legend of Zelda levels (*loz-img*), both CV-based metrics are tied on image-based Candy Crush Saga levels (*ccs-img*). Tile frequencies are the overall third-best-performing approximate metric across all experimental conditions. In fourth place, the general-purpose metrics, Normalised Compression Distance (NCD) and Hamming Distance are tied in terms of overall error on Candy Crush Saga levels (*ccs-img*, *ccs-pat*). However, for Legend of Zelda levels in both representations (*loz-img*, *loz-pat*), Hamming Distance performs almost equally as well as Tile Frequencies. Tile Patterns in various configurations ($2 \times 2$, $3 \times 3$ and $4 \times 4$) are not good approximations for our collected human judgements. We observe that a larger pattern size leads to a higher error. We provide an explanation in the discussion (Section 6.5). Similarly, Symmetry metrics in all configurations (horizontal, vertical, as well as diagonal forward and backward) yield comparatively high overall errors.

In a supporting inter-rater agreement analysis, we calculated the agreement between every pair of individual human participants and computational metric. A summarising box plot shows the median agreements and the interquartile ranges, where a higher score indicates higher agreement (Figure 6.6; full raincloud plot available in Section 6.3.6). The agreements between participants and metrics, according to Cohen's kappa, are overall low to moderate. We discuss this further in Section 6.5.1. Yet, the results are nuanced enough to allow for interpretation and conclusions. As all metrics exhibit roughly similar interquartile ranges, we will focus our description on their median agreements with participant judgements. Out of all metrics, DreamSim shows the overall highest agreement. This is followed by

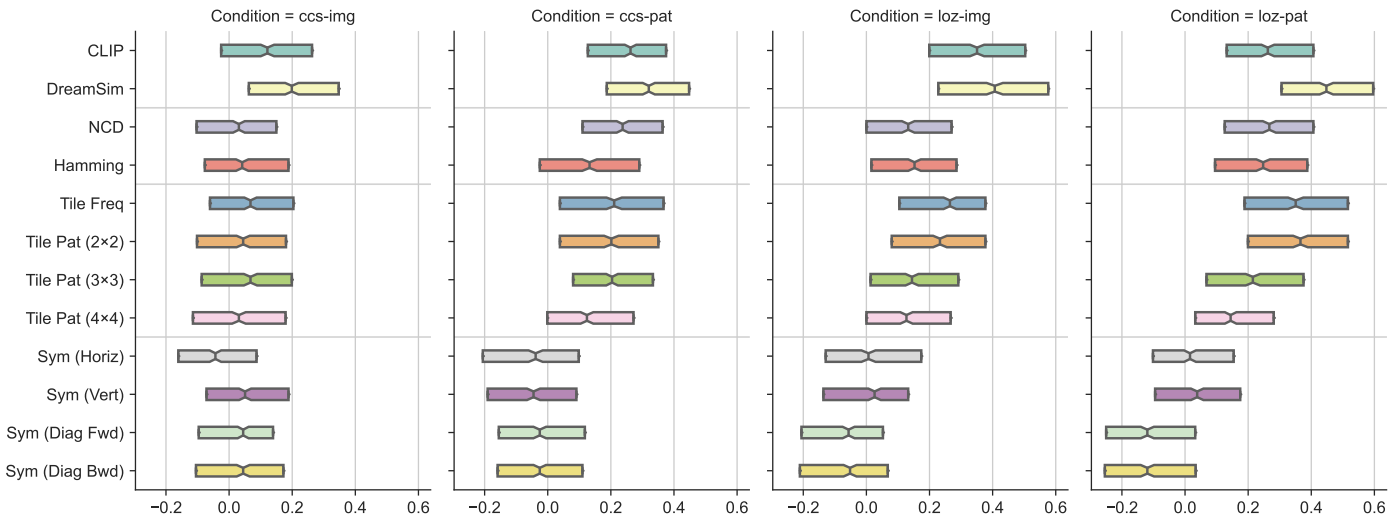



Figure 6.6: Cohen's kappa (higher is better): inter-rater agreement between human participants and computational metrics over all experimental conditions (subplots). Summaries here show box plots with median values and the interquartile ranges. Full raincloud plots can be found in Section 6.3.6.

CLIP which has the second-highest agreement in most conditions. The only exception is the pattern-based representation of Legend of Zelda levels (*loz-pat*), where Tile Patterns ($2 \times 2$) and Tile Frequencies beat CLIP and share a close second-highest agreement. Tile Frequencies has the third-highest agreement for images of Legend of Zelda levels (*loz-img*), again closely followed by Tile Patterns ($2 \times 2$). For the pattern-based representation of Candy Crush Saga levels (*ccs-pat*), Normalised Compression Distance has the third-highest agreement, but only by a small margin when compared to Tile Frequencies as well as Tile Patterns ($2 \times 2$) and ($3 \times 3$). Third-highest agreement in Candy Crush Saga images (*ccs-img*) is shared by Tile Frequencies and Tile Patterns ($3 \times 3$), though very closely followed by several other metrics.

We performed statistical significance testing (Vornhagen, Tyack & Mekler, 2020) on the agreement between metrics and participant judgements (Cohen's kappa). First, we test our basic assumption: (H1) there are significant differences in the performance of metrics in individual conditions. For this, we perform a one-way ANOVA separately for each condition. We further seek to evaluate two other hypotheses: (H2) DreamSim, from the CV group,

performs better than Tile Frequencies, the next-best metric from a different group, i. e. PCG expert metrics; (H3) metrics have a higher agreement with participant judgements of the pattern-based representation of levels than with judgements of level images. H2 is tested with a paired student's t-test of the two related samples within individual conditions: participant agreement with DreamSim and with Tile Frequencies. H3 is tested with a separate independent student's t-test of each metric between individual conditions. As H3 entails multiple comparisons, we correct p-values with the Benjamini-Hochberg procedure (Benjamini & Hochberg, 1995). We perform these tests on Cohen's kappa and not on the approximation errors, as the tests require a minimum number of samples.

One-way ANOVAs, separately for each condition, confirm that there are significant differences (all $p < 0.01$) in the agreement between participant and metrics (H1). Paired student's t-tests in each condition confirm that DreamSim has a significantly higher agreement (all $p < 0.01$) than Tile Frequencies (H2). Independent student's t-tests, followed by p-value correction, confirm that the best metrics from each group, DreamSim (CV), Hamming Distance (General), and Tile Frequencies (PCG), have higher agreement (all $p < 0.01$) for pattern-based representations than images (H3). However, this does not hold for all metrics.

### 6.3.6 RAINCLOUD PLOTS

On the following pages, we present the full raincloud plots of Cohen's kappa from our inter-rater agreement analysis (Figure 6.7). To support our main analysis, we report three additional statistics of inter-rater agreement between human participants and computational metrics: (a) unachieved agreement (Figure 6.8), (b) quantity disagreement (Figure 6.9), and (c) allocation disagreement (Figure 6.10). In all three statistics, lower scores indicate higher agreement. While results are difficult to interpret across all statistics and experimental conditions, there are a few observable patterns, supporting the main analysis. DreamSim has the overall lowest median

scores and interquartile ranges, followed by CLIP and Tile Frequencies. Other metrics occasionally perform better than some of the three but not across all statistics and conditions.

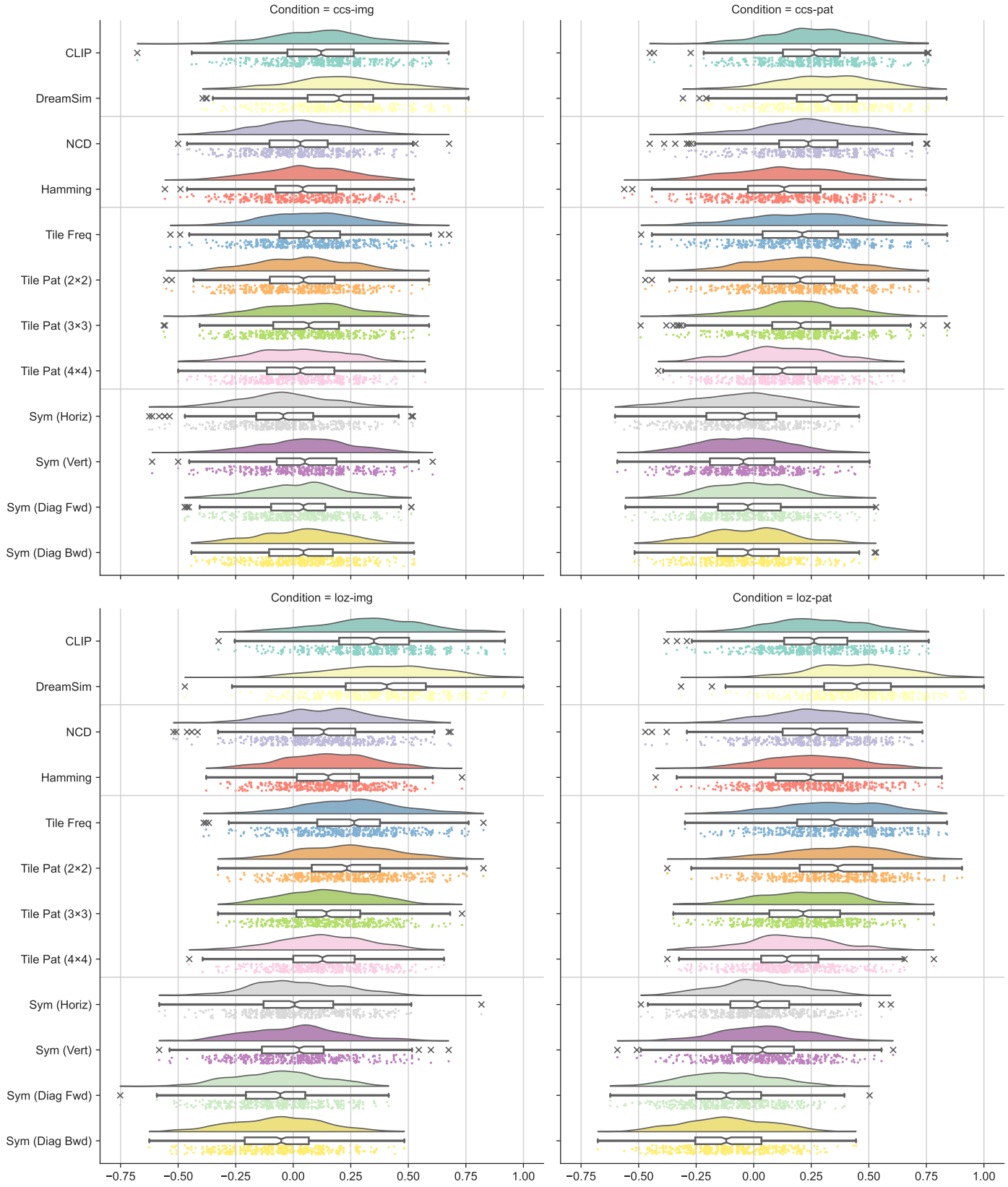



Figure 6.7: Cohen's kappa (higher is better): inter-rater agreement between human participants and computational metrics over all experimental conditions (subplots). Each data point indicates Cohen's kappa comparing the similarity judgements of a single participant against those of a given metric on the same subset of triplets. Each raincloud plot features individual data points as dots, the estimated kernel density over the data as a curve above the data points, and a box plot with the sample minimum, maximum and median, as well as the first and third quartiles and outliers.

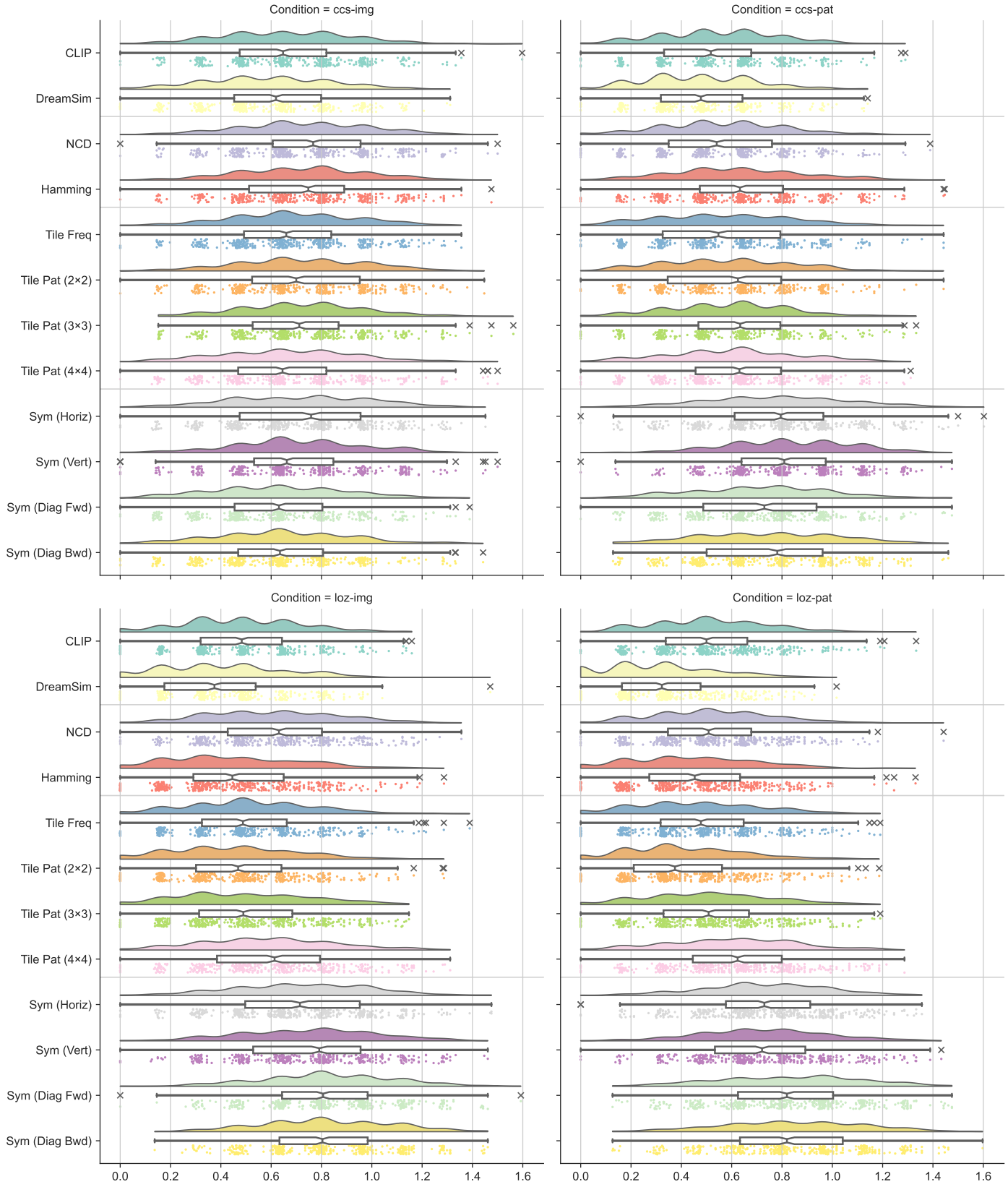



Figure 6.8: Unachieved agreement (lower is better): difference of the maximum value and Cohen's kappa of the inter-rater agreement between human participants and computational metrics over all experimental conditions (subplots). Each data point indicates Cohen's kappa subtracted from $\kappa_{\max}$, when comparing the similarity judgements of a single participant against those of a given metric on the same subset of triplets. Each raincloud plot features individual data points as dots, the estimated kernel density over the data as a curve above the data points, and a box plot with the sample minimum, maximum and median, as well as the first and third quartiles and outliers.

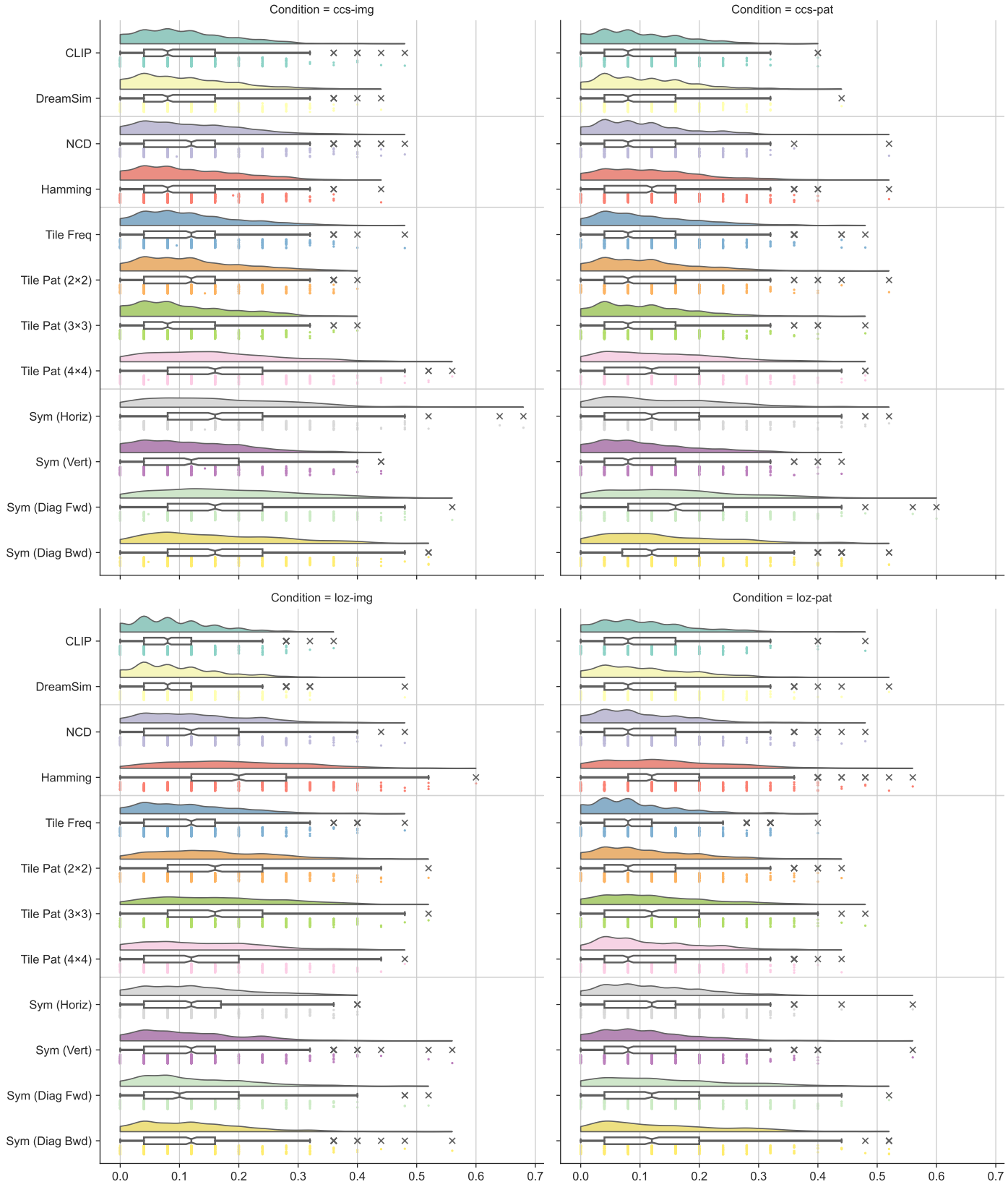



Figure 6.9: Quantity disagreement (lower is better) between human participants and computational metrics over all experimental conditions (subplots). Each data point indicates disagreement between a single participant and a given metric on the same subset of triplets. Each raincloud plot features individual data points as dots, the estimated kernel density over the data as a curve above the data points, and a box plot with the sample minimum, maximum and median, as well as the first and third quartiles and outliers.

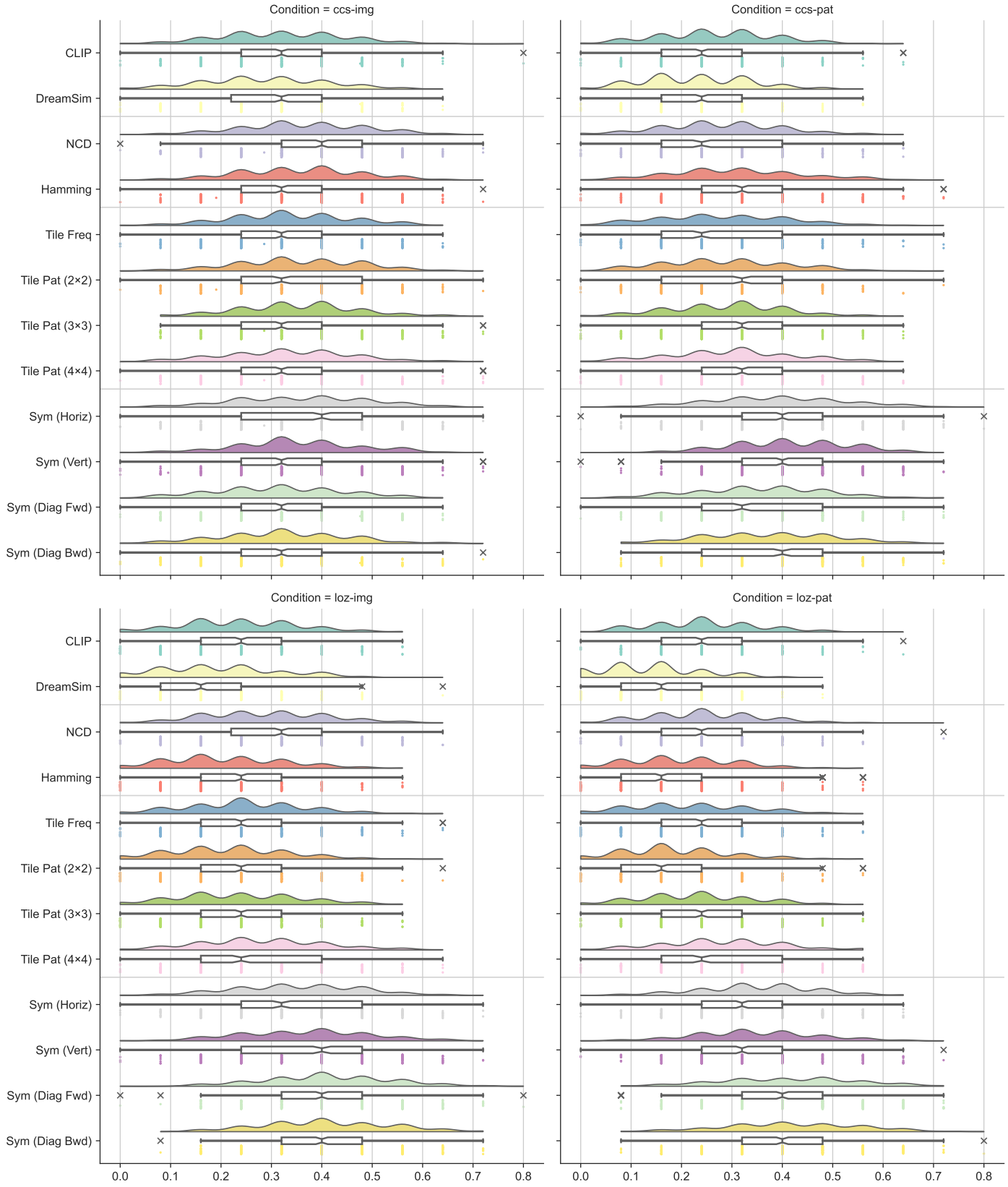



Figure 6.10: Allocation disagreement (lower is better) between human participants and computational metrics over all experimental conditions (subplots). Each data point indicates disagreement between a single participant and a given metric on the same subset of triplets. Each raincloud plot features individual data points as dots, the estimated kernel density over the data as a curve above the data points, and a box plot with the sample minimum, maximum and median, as well as the first and third quartiles and outliers.

## 6.4 STUDY 2: INTERPRETATION OF SIMILARITY DIMENSIONS

Our first study has shown that the approximation of human similarity assessments through custom-tailored metrics leaves space for improvements. In our second study, we identified the dimensions underlying the human similarity assessment to better understand this phenomenon and inform the future development of fast and compact similarity metrics tailored to this domain. We thus adopt a similar methodology as in other work on the human perception of similarity (Section 6.2.1) and re-use the perceptual spaces from the first study identified through t-STE on the triplet judgements (Section 6.3.4.1), to ask participants in focus groups to interpret their dimensions. To prevent participant fatigue, we employed a mixed design where each condition was assigned to one focus group, tasked to provide interpretations for all four dimensions of the associated perceptual space. We obtained approval from the Queen Mary Ethics of Research Committee.[5]

### 6.4.1 MATERIALS

We prepared a guide for all participants with a tutorial to demonstrate the exercise. It shows a horizontal axis with several circles arranged by increasing size from left to right. The suggested label for this example is "pattern size" or "from small to big". For each of the four focus groups, we prepare an A2 printout composing all four embedding dimensions of the corresponding condition, to be handed to each participant within. We leave space under each axis for people to note their ideas. The dimensions are not provided on screen to improve readability and avoid distractions. We used the same demographics and experience questionnaire as in the first study (Section 6.3.1) but as a printout.

5 Reference number: QMERC20.565.DSEECS23.055

### 6.4.2 PARTICIPANTS

Our focus groups were composed of a total of eight participants (two per experimental condition) with backgrounds covering HCI and psychology, game AI research, as well as game design and development. These participants were recruited from the IGGI PhD programme, a doctoral training centre spanning multiple universities and focusing on video game research with a strong industry orientation. The study was open for everyone over 18 with normal or corrected to normal vision, which was not assessed. Participants were incentivised with a £15 gift voucher.

Out of the eight participants, seven reported their gender as male, and one as female. The median reported age is 28. Participants in our second study have overall higher experience with the relevant tile-based video games than our general demographic in the first (Table 6.2).

### 6.4.3 PROCEDURE

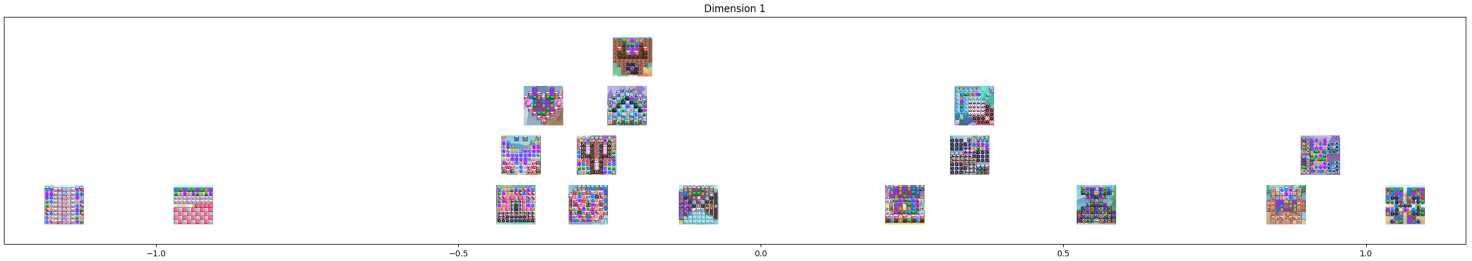
The focus groups were conducted as part of a workshop run at Queen Mary University of London and lasted about 45 minutes each. We ran a total of four individual focus group sessions. All sessions followed the same procedure, described below, but focused on interpreting the dimensions from different conditions.

At the beginning of each session, participants were informed about the goals of the study through the participant information sheet. They were particularly reminded that multiple interpretations for each dimension are possible, that there are no right or wrong answers, and that their subjective opinion counts. After giving informed consent, they were familiarised with the task through the tutorial sheet and offered help with any remaining questions. They were then handed the sheets with the dimensions to label, one for each participant.

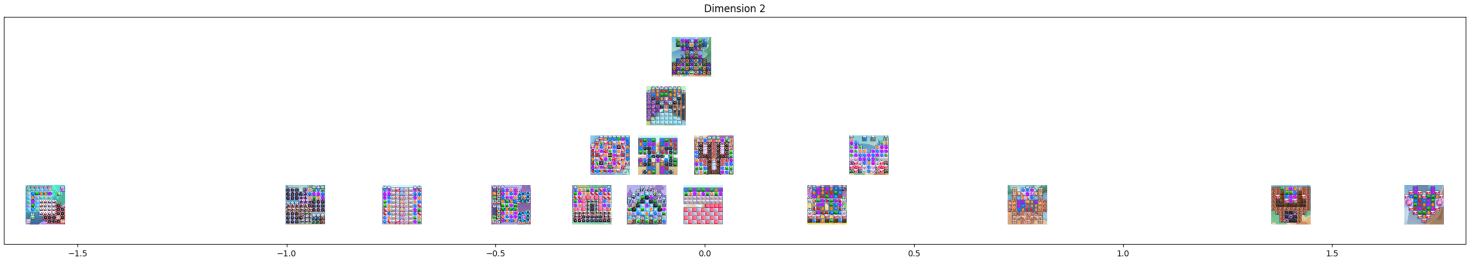
Each session was split into four parts, corresponding to the dimensions on the paper provided to the participants. The experimenter initiated each

part by asking the participants to write down their interpretations of the respective dimension silently by themselves. After 5 minutes, they were asked to discuss their proposals with the other members to identify the best interpretation, which they were instructed to write down and highlight. After at most five minutes, the next part was initiated. We decided to interleave the silent individual interpretation task to prevent forgetting about the interpretations and to inspire and inform their upcoming interpretations.

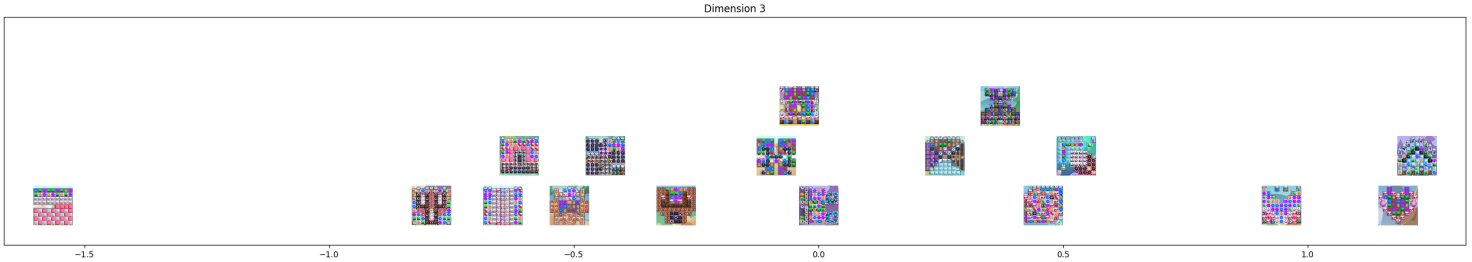
In the debriefing, participants were finally thanked and asked to fill in the demographics and expertise questionnaires. They were then invited to ask any questions, and finally received their incentive, which concluded the session.



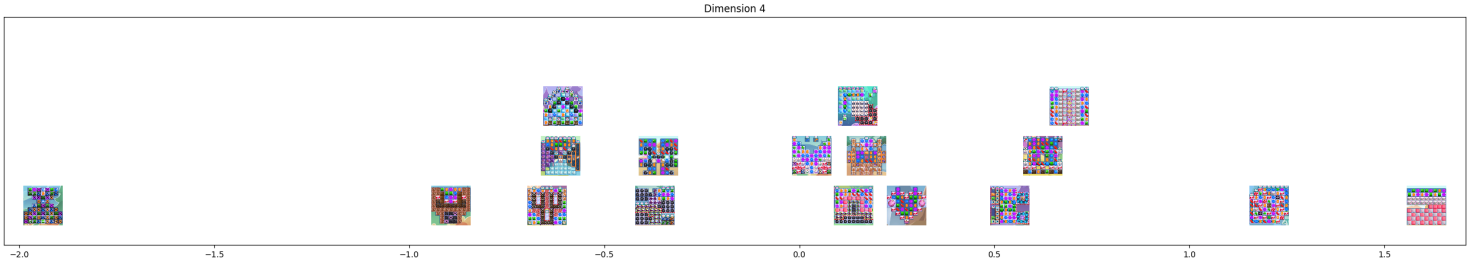

(a) ccs-img, dimension 1. P1: “From bespoke to generative”. P2: “Irregularity of shapes”. Consensus: “Shape irregularity (from square blocks to non-contiguous shapes)”




(b) ccs-img, dimension 2. P1: “Inverse difficulty (from hard to easy), i. e. more blocks requiring multiple ? (interactions?)”. P2: “Roundness, how much does it look like a circle”. Consensus: “Level difficulty (from low to high)”

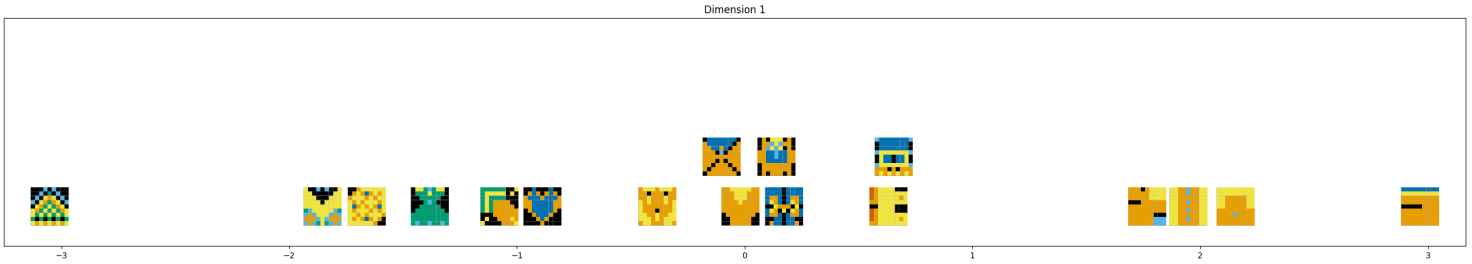



(c) ccs-img, dimension 3. P1: “Diagonal angularity (from squareness of level design to diagonalness)”. P2: “Amount of candy/fruit blocks compared to other blocks (just a guess)”. Consensus: “Squareness (from vertical/horizontal to diagonal shapes)”

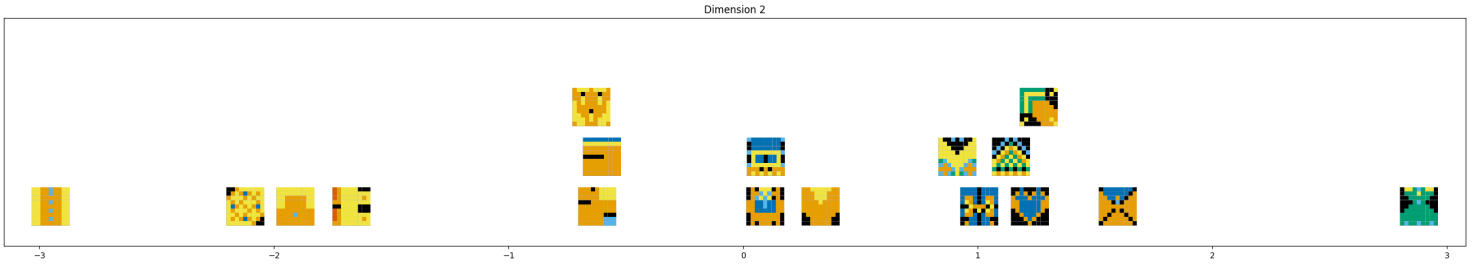



(d) ccs-img, dimension 4. P1: “Most to least likely generative (guess)”. P2: “Brightness (from dark to light)”. Consensus: “Brightness of tile colours (from dark to light colours)”

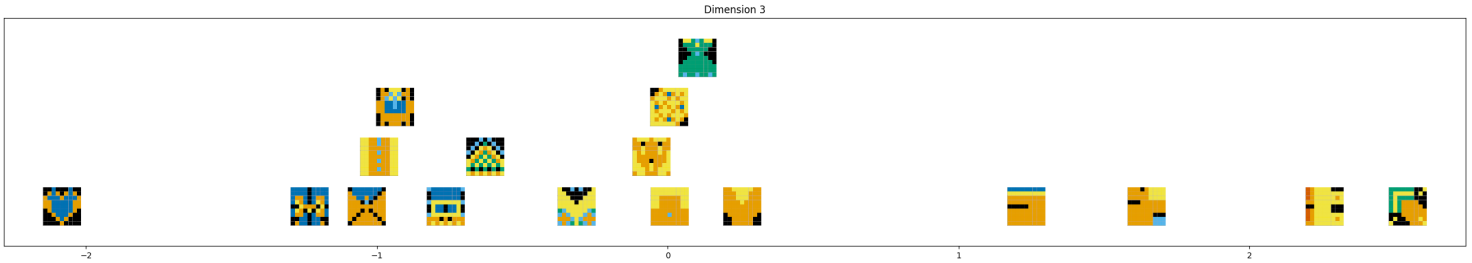
Figure 6.11: Labelled embedding dimensions for condition *ccs-img*



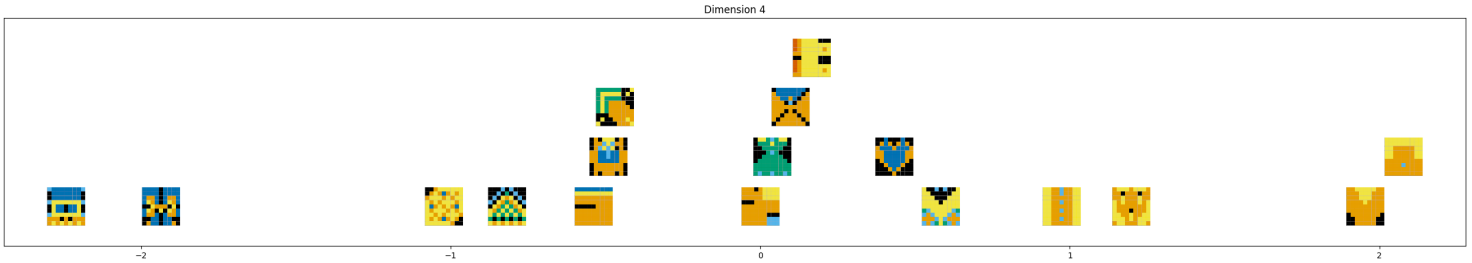

(a) ccs-pat, dimension 1. P3: “Number of straight horizontal lines; pixels are grouped”. P4: “From intricate to simple; Colors tend to loose darker shades from left to right”. Consensus: “Pattern complexity (from intricate to simple patterns)”




(b) ccs-pat, dimension 2. P3: “More green and black, less yellow and blue as x increases”. P4: “Colors tend to go from orange-yellow colorspace to black to green-blue colors (CMY–Black–RGB); Patterns tend too go lateral-symmetric-radial”. Consensus: “Tile colours (from bright to dark)”

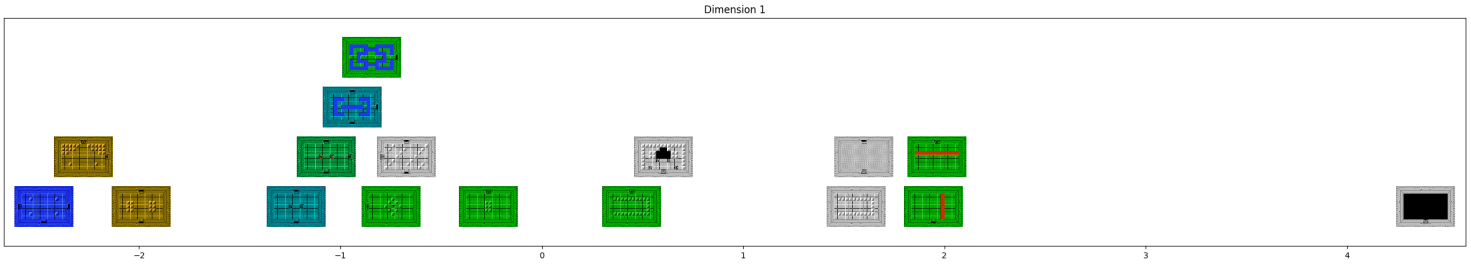



(c) ccs-pat, dimension 3. P3: “Blue swaps for green and yellow”. P4: “Pattern from lateral symmetric”. Consensus: “Pattern symmetry (from vertical symmetric to asymmetric)”

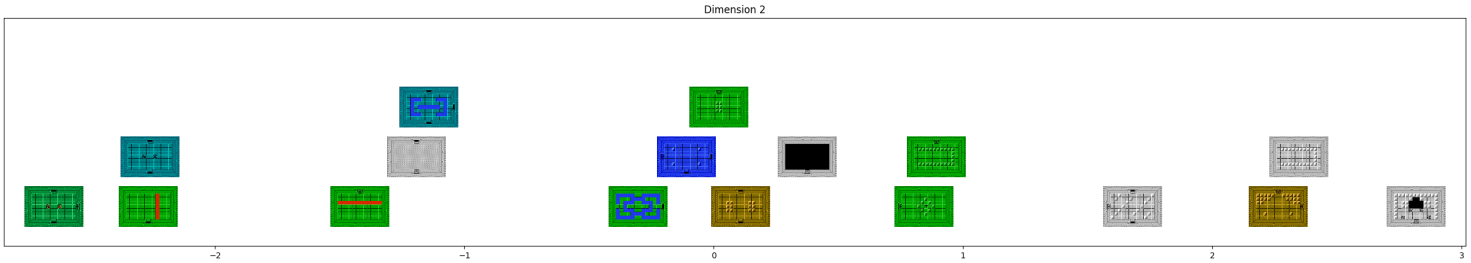



(d) ccs-pat, dimension 4. P3: “More orange, less blue as x increases”. P4: “The patterns tend to move up to down going left to right”. Consensus: “Tile colours (from blue to orange)”

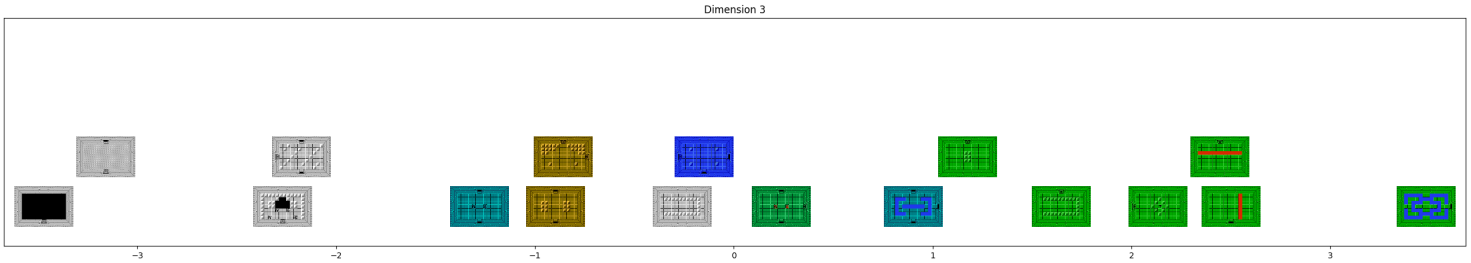
Figure 6.12: Labelled embedding dimensions for condition *ccs-pat*



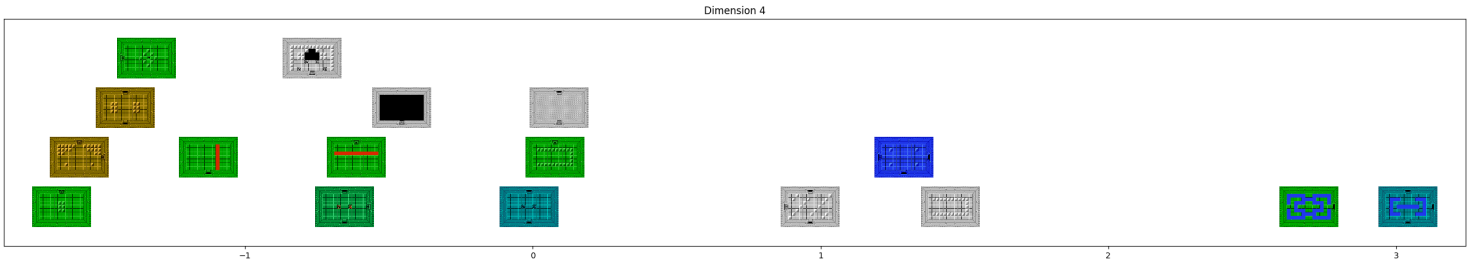

(a) loz-img, dimension 1. P5: “Symmetrical arrangement of tiles high – low”. P6: “Asymmetry”. Consensus: “Symmetry (from high to low)”




(b) loz-img, dimension 2. P5: “Interesting patterns low – high”. P6: “Complexity”. Consensus: “Interesting patterns”

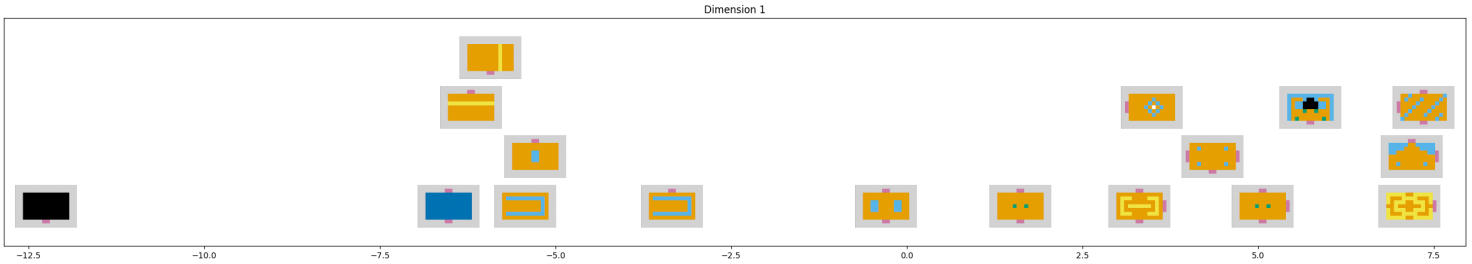



(c) loz-img, dimension 3. P5: “Colours variation low – high”. P6: “Incohesion”. Consensus: “Colourfulness (from low to high)”

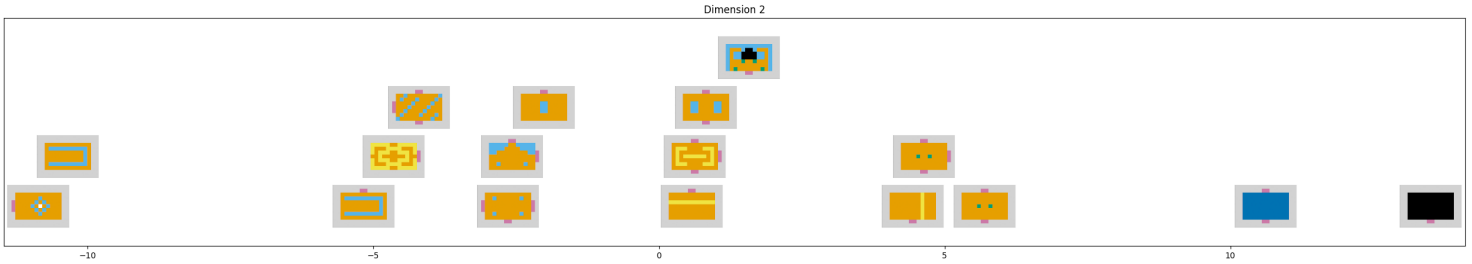



(d) loz-img, dimension 4. P5: “Coherence low – high”. P6: “Complexity”. No consensus

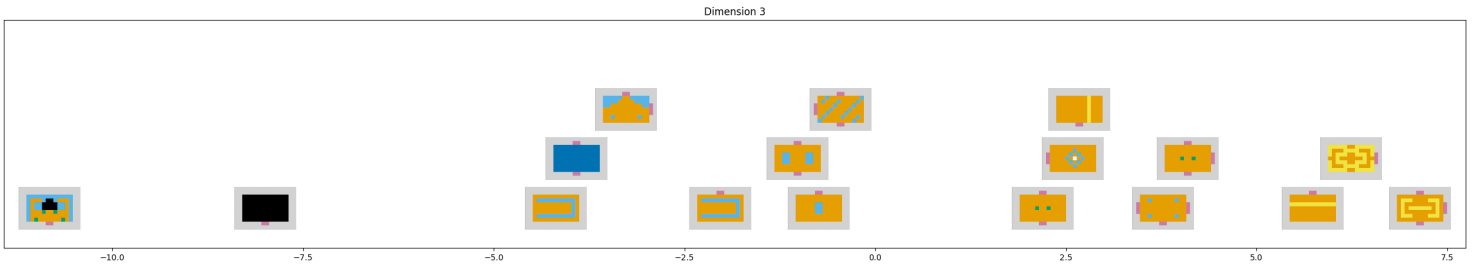
Figure 6.13: Labelled embedding dimensions for condition *loz-img*



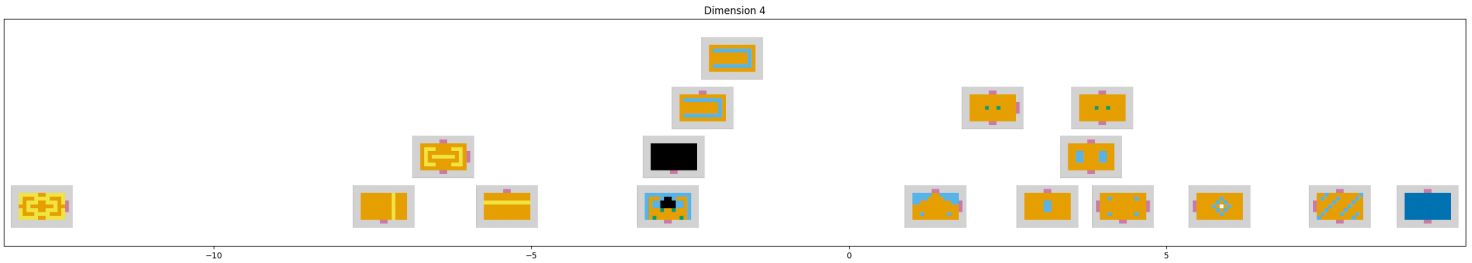

(a) loz-pat, dimension 1.P7: 'From "game started" to "20 minutes in"'. P8: "Connected components of color (not necessarily of different colors), ignoring outer side rectangles". Consensus: "Complexity (from low to high)"




(b) loz-pat, dimension 2. P7: 'I can't unsee Zelda, so I'm gonna say from "More hidden secrets" –> "less hidden secrets" or "Exploration-focused gameplay" –> "Challenge-focused gameplay"'. P8: "No idea". No consensus




(c) loz-pat, dimension 3. P7: '"Closed-up areas" –> "Open-ended areas"; Maybe something like "linear progression" –> "Open worlds"; Colour/amount of yellow seems to be a factor too. Maybe "from coast to desert"???; Theme'. P8: 'Different tile type "theme"; cutscene –> start –> water –> land –> yellow (?)'. Consensus: "Level theme"




(d) loz-pat, dimension 4. P7: 'More unique to less unique? In the sense of "tile is never repeated in game" –> "tile is often repeated"; Maybe theme again. Yellowish to blueish; Challenging desert section to more relaxed water section. Hard to easy?'. P8: 'Yellows –> blues'. No consensus

Figure 6.14: Labelled embedding dimensions for condition *loz-pat*

### 6.4.4 RESULTS

Our findings reflect a diverse range of perspectives, echoing our participants' varied backgrounds. They noted the difficulty of the labelling task and agreed that discussions within the groups benefited their individual insights. While some groups found it easy to determine consensus labels, not all groups succeeded. All embedding dimensions with their assigned labels are shown in Figures 6.11 to 6.14. We compare the consensus labels between conditions in Table 6.3. Below, we summarise the most important findings.

*ccs-img* The participants believe shape to be of high importance. Their labels include 'squareness' (dim. 3) and 'shape irregularity' (dim. 1). One participant further mentions 'roundness' (dim. 2). The group assigns 'brightness' of tile colours as another label (dim. 4).

*ccs-pat* 'Tile colours', and its range from bright to dark as well as from blue to orange, was assigned as a label to two dimensions (2 and 4). The group further agreed on 'pattern complexity' (dim. 1) and 'pattern symmetry' (dim. 3).

*loz-img* The group highlighted 'symmetry' and 'colourfulness' as possible labels (dim. 1 and 3, respectively) and agreed on 'interesting patterns' (dim. 2). One participant further mentioned the 'complexity' of patterns in relation to two dimensions (2 and 4).

*loz-pat* One participant misinterpreted the tile colours to indicate functionality (blue for water, yellow for desert) and thus focused on game design aspects, describing 'themes' of different levels (dim. 3) and the difficulty of solving them. However, the participant also commented on the repetition of tiles (dim. 4), alluding to the distribution of tile types. The group only gave one relevant consensus label: 'pattern complexity' (dim. 1).

Table 6.3: Consensus labels for dimensions of the perceptual embeddings (rows) as proposed by individual focus groups per condition (columns) in study 2 (Section 6.4).

| Dim. | ccs-img | ccs-pat | loz-img | loz-pat |
|---|---|---|---|---|
| 1 | Shape irregularity (from square blocks to non-contiguous shapes) | Pattern complexity (from intricate to simple patterns) | Symmetry (from high to low) | Complexity (from low to high) |
| 2 | Level difficulty (from low to high) | Tile colours (from bright to dark) | Interesting patterns | — |
| 3 | Squareness (from vertical/horizontal to diagonal shapes) | Pattern symmetry (from vertical symmetric to asymmetric) | Colourfulness (from low to high) | Level theme |
| 4 | Brightness of tile colours (from dark to light colours) | Tile colours (from blue to orange) | — | — |

## 6.5 DISCUSSION

We discuss the findings from our first study (Section 6.3) to contribute to our first research question (*Which existing metrics approximate the human similarity perception of grid-based video game levels best?*). For this, we primarily focus on a metric's approximation capabilities as quantified by mean squared error in our comparison of similarity matrices, since computing pairwise similarities comes closest to the application scenarios in game development and research. We support these findings with the result from the inter-rater agreement analysis (Section 6.3.4.3), which allows for a more direct, albeit limited, comparison of the metrics to the participant judgements.

The results suggest that CV-based similarity metrics (CLIP, DreamSim) provide the overall best approximation to the collected participant judgements, outperforming the PCG expert metrics and general-purpose metrics. In particular, results for the artificial neural network-based image embedding DreamSim exhibit the overall lowest approximation error and highest agreement when compared to our participant judgements. While this may be unsurprising, given that the image embedding was specifically fine-tuned to align with human perception of synthetic natural images, our results confirm that this equally benefits similarity estimation of video game levels. Yet,

accuracy is not everything. A downside of artificial neural network-based image embeddings however is their size, complexity and dependence on specialised hardware for fast inference. For example, DreamSim requires a CUDA-compatible GPU with 1.6 GB memory available to load the model (additional memory required to compute image embeddings). This can be problematic, considering the limited resources available when relying on such metrics in games at runtime, in particular on mobile devices. Furthermore, sub-symbolic approaches (artificial neural networks) are limited in their transparency, as it is more difficult to explain why a particular pair of levels is attributed to high similarity. In contrast, symbolic approaches (the PCG expert metrics) with their transparent design choices can more easily be broken down into specific rules.

Between the three expert metrics from the PCG literature (with a total of eight configurations), we can observe big differences in performance. The Symmetry metrics in any of its configurations only seem to capture a single aspect relevant to our sample of participants (cf. Table 6.3), yielding high approximation errors and overall low, often even negative agreement. With the closely related Tile Pattern and Tile Frequencies (identical to the tile pattern size $1 \times 1$) metrics, we observe a correlation in the results: the larger the patterns, the higher the metric's approximation error (Figure 6.5). This correlation has a simple explanation: the larger the patterns, the fewer patterns there are in a level to compare. That is to say, a lower granularity of patterns (in the extreme case $1 \times 1$, i. e. Tile Frequencies) allows for a more nuanced comparison between levels. If there is little data to compare (e. g. only a few large $4 \times 4$ patterns) it will be difficult to determine whether two levels are slightly more similar than another pair. This can lead to high errors in our similarity matrix analysis. Furthermore, our collection of stimuli is a particularly small dataset, which likely does not provide much overlap in patterns across levels. This explanation is supported by the results on Legend of Zelda levels (*loz-img*, *loz-pat*), which share more patterns due to the common layout of rooms. Consequently, Tile Patterns of size $2 \times 2$ and $3 \times 3$ perform much better on levels from this title than on Candy Crush Saga levels. Tile Frequencies being the third-best approximating

metric is surprising, given that this metric only compares the number of different tiles in a level but entirely disregards their positions. Nonetheless, our results suggest that Tile Frequencies is a reliable PCG expert metric across all experimental conditions.

The effect of different level structures on metric performance can be observed in the results for Hamming Distance, the best out of two general-purpose metrics. Hamming Distance performs much better on Legend of Zelda than on Candy Crush Saga levels. As mentioned above, the common structure of Legend of Zelda levels puts a focus on the differences in the room interiors between levels. All rooms are the same size and are surrounded by walls and doors. It is thus more important whether rooms are filled with obstacles, enemies or staircases. For participants in our first study, these details may have also been the most similarity-relevant criteria. In contrast, Candy Crush Saga levels can have very different shapes and compositions, making it a more difficult task for a tile position-sensitive metric. Given a more homogeneous collection of Candy Crush Saga levels, Hamming Distance might have performed better on this title. More work is required to test this hypothesis. Hamming Distance has a competitive performance when levels share a common structure and differences between them consist in smaller but important details.

One may argue that in our first study participants with experience of the relevant video game titles (Candy Crush Saga, Legend of Zelda) or similar ones from the same genre might have a better idea of the expressive range amongst levels, therefore making different similarity judgements. Even more so, the perception of expressive ranges between participants, even with similar experiences, might differ. Yet, the design of the triplet judgement task as two alternative forced choice aims to prevent exactly these variances. Participants are only asked to make a simple binary choice, rather than a more nuanced judgement of similarity.

Our second study (Section 6.4) allows us to probe this assumption, and highlights two principal similarity-relevant criteria in this specific scenario as an answer to our second research question (*What are the dimensions that govern players' similarity perception?*). First, the design of patterns in terms of

shape ('irregularity', 'squareness'), symmetry and tile composition ('complexity'). Second, the choice of sprites ('tile colours', 'colourfulness' and 'brightness'), which might explain the performance advantage of image embedding metrics, DreamSim and CLIP, over the tile representation-based metric. While symmetry along various axes is already covered by specialised metrics compared in our study, most other criteria are not explicitly accounted for. In particular, the visual qualities of the sprite design are not reflected in tile representations. Moreover, participants also thought about gameplay-specific criteria, like level 'difficulty' and game narrative ('themes'), which are not yet covered by any metric. For Legend of Zelda, it is easier to infer the gameplay of a level only from its layout, since most of its elements (walls, doors, steps) are static and players can easily imagine how to move through a room. However, game dynamics are much more complex and random in Candy Crush Saga, where tiles that are cleared from the level are replaced by new ones falling from the top. We discuss the difficulty of extending our study setup to include aspects of gameplay in future work (Section 8.1). All in all, in the context of video games, expert metrics find their purpose in providing robust performance in a dynamic, potentially low-resource environment. These findings can contribute to the future development of custom metrics that meet these requirements and are more closely aligned with human perception.

### 6.5.1 STUDY LIMITATIONS

The present work focuses on visual similarity estimation in two tile-based video games. We note two limitations on *generality*. First, we have not taken into account other game genres beyond tile-based games. Moreover, constrained by the triplet comparison data collection methodology, we could only include a limited amount of stimuli. We tried to mitigate this constraint by systematically selecting stimuli for diversity and through our mixed design. While we selected our two game titles to capture diversity and popularity in the space of tile-based games, there exists much more

variation in video game titles that could not be accounted for. Second, the same applies to variation within the levels of each title which, despite our systematic procedure, could not be captured in its entirety. Crucially though, we hold that the dimensions governing similarity between levels here can inform stimulus selection in future studies extending our work. Moreover, we are confident that the choices of games and levels in this work reflect many use cases in the industry.

Beyond limitations to generality, we note that our study only considers similarity judgements of tile-based video games with respect to *visual information*. We agree with related work on player modelling in that functional and dynamic elements of gameplay such as power-ups or tile cascades are also important determinants of player perception, experience and behaviour. Minor differences in the layout of any two levels may have little effect on their visual similarity, yet might make a big difference in terms of gameplay. While many of these elements can be identified visually, we expect players' similarity assessment to be considerably shaped by their active interaction with them. This research thus represents a specialised lens on visual and static game content, contributing to the future development of holistic models of players' similarity judgement.

The setup of experimental conditions and in particular the fact that CV metrics receive different inputs depending on the *Representation* of the condition, limits our study in that we do not cover all possible comparisons for the image-based similarity metrics. We thus do not investigate the discrepancies between the participant judgements across visual representations while keeping the metric representation static. However, we deliver on our plans. As the input to the image-based metrics is varied based on the condition to match what the participants see, we get direct comparisons for how well the image-based metrics approximate the participant judgements for that condition. In this work, we focus on this aspect and leave other comparisons for future work. We acknowledge that the mapping from *img* to *pat* representation does encode some assumptions around the similarity of the different level objects. However, these assumptions do not stem from

our own biases but instead rely on the experience of the domain experts for the respective games.

We identify two limitations stemming from the design of the stimuli selection process. First, selecting stimuli that cover the wide range of level designs increases the difficulty of the triplet judgement task. We argue, however, that the data collected from forced-choice judgement tasks is still useful as overall relations between stimuli are captured in the aggregated judgements of a large group of participants. Our results confirm this; despite the difficult triplet combinations, the best metrics compared in our work were able to approximate stimuli relations with very little error. Second, our conclusions about the performance of CLIP are insofar limited as we also *leveraged CLIP in the stimuli selection procedure*. This choice in the selection process allowed us to maximise the diversity of levels, thus benefiting the fair evaluation of all metrics, at the expense of introducing a bias on the performance of a single model. We chose CLIP for the selection procedure as we expected it to be amongst the strongest candidates, thus leaving more space for fine-grained differentiation between the other metrics. And despite our use of CLIP in the selection process, our results point to a different CV-based metric as the best-performing metric: DreamSim. We considered multiple other stimuli selection strategies. Here we discuss the advantages and drawbacks of three options which ultimately led us to adopt the approach presented earlier. The first alternative, random sampling of stimuli, is the most unbiased approach, yet unlikely to cover the diverse level design space (e. g. out of 2,792 Candy Crush Saga levels we were only able to select 17). Second, a selection of stimuli informed by a pilot study is also relatively unbiased. However, participants would have to assess an overwhelmingly large amount of stimuli (*ccs*: 2,792 levels; *loz*: 225 levels). A cognitively very demanding task that would require additional recruitment of reliable participants. Third, instead of CLIP, we could use a different embedding (e. g. another CV-based model). While this would not introduce a bias in favour of any of the metrics compared in this work, it would nonetheless introduce a bias towards a different metric for which the relations to the other metrics are not explicitly assessed.

We leveraged inter-rater agreement statistics to facilitate direct comparisons of metric performance with raw data from individual participants. However, overall we only found *low to moderate agreement* between participants and metrics. Since triplet comparisons only require binary decisions, we had no information on participants' confidence in their ratings. We deliberately chose not to leverage disagreement ratios as a proxy for rating confidence due to the low number of samples per stimulus. Future studies could include an additional confidence rating or leverage a different rating task to facilitate a closer comparison between human judgements and the continuous similarity values provided by metrics.

The focus group labelling task in our second study is a naturally noisy process because 1) labels are subjective and 2) the dimensions of the perceptual embeddings are the product of noisy participant judgements. Given the difficulty of the labelling exercise, some groups were not able to provide a consensus label for some dimensions. Yet, rather than labelling exhaustively, our goal was to obtain as much relevant information as possible. The labels identified in our second study nonetheless are a valuable resource to explain the total variance of the similarity judgements. We were only able to obtain data from a relatively *small group of participants* per condition. More participants would have provided higher robustness, as the quality of consensus labels benefits from a variety of perspectives. However, our decision on an on-site study to limit distractions and foster discussion imposed constraints on how many participants could be possibly recruited. Given the complexity of the domain and task, we hold that our findings provide good pointers for future work. Moreover, We published our dataset and interpretation scales to enable other researchers to further validate and extend our findings.

# Chapter 7

# RELATED WORK

In this chapter, we report on the related work for the four research projects introduced in Chapters 4 to 6. While no prior work evaluated the benefits and drawbacks of using generative models with evolutionary algorithms before our systematic study (Chapter 4), we collect some related works that effectively use models' latent spaces. Drawing from different fields of research, we review several measures of diversity for their applicability to generative machine learning. We then give an overview of related work on dataset biases in machine learning, in particular computer vision and classification tasks. The effects of dataset biases on generative models have received little attention. Drawing on these insights, we point out that reducing biases by simply adding more examples to a dataset is often not a trivial task, requiring more principled approaches. While this gives further motivation to apply our work to efforts in diversity, equity, and inclusion (DEI), such work is beyond the scope of this thesis. As related work, we thus survey different methods that address the under-representation of minority groups in generative models and compare them to our *mode balancing* approach (Chapter 5). Finally, we cover work related to the human perception of similarity by outlining the conventional data collection methodology, highlighting the lack of studies on similarity perception in video games, and comparing our study (Chapter 6) to related work in computer vision.

CONTENTS

## 7.1 DATA BIASES IN MACHINE LEARNING

Biases in datasets have been studied from the perspective of machine learning under different names, including *sample selection bias* (Heckman, 1979; Zadrozny, 2004), *covariate shift* (Shimodaira, 2000), and *class imbalance* (Japkowicz & Stephen, 2002). The authors focus on how dataset biases can limit the generalisability of learned models, in particular for classification tasks in supervised learning. Torralba and Efros (2011) present an analysis of common computer vision datasets for object recognition and propose measures to quantify *capture bias*, *category* or *label bias*, and *negative set bias*. A de-biasing method is presented in Khosla et al. (2012) that explicitly defines the bias associated with each dataset by learning an individual bias vector. The method attempts to then approximate a model of the common "visual world" with better generalisation ability by undoing the bias from each dataset. Tommasi et al. (2017) pick up these two previous works, extend the dataset bias analysis to deep learning-based convolutional feature extraction methods, and propose an additional measure to quantify the performance of the de-biasing method (Khosla et al., 2012). They conclude that the evaluated dataset biases can be reduced but not eliminated.

Technical impact of biases

With the increasing use of computer vision technologies, the attention shifted to the technology's impact on the general public and the effect of dataset biases on individuals, specific demographics and society at large (Lohia et al., 2019). As high-stakes decisions in applications (e. g. credit, employment, criminal justice) become automated, pressure is increasing to address dataset biases and to ensure *fairness* in classification (Dwork et al., 2012). A mathematical formulation of fairness has been proposed by Friedler, Scheidegger and Venkatasubramanian (2016). Frameworks that aim to ensure 'non-discriminatory' predictions have been presented as *equalised odds* (Hardt et al., 2016) and as *disparate mistreatment* (Zafar et al., 2017). Recent studies further look at race and gender bias in commercial gender classification systems for image data (Buolamwini & Gebru, 2018), and gender biases in natural language and its effect on image captioning and

Social impact of biases

semantic role labelling (Zhao et al., 2017; Hendricks et al., 2018). Various authors highlight the problem of *bias amplification*, where bias is not only learned from data but increased by a model (Bolukbasi et al., 2016; Zhao et al., 2017; Hendricks et al., 2018).

Balancing datasets

Crucially, simply fixing a dataset bias by balancing the number of specific examples is not trivial and not always possible. In the discriminative setting, Stock and Cisse (2018) find that models classify images as "basketball" based on the presence of a black person, even though white people appear equally as often in "basketball" images. The authors hypothesise that this stems from the dataset containing more images of white people overall. The classification model thus, responding to a spurious correlation, assigns the "basketball" label based on the appearance of the person rather than the activity. In this case, improving the dataset would require balancing all other classes (e. g. "ping-pong", "rugby", "baseball", "volleyball") to a similar degree as the "basketball" class. In the generative setting, without a clear separation of images by classes, such imbalances are even more difficult to identify and address.

## 7.2 DE-BIASING GENERATIVE MODELS

Existing work on data biases in generative models primarily focuses on the under-representation of minority groups. The objectives of different approaches range from mitigating such biases to improving minority coverage, i. e. achieving better image fidelity for under-represented data examples. Some approaches employ a weighted sampling scheme where weights are derived from density ratios, either via an approximation based on the discriminator's prediction (Lee et al., 2021) or via an additional probabilistic classifier (Grover et al., 2019). Others propose an implicit maximum likelihood estimation framework to improve the overall mode coverage in GANs (Yu et al., 2020). These methods either depend on additional adversarily trained models or on more problem-specific solutions through hybrid models that do not necessarily generalise to other settings. Our *mode balancing*

approach (Chapter 5), instead, has two major benefits over this related work. First, it is model-agnostic and thus potentially applicable to a wide range of network architectures and training schemes. Second, it only adds an offline pre-computation step before conventional training procedures and during training solely intervenes at the data sampling stage.

Authors of previous work further argue for increased diversity but do not evaluate explicit measures of diversity. Results are reported on the standard metrics IS, as well as FID and PR which rely on the training dataset for reference (Section 2.4). Consequently, they can only estimate sample fidelity and mode coverage as present in the data, but not independently of it. In our work on *mode balancing* (Chapter 5), we instead evaluate measures explicitly designed to objectively quantify diversity without relying on dataset statistics for reference (Section 2.5).

# Chapter 8

# CONCLUSIONS

## Contributions

The contributions presented in this thesis cover the topic of diversity in generative machine learning across multiple research modalities, including conceptual and analytical work, formalisation, systematic experimentation, and studies with human participants.

We coined the term *active divergence* to describe a common theme in the artistic uses of generative models where people consciously break, tweak or otherwise intervene in a data-driven generative process to produce culturally valuable but from a pure modelling perspective sub-optimal artefacts. We presented a taxonomy and survey of such active divergence techniques in generative deep learning, highlighting their potential for computational creativity research. We developed a formal framework to automate generative deep learning for artistic purposes that provides opportunities to hand over creative responsibilities to a generative system. For this, we defined the conventional generative deep learning pipeline and contrasted several deviations in the artistic settings, resulting in an overview of targets for automation.

To analyse the capabilities and limitations in expressivity of generative models, we performed a series of experiments evaluating the output diversity of a VAE in a principled way. Empirical evidence demonstrates that QD search in parameter space yields a more diverse collection of outputs than search in the VAE latent space. This suggests that the VAE used in the experiments is limited in its expressivity, i. e. the capacity to generate artefacts beyond the training examples.

To increase the output diversity of generative models, we introduced *diversity weights*, a method to derive a weight vector over the examples in a training dataset, which indicates their individual contribution to the dataset's

overall diversity. With this *mode balancing* approach, we changed the conceptual objective from covering all modes in the training data exactly to balancing them such that they are equally likely under the model. The weights allow training a generative model with a diversity-weighted sampling scheme, such that the model's output diversity increases. Our work is motivated by potential benefits for computational creativity applications which aim to produce a wide range of diverse output for further design iterations, ranging from artistic to constrained to scientific creativity. We also highlight a connection to issues of data bias in generative machine learning, in particular data imbalances and the under-representation of minority features. The impracticality of easily mitigating data imbalances in an unsupervised setting further motivated our work. In a proof-of-concept study, we experimentally demonstrated that our method increases model output diversity when compared to the standard GAN training process. The results highlight a trade-off between artefact diversity and artefact typicality, i. e. the extent to which an artefact is a typical training example. Our method provides control over this trade-off via a loss balance hyperparameter.

As a step towards better measures of diversity, in Chapter 6, we sought to answer two research questions: (1) Which existing metrics approximate the human similarity perception of grid-based video game levels best? And, as a stepping stone toward the development of better metrics, (2) which dimensions govern the similarity perception in this scenario? Of immediate practical relevance, we probe the common belief that the development of good similarity metrics requires a deep understanding of games as an application domain. To this end, we compared 7 metrics in 12 configurations, grouped into custom-made PCG, general-purpose, and computer vision metrics. We found that the DreamSim image embedding (S. Fu et al., 2023) exhibits the overall best performance (low overall approximation error and high agreement with human participants), followed by the CLIP embedding model (Radford et al., 2021) from the same group of CV-based metrics. Since such artificial neural network-based approaches can be too resource-intensive for deployment within a video game, we recommend their use for the offline generation of video game assets. As an alternative, for in-game

use, we found that Tile Frequencies (Summerville et al., 2017), a simple baseline metric from the PCG literature more suitable for low-resource environments, shows the next-best performance. Furthermore, Hamming Distance is competitive with Tile Frequencies when levels share a common structure and differences between them consist of smaller but important details, e. g. our collection of Legend of Zelda levels.

However, our findings also show that there is room for improvement. Opportunities for advancements of similarity metrics were revealed through our second study, in which we asked focus groups with relevant experience to interpret the dimensions underlying the similarity judgement as captured by our data. Participants particularly highlighted the importance of pattern design in terms of shape, symmetry and tile composition, as well as the choice of tile sprites as similarity-relevant criteria of human perception in this specific domain. Our findings contribute to a better understanding of similarity estimation in people and its alignment with existing metrics for tile-based video game levels, and through this inform similarity estimation via computational metrics.

Together, our findings can advance a wide range of tasks in research and industry, from developing better player models, more satisfying PCG, believable NPCs, and increasingly plausible automated play-testing approaches. They thus benefit both the game AI and game user research communities and enable new work at the crossroads.

Furthermore, we made several code repositories publicly available.

## 8.1 FUTURE WORK

Here we discuss potential extensions to the work presented in this thesis.

The formulation of our framework for the automation of generative deep learning for artistic purposes (Chapter 3) stems from a time before the development of large consumer generative interfaces such as Google Gemini and OpenAI's Dall-E and ChatGPT. Future work could thus study how deep learning researchers, practitioners and artists work with more recent

generative systems, in particular where they have added and could add levels of automation. Some of the techniques that artists apply, such as dataset curation and iteration, as well as the selection of generated outputs, are promising avenues for automation and require further investigation. Applying our framework to practical projects would further provide demonstrative examples of how some of the challenges in automation can be tackled and show the surprising results that automation can afford. For the evaluation of such demonstrative examples, we consider the FACE descriptive model of creative acts an appropriate choice (Colton, Charnley & Pease, 2011; Pease & Colton, 2011). In the formulation of our framework, we only briefly mentioned the automation of creative responsibilities via the usage of machine learning (ML) models. Multiple models can be trained and deployed within the same system or communicate across systems, adding *interaction-awareness* as an aspect of creative self-awareness (Linkola et al., 2017). Our framework could thus be extended by considerations for organisational structures, in which we think of individual models as agents in a multi-agent system. To use our framework in co-creative applications, augmenting a system with the ability to communicate its adjustments and intentions would be especially beneficial. Moreover, to address our framework's limitations, further work is needed to consider applications which use generative DL but are not artistically focused. This could potentially inform a more general automated ML framework, which would benefit from more formal definitions.

More work is needed to extend our first results on the limitations of generative models (Chapter 4). For better generalisability, a large-scale systematic study of the individual parts of our setup and their influence on expressivity is needed. This could include testing different priors for a VAE latent distribution, the size of the training dataset, mapping training examples to a higher-dimensional latent space, and different architectures of the generative model. Comparing the performance of a VAE to that of auto-regressive, adversarially-trained, flow-based or transformer-based models could highlight strengths and weaknesses of individual modelling techniques and architectures. It would allow for a more general understanding of their generative capabilities and limitations. Furthermore, following

our discussion in Section 4.6, we propose future work to better understand the compression afforded by a generative model's latent space and its benefits in facilitating a search space of smaller dimensionality. For this, we suggest using an experimental setup in which the complexity of a search space can be arbitrarily increased to find the point at which QD search in latent space outperforms search in parameter space. For example, through procedurally generating grid-based video game levels for a title like Candy Crush Saga, Legend of Zelda, or Overcooked (Carroll et al., 2019; Fontaine, Hsu et al., 2021). In such a scenario, the complexity of parametric search grows exponentially as the size of a level increases, while the dimensionality of the corresponding latent search space could, subject to some constraints and trade-offs, be held constant. This would allow us to quantify the degree of complexity at which the compression of a learned latent space facilitates QD search which would otherwise have worse performance or be entirely infeasible in parameter search.

Our work on *diversity weights* can be improved in several ways. First, by refining our method, in particular, the training procedure to improve overall sample fidelity. For this, a thorough analysis and systematic comparison to related work is needed. Furthermore, the loss balance hyperparameter could be tuned automatically by including it as a learnable parameter in the optimisation procedure. Apart from our gradient descent approach, there might be alternative exact or approximate methods for the diversity weight optimisation, e. g. through analytical solutions or via constraint optimisation.

Second, generalisability could be extended beyond the proof-of-concept. For this, experiments with other generative models and on bigger and more complex datasets are required to demonstrate the scalability of our approach. Since our method is architecture-agnostic, there remain many opportunities for future work to understand the effect and potential benefits of our method in other modelling techniques. As GAN training is notoriously unstable and requires careful tuning, other modelling techniques might prove more appropriate. Further experiments with human data, e. g. images of human faces, or datasets that otherwise concern people can demonstrate

the capability of our method to mitigate issues of DEI resulting from data imbalances. Moreover, empirical studies will be necessary to investigate how the shift from mode coverage to mode balancing can support diversity in a large range of CC applications.

There exists an inconsistency between the Diversity of order q and the Vendi Score (VS). The authors of both measures claim that their measure yields an *effective number* (Hill, 1973), representing the count of absolutely dissimilar items of equal abundance in a dataset. However, given the same data, the measures do not agree, producing different scores and thus different 'effective numbers'. Future work is required to determine which of the two measures does in fact give an *effective* estimate and, if the two measures are related, what transformation of one score to the other explains their disagreement.

Our findings on human similarity perception can inform metric selection in game development and as an element of research studies on games more generally. Moreover, they highlight potential avenues for improvement of existing metrics and the development of future ones. We particularly advocate supporting further research on this topic through various uses of machine learning. To select a small subset of stimuli from a large dataset that covers the variation in the dataset, an auto-encoding artificial neural network can be trained on the full dataset. A subset of stimuli can then be selected based on their pairwise distances in the model's latent space. Other stimuli selection strategies may be applied in future work: random sampling, grid-based selection, etc. To further advance data-driven metrics, we can fine-tune an existing image embedding on a curated dataset of annotated video game levels to obtain a specialised embedding space for the video game domain. Moreover, as DreamSim (Section 6.2.2) has demonstrated, we can bootstrap an ensemble of metrics to train a prediction model of human judgement on top of the metrics' respective calculations. Yet, these efforts have to be assessed in comparison to the performance of much simpler general-purpose metrics. In the video game context, in particular for applications on-device, only limited resources might be available which need to be managed carefully. This work can inform which metrics to

include in further benchmarks. We note that this work contributes to the bigger effort of developing holistic models of human similarity judgement in games. Our study setup leaves open for future work further investigations of agreement with CV metrics. Our study shows that when participants and image-based metrics are given the same level representation (*img* or *pat*), CV metrics perform best overall. But further study is needed to understand their performance in scenarios where participants are shown the same level screenshots (*img*) between conditions, while the input to image-based metrics is changed from *img* to *pat*. The publication of our data and implementation opens these avenues for future work to the whole research community. More work is needed to extend our analysis to other video game titles, as well as alternative mappings from level objects to abstract colour tiles. While we have focused on the perception of visual similarity in static content, we expect players' similarity judgement to be also shaped by the dynamic gameplay behaviour that levels afford, and the experiences they are expected to provide. Consequently, an important avenue for future work will be to understand how these static and dynamic aspects can be combined. For example, through representations that can more explicitly encode gameplay, as used in the Video Game Affordances Corpus (VGAC) (Bentley & Osborn, 2019). This is particularly important for Candy Crush Saga, where complex game dynamics make it difficult to infer the gameplay of a level only from a static image of its initial state. If we consider video games in which players freely move a character around in a three-dimensional world, it becomes obvious how important experiencing the gameplay of a particular level is to judge its similarity to another level. In future studies, such complex gameplay dynamics thus require participants to play a video game before being able to make a meaningful judgement. Finally, while the focus of Chapter 6 was on similarity, we advocate for research into how well the identified metrics can estimate the human perception of diversity as a natural next step toward supporting a wider range of game AI applications.

# BIBLIOGRAPHY


A Deep Dive Into Exploring the Preference Hypervolume. (n.d.).

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A Learning Algorithm for Boltzmann Machines. *Cognitive Science*, *9*(1), 147–169.

Agarwal, S., Wills, J., Cayton, L., Lanckriet, G., Kriegman, D., & Belongie, S. (2007). Generalized Non-metric Multidimensional Scaling. *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, *2*.

Agnese, J., Herrera, J., Tao, H., & Zhu, X. (2020). A survey and taxonomy of adversarial neural networks for text-to-image synthesis. *WIREs Data Mining and Knowledge Discovery*, *10*(4).

Alvarez, A., Dahlskog, S., Font, J., Holmberg, J., & Johansson, S. (2018). Assessing Aesthetic Criteria in the Evolutionary Dungeon Designer. *Proceedings of the 13th International Conference on the Foundations of Digital Games*.

Arjovsky, M., & Bottou, L. (2017). Towards Principled Methods for Training Generative Adversarial Networks. *International Conference on Learning Representations*.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein Generative Adversarial Networks. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning* (pp. 214–223, Vol. 70). PMLR.

Arora, S., Ge, R., Liang, Y., Ma, T., & Zhang, Y. (2017). Generalization and Equilibrium in Generative Adversarial Nets (GANs). In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning* (pp. 224–232, Vol. 70). PMLR.

Azadi, S., Olsson, C., Darrell, T., Goodfellow, I., & Odena, A. (2019). Discriminator Rejection Sampling. *International Conference on Learning Representations (ICLR)*.

Barratt, S., & Sharma, R. (2018). A Note on the Inception Score. *ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*.

Bau, D., Liu, S., Wang, T., Zhu, J.-Y., & Torralba, A. (2020). Rewriting a Deep Generative Model. *Proceedings of the European Conference on Computer Vision (ECCV)*, 351–369.

Bauer, M., & Mnih, A. (2019). Resampled Priors for Variational Autoencoders. *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, 66–75.

Bautista, M. A., Guo, P., Abnar, S., Talbott, W., Toshev, A., Chen, Z., Dinh, L., Zhai, S., Goh, H., Ulbricht, D., Dehghan, A., & Susskind, J. (2022). GAUDI: A Neural Architect for Immersive 3D Scene Generation. In S.

Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho & A. Oh (Eds.), *Advances in neural information processing systems* (pp. 25102–25116, Vol. 35).

Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, *57* (1), 289–300.

Bentley, G. R., & Osborn, J. C. (2019). The Videogame Affordances Corpus. *AIIDE Workshop on Experimental AI in Games (EXAG)*.

Berlyne, D. E. (1960). *Conflict, Arousal and Curiosity*. McGraw-Hill.

Berns, S., Broad, T., Guckelsberger, C., & Colton, S. (2021). Automating Generative Deep Learning for Artistic Purposes: Challenges and Opportunities. *Proceedings of the 12th International Conference on Computational Creativity (ICCC)*.

Berns, S., & Colton, S. (2020). Bridging Generative Deep Learning and Computational Creativity. *Proceedings of the 11th International Conference on Computational Creativity (ICCC)*.

Berns, S., Colton, S., & Guckelsberger, C. (2023). Towards Mode Balancing of Generative Models via Diversity Weights. *Proceedings of the 14th International Conference on Computational Creativity (ICCC)*.

Berns, S., Volz, V., Tokarchuk, L., Snodgrass, S., & Guckelsberger, C. (2024). Not All the Same: Understanding and Informing Similarity Estimation in Tile-Based Video Games. *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*.

Bhaumik, D., Togelius, J., Yannakakis, G. N., & Khalifa, A. (2023). Lode Enhancer: Level Co-creation Through Scaling. *Proceedings of the 18th International Conference on the Foundations of Digital Games (FDG)*.

Bisong, E. (2019). Google Colaboratory. In *Building machine learning and deep learning models on google cloud platform*. Springer.

Boden, M. A. (2004). *The Creative Mind: Myths and Mechanisms* (Second). Routledge.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man Is to Computer Programmer as Woman Is to Homemaker? Debiasing Word Embeddings. *Advances in Neural Information Processing Systems*, *29*.

Bontrager, P., Roy, A., Togelius, J., Memon, N., & Ross, A. (2018). Deepmasterprints: Generating Masterprints for Dictionary Attacks via Latent Variable Evolution. *Proceedings of the International Conference on Biometrics Theory, Applications and Systems (BTAS)*.

Bradner, E., Iorio, F., & Davis, M. (2014). Parameters Tell the Design Story: Ideation and Abstraction in Design Optimization. *Proceedings of the Symposium on Simulation for Architecture & Urban Design*.

Broad, T., Berns, S., Colton, S., & Grierson, M. (2021). Active Divergence with Generative Deep Learning - A Survey and Taxonomy. *Proceedings of the 12th International Conference on Computational Creativity (ICCC)*.

Broad, T., & Grierson, M. (2017). Autoencoding Blade Runner: Reconstructing Films with Artificial Neural Networks. *Leonardo*, *50*(4).

Broad, T., Leymarie, F. F., & Grierson, M. (2020). Amplifying The Uncanny. *8th Conference on Computation, Communication, Aesthetics & X (xCoAx 2020)*.

Broad, T., Leymarie, F. F., & Grierson, M. (2021). Network Bending: Expressive Manipulation of Deep Generative Models. *Proceedings of the International Conference on Computational Intelligence in Music, Sound, Art and Design (EvoMusArt)*.

Brock, A., Donahue, J., & Simonyan, K. (2019). Large Scale GAN Training for High Fidelity Natural Image Synthesis. *International Conference on Learning Representations (ICLR)*.

Buolamwini, J., & Gebru, T. (2018). Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., & Lerchner, A. (2017). Understanding Disentangling in Beta-VAE. *NIPS Workshop on Learning Disentangled Representations*.

Canossa, A., & Smith, G. (2015). Towards a Procedural Evaluation Technique: Metrics for Level Design. *The 10th International Conference on the Foundations of Digital Games*, 8.

Carroll, M., Shah, R., Ho, M. K., Griffiths, T., Seshia, S., Abbeel, P., & Dragan, A. (2019). On the Utility of Learning about Humans for Human-AI Coordination. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 32). Curran Associates, Inc.

Catmull, E., & Rom, R. (1974). A Class of Local Interpolating Splines. In *Computer aided geometric design*.

Chang, A., Fontaine, M., Booth, S., Matarić, M. J., & Nikolaidis, S. (2023). Quality-Diversity Generative Sampling for Learning with Synthetic Data. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Charnley, J. W., Colton, S., & Llano, M. T. (2014). The FloWr Framework: Automated Flowchart Construction, Optimisation and Alteration for Creative Systems. *Proceedings of the Fifth International Conference on Computational Creativity (ICCC)*.

Chauhan, K., Jani, S., Thakkar, D., Dave, R., Bhatia, J., Tanwar, S., & Obaidat, M. S. (2020). Automated Machine Learning: The New Wave of Machine Learning. *Proceedings of the International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*.

Che, T., Li, Y., Jacob, A. P., Bengio, Y., & Li, W. (2017). Mode Regularized Generative Adversarial Networks. *International Conference on Learning Representations (ICLR)*.

Collins, E., Bala, R., Price, B., & Susstrunk, S. (2020). Editing in Style: Uncovering the Local Semantics of GANs. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5771–5780.

Colton, S. (2008a). Automatic Invention of Fitness Functions with Application to Scene Generation. *Workshops on Applications of Evolutionary Computation*.

Colton, S. (2008b). Creativity Versus the Perception of Creativity in Computational Systems. *Proceedings of the AAAI Spring Symposium: Creative Intelligent Systems*.

Colton, S. (2009). Seven Catchy Phrases for Computational Creativity Research. *Dagstuhl Seminar Proceedings*.

Colton, S. (2022). Towards Educating Artificial Neural Systems. *Proceedings of the International Workshop on Neuro-Symbolic Learning and Reasoning*.

Colton, S., Charnley, J. W., & Pease, A. (2011). Computational Creativity Theory: The FACE and IDEA Descriptive Models. *Proc. ICCC*.

Colton, S., Smith, A., Berns, S., Murdock, R., & Cook, M. (2021). Generative Search Engines: Initial Experiments. *Proceedings of the 12th International Conference on Computational Creativity (ICCC)*.

Colton, S., & Wiggins, G. A. (2012). Computational Creativity: The Final Frontier? *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*.

Compton, K., & Mateas, M. (2015). Casual Creators. *Proceedings of International Conference on Computational Creativity*, 228–235.

Cook, M., & Colton, S. (2018). Neighbouring Communities: Interaction, Lessons and Opportunities. *Proceedings of the Ninth International Conference on Computational Creativity (ICCC)*, 256–263.

Cook, M., Gow, J., Smith, G., & Colton, S. (2021). Danesh: Interactive Tools for Understanding Procedural Content Generators. *IEEE Transactions on Games*, *14*(3), 329–338.

Costa, V., Lourenço, N., Correia, J., & Machado, P. (2019). COEGAN: Evaluating the Coevolution Effect in Generative Adversarial Networks. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 374–382.

Cully, A. (2019). Autonomous Skill Discovery with Quality-Diversity and Unsupervised Descriptors. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*.

Cully, A., Clune, J., Tarapore, D., & Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature*, *521*(7553).

Cully, A., & Demiris, Y. (2018a). Hierarchical Behavioral Repertoires with Unsupervised Descriptors. *Proceedings of the Genetic and Evolutionary Computation Conference*.

Cully, A., & Demiris, Y. (2018b). Quality and Diversity Optimization: A Unifying Modular Framework. *IEEE Transactions on Evolutionary Computation*, *22*(2).

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6* (2).

Deb, K., & Saha, A. (2012). Multimodal Optimization Using a Bi-Objective Evolutionary Algorithm. *Evolutionary Computation*, *20*(1).

Demiralp, Ç., Bernstein, M. S., & Heer, J. (2014). Learning Perceptual Kernels for Visualization Design. *IEEE Transactions on Visualization and Computer Graphics*, *20*(12).

Dhariwal, P., & Nichol, A. (2021). Diffusion Models Beat GANs on Image Synthesis. *Advances in Neural Information Processing Systems*, *34*, 8780–8794.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2022). An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations*.

Dudley, J. M. (2013). Defending Basic Research. *Nature Photonics*, *7* (5), 338–339.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through Awareness. *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*.

Edwards, M., Jiang, M., & Togelius, J. (2021). Search-Based Exploration and Diagnosis of TOAD-GAN. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, *17*, 140–147.

Eggensperger, K., Garnett, R., Vanschoren, J., Lindauer, M., & Gardner, J. R. (Eds.). (2024). *Proceedings of the Third International Conference on Automated Machine Learning* (Vol. 256). PMLR.

Elgammal, A., Liu, B., Elhoseiny, M., & Mazzone, M. (2017). CAN: Creative Adversarial Networks, Generating Art by Learning About Styles and Deviating from Style Norms. *Proceedings of the Eighth International Conference on Computational Creativity (ICCC)*, 96–103.

Esser, P., Rombach, R., & Ommer, B. (2021). Taming Transformers for High-Resolution Image Synthesis. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12873–12883.

Evans, Z., Carr, C., Taylor, J., Hawley, S. H., & Pons, J. (2024). Fast Timing-Conditioned Latent Audio Diffusion. *Proceedings of the 41st International Conference on Machine Learning*.

Faust, A., Garnett, R., White, C., Hutter, F., & Gardner, J. R. (Eds.). (2023). *Proceedings of the Second International Conference on Automated Machine Learning* (Vol. 224). PMLR.

Fechner, G. T. (1860). *Elemente der Psychophysik* (Vol. 2). Breitkopf u. Härtel.

Feinman, R., & Lake, B. M. (2020). Generating New Concepts with Hybrid Neuro-Symbolic Models. *Proceedings of the 42nd Annual Conference of the Cognitive Science Society*.

Fontaine, M., Hsu, Y.-C., Zhang, Y., Tjanaka, B., & Nikolaidis, S. (2021). On the Importance of Environments in Human-Robot Coordination. *Robotics: Science and Systems XVII*, *17*.

Fontaine, M., Liu, R., Khalifa, A., Modi, J., Togelius, J., Hoover, A. K., & Nikolaidis, S. (2021). Illuminating Mario Scenes in the Latent Space of a Generative Adversarial Network. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*, 5922–5930.

Friedler, S. A., Scheidegger, C., & Venkatasubramanian, S. (2016). *On the (Im)Possibility of Fairness*. arXiv: 1609.07236v1.

Friedman, D., & Dieng, A. B. (2023). The Vendi Score: A Diversity Evaluation Metric for Machine Learning. *Transactions on Machine Learning Research*.

Fu, S., Tamir, N., Sundaram, S., Chai, L., Zhang, R., Dekel, T., & Isola, P. (2023). Dreamsim: Learning New Dimensions of Human Visual Similarity using Synthetic Data. *Advances in Neural Information Processing Systems*, *36*.

Fu, Y., Chen, W., Wang, H., Li, H., Lin, Y., & Wang, Z. (2020). AutoGAN-Distiller: Searching to Compress Generative Adversarial Networks. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (pp. 3292–3303, Vol. 119). PMLR.

Gaier, A., Asteroth, A., & Mouret, J.-B. (2020). Discovering Representations for Black-box Optimization. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, *11*, 103–111.

Gao, C., Chen, Y., Liu, S., Tan, Z., & Yan, S. (2020). AdversarialNAS: Adversarial Neural Architecture Search for GANs. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z., & Fidler, S. (2022). GET3D: A Generative Model of High Quality 3D Textured Shapes Learned from Images. In S. Koyejo, S. Mohamed, A. Agarwal,

D. Belgrave, K. Cho & A. Oh (Eds.), *Advances in neural information processing systems* (pp. 31841–31854, Vol. 35).

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image Style Transfer Using Convolutional Neural Networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2414–2423.

Giacomello, E., Lanzi, P. L., & Loiacono, D. (2019). Searching the Latent Space of a Generative Adversarial Network to Generate DOOM Levels. *2019 IEEE Conference on Games (CoG)*.

Glorot, X., & Bengio, Y. (2010). Understanding the Difficulty of Training Deep Feedforward Neural Networks. In Y. W. Teh & M. Titterington (Eds.), *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256, Vol. 9). PMLR.

Gong, X., Chang, S., Jiang, Y., & Wang, Z. (2019). AutoGAN: Neural Architecture Search for Generative Adversarial Networks. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Grinblat, J., & Bucklew, C. B. (2010). Caves of Qud.

Grover, A., Song, J., Kapoor, A., Tran, K., Agarwal, A., Horvitz, E. J., & Ermon, S. (2019). Bias Correction of Learned Generative Models using Likelihood-Free Importance Weighting. *Advances in Neural Information Processing Systems*, *32*.

Guckelsberger, C., Salge, C., & Colton, S. (2017). Addressing the "Why?" in Computational Creativity: A Non-Anthropocentric, Minimal Model of Intentional Creative Agency. *Proceedings of the Eighth International Conference on Computational Creativity (ICCC)*.

Guckelsberger, C., Salge, C., Gow, J., & Cairns, P. (2017). Predicting Player Experience without the Player. An Exploratory Study. *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, 305–315.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved Training of Wasserstein GANs. *Advances in Neural Information Processing Systems*, *30*.

Guyon, I., Lindauer, M., Schaar, M., Hutter, F., & Garnett, R. (Eds.). (2022). *Proceedings of the First International Conference on Automated Machine Learning* (Vol. 188). PMLR.

Guyon, I., Sun-Hosoya, L., Boullé, M., Escalante, H. J., Escalera, S., Liu, Z., Jajetic, D., Ray, B., Saeed, M., Sebag, M., Statnikov, A., Tu, W., & Viegas, E. (2019). Analysis of the AutoML Challenge series 2015-2018. *AutoML*.

Guzdial, M., & Riedl, M. O. (2019). Combinets: Creativity via Recombination of Neural Networks. *Proceedings of the Tenth International Conference on Computational Creativity (ICCC)*.

Hagg, A. (2021). Phenotypic Niching Using Quality Diversity Algorithms. In M. Preuss, M. G. Epitropakis, X. Li & J. E. Fieldsend (Eds.), *Metaheuristics for finding multiple solutions* (pp. 287–315). Springer International Publishing.

Hagg, A., Preuss, M., Asteroth, A., & Bäck, T. (2020). An Analysis of Phenotypic Diversity in Multi-Solution Optimization. *International Conference on Bioinspired Methods and Their Applications (BIOMA)*, 43–55.

Hagg, A., Wilde, D., Asteroth, A., & Bäck, T. (2020). Designing Air Flow with Surrogate-assisted Phenotypic Niching. *International Conference on Parallel Problem Solving from Nature (PPSN)*, 140–153.

Hardt, M., Price, E., Price, E., & Srebro, N. (2016). Equality of Opportunity in Supervised Learning. *Advances in Neural Information Processing Systems*, *29*.

Härkönen, E., Hertzmann, A., Lehtinen, J., & Paris, S. (2020). GANSpace: Discovering Interpretable GAN Controls. *Advances in Neural Information Processing Systems*, *33*.

He, X., Zhao, K., & Chu, X. (2021). AutoML: A Survey of the State-of-the-Art. *Knowledge-Based Systems*, *212*.

Heckman, J. J. (1979). Sample Selection Bias as a Specification Error. *Econometrica*, *47* (1).

Hendricks, L. A., Burns, K., Saenko, K., Darrell, T., & Rohrbach, A. (2018). Women Also Snowboard: Overcoming Bias in Captioning Models. *Proceedings of ECCV*.

Hertzmann, A. (2019). Visual Indeterminacy in Generative Neural Art. *2019 NeurIPS Workshop on Creativity for Learning and Design*.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *Advances in Neural Information Processing Systems*, *30*.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., & Lerchner, A. (2016). Beta-VAE: Learning Basic Visual Concepts With a Constrained Variational Framework. *International Conference on Learning Representations (ICLR)*.

Hill, M. O. (1973). Diversity and Evenness: A Unifying Notation and Its Consequences. *Ecology*, *54* (2), 427–432.

Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). *Automated Machine Learning: Methods, Systems, Challenges*. Springer Nature.

Japkowicz, N., & Stephen, S. (2002). The Class Imbalance Problem: A Systematic Study. *Intelligent Data Analysis*, *6* (5).

Jennings, K. E. (2010). Developing Creativity: Artificial Barriers in Artificial Intelligence. *Minds and Machines*, *20* (4).

Jordanous, A. (2013). *Evaluating Computational Creativity: A Standardised Procedure for Evaluating Creative Systems and Its Application* [PhD Thesis]. University of Sussex.

Jordanous, A. (2016). Four PPPPerspectives on Computational Creativity in Theory and in Practice. *Connection Science*, *28*(2).

Jordanous, A., & Keller, B. (2016). Modelling Creativity: Identifying Key Components through a Corpus-Based Approach. *PloS one*.

Kandpal, N., Wallace, E., & Raffel, C. (2022). Deduplicating Training Data Mitigates Privacy Risks in Language Models. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu & S. Sabato (Eds.), *Proceedings of the 39th International Conference on Machine Learning* (pp. 10697–10707, Vol. 162). PMLR.

Kantosalo, A., Toivanen, J. M., Xiao, P., & Toivonen, H. (2014). From Isolation to Involvement: Adapting Machine Creativity Software to Support Human-Computer Co-Creation. *Proceedings of the Fifth International Conference on Computational Creativity (ICCC)*.

Kantosalo, A., & Toivonen, H. (2016). Modes for Creative Human-Computer Collaboration: Alternating and Task-Divided Co-Creativity. *Proceedings of the Seventh International Conference on Computational Creativity (ICCC)*.

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). Progressive Growing of GANs for Improved Quality, Stability, and Variation. *International Conference on Learning Representations (ICLR)*.

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., & Aila, T. (2020). Training Generative Adversarial Networks with Limited Data. *Advances in Neural Information Processing Systems*, *33*.

Karras, T., Laine, S., & Aila, T. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4401–4410.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and Improving the Image Quality of StyleGAN. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8110–8119.

Khosla, A., Zhou, T., Malisiewicz, T., Efros, A. A., & Torralba, A. (2012). Undoing the Damage of Dataset Bias. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato & C. Schmid (Eds.), *Proceedings of ECCV*.

King, T., Butcher, S., & Zalewski, L. (2017, March). *Apocrita - High Performance Computing Cluster for Queen Mary University of London*. https://doi.org/10.5281/zenodo.438045

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations (ICLR)*.

Kingma, D. P., & Welling, M. (2014). Auto-Encoding Variational Bayes. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Ko, H.-K., Park, G., Jeon, H., Jo, J., Kim, J., & Seo, J. (2023). Large-scale Text-to-Image Generation Models for Visual Artists' Creative Works. *International Conference on Intelligent User Interfaces*.

Kohonen, T. (1988). *Self-Organization and Associative Memory*. Springer Berlin Heidelberg.

Kynkäänniemi, T., Karras, T., Aittala, M., Aila, T., & Lehtinen, J. (2023). The Role of ImageNet Classes in Fréchet Inception Distance. *International Conference on Learning Representations (ICLR)*.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., & Aila, T. (2019). Improved Precision and Recall Metric for Assessing Generative Models. *Advances in Neural Information Processing Systems*, *32*, 3929–3938.

Lagunas, M., Malpica, S., Serrano, A., Garces, E., Gutierrez, D., & Masia, B. (2019). A Similarity Measure for Material Appearance. *ACM Trans. Graph.*, *38*(4).

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, *86* (11), 2278–2324.

LeCun, Y., Cortes, C., & Burges, C. J. (2010). *The MNIST Database of Handwritten Digits*. Speech and Image Processing Services Research Laboratory, AT&T Labs-Research. http://yann.lecun.com/exdb/mnist/

Lee, J., Hong, Y., Kim, H., & Chung, H. W. (2021). Self-Diagnosing GAN: Diagnosing Underrepresented Samples in Generative Adversarial Networks. *Advances in Neural Information Processing Systems*, *34*.

Lehman, J., Meyerson, E., El-Gaaly, T., Stanley, K. O., & Ziyaee, T. (2025). *Evolution and The Knightian Blindspot of Machine Learning*. arXiv: 2501 . 13075v1.

Lehman, J., & Stanley, K. O. (2011). Evolving a Diversity of Virtual Creatures Through Novelty Search and Local Competition. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*.

Leinster, T., & Cobbold, C. A. (2012). Measuring Diversity: The Importance of Species Similarity. *Ecology*, *93*(3), 477–489.

Li, L., Li, H., Zheng, X., Wu, J., Xiao, X., Wang, R., Zheng, M., Pan, X., Chao, F., & Ji, R. (2023). AutoDiffusion: Training-Free Optimization of Time Steps and Architectures for Automated Diffusion Model Acceleration. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 7105–7114.

Li, M., Chen, X., Li, X., Ma, B., & Vitanyi, P. (2004). The Similarity Metric. *IEEE Transactions on Information Theory*, *50*(12).

Li, M., Lin, J., Ding, Y., Liu, Z., Zhu, J.-Y., & Han, S. (2020). GAN Compression: Efficient Architectures for Interactive Conditional GANs. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liapis, A., Yannakakis, G. N., & Togelius, J. (2014). Computational Game Creativity. *Proceedings of the Fifth International Conference on Computational Creativity (ICCC)*.

Linkola, S., Kantosalo, A., Männistö, T., & Toivonen, H. (2017). Aspects of Self-Awareness: An Anatomy of Metacreative Systems. *Proceedings of the Eighth International Conference on Computational Creativity (ICCC)*.

Llano, M. T., Colton, S., Hepworth, R., & Gow, J. (2016). Automated Fictional Ideation via Knowledge Base Manipulation. *Cognitive Computation*, *8*(2).

Loaiza-Ganem, G., & Cunningham, J. P. (2019). The continuous Bernoulli: Fixing a pervasive error in variational autoencoders. *Advances in Neural Information Processing Systems*, *32*.

Lohia, P. K., Natesan Ramamurthy, K., Bhide, M., Saha, D., Varshney, K. R., & Puri, R. (2019). Bias Mitigation Post-Processing for Individual and Group Fairness. *IEEE International Conference on Acoustics, Speech and Signal Processing*.

Loughran, R. (2022). Bias and Creativity. *Proceedings of the 13th International Conference on Computational Creativity (ICCC)*.

Loughran, R., & O'Neill, M. (2017). Application Domains Considered in Computational Creativity. *Proceedings of the Eighth International Conference on Computational Creativity (ICCC)*.

Lucas, S. M., & Volz, V. (2019). Tile Pattern KL-divergence for Analysing and Evolving Game Levels. *Proceedings of the Genetic and Evolutionary Computation Conference*, 170–178.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., & Bousquet, O. (2018). Are GANs Created Equal? A Large-Scale Study. *Advances in Neural Information Processing Systems*, 700–709.

Lux, F., Meyer, S., Behringer, L., Zalkow, F., Do, P., Coler, M., Habets, E. A. P., & Vu, N. T. (2024). Meta Learning Text-to-Speech Synthesis in over 7000 Languages. *Interspeech*.

Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos, J. L., Xiong, C., Sun, Z. Z., Socher, R., Fraser, J. S., & Naik, N. (2023). Large Language Models Generate Functional Protein Sequences across Diverse Families. *Nature Biotechnology*.

Maiberg, E. (2016). 'No Man's Sky' Is Like 18 Quintillion Bowls of Oatmeal [[Online; accessed 27-November-2023]].

Mariño, J., Reis, W., & Lelis, L. (2015). An Empirical Evaluation of Evaluation Metrics of Procedurally Generated Mario Levels. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, *11*(1), 44–50.

McCormack, J., Gifford, T., & Hutchings, P. (2019). Autonomy, Authenticity, Authorship and Intention in Computer Generated Art. *Proceedings of the International Conference on Computational Intelligence in Music, Sound, Art and Design (EvoMusArt)*, 35–50.

Meinl, T., Ostermann, C., & Berthold, M. R. (2011). Maximum-Score Diversity Selection for Early Drug Discovery. *Journal of Chemical Information and Modeling*, *51*(2).

Meyerson, E., Lehman, J., & Miikkulainen, R. (2016). Learning Behavior Characterizations for Novelty Search. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 149–156.

Minasny, B., & McBratney, A. B. (2006). A Conditioned Latin Hypercube Method for Sampling in the Presence of Ancillary Information. *Computers & Geosciences*, *32*(9).

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral Normalization for Generative Adversarial Networks. *International Conference on Learning Representations (ICLR)*.

Mouret, J.-B. (2011). Novelty-Based Multiobjectivization. In S. Doncieux, N. Bredèche & J.-B. Mouret (Eds.), *New Horizons in Evolutionary Robotics*.

Mouret, J.-B., & Clune, J. (2015). *Illuminating Search Spaces by Mapping Elites*. arXiv: 1504.04909v1.

Murdock, R. (2021). The Big Sleep colab notebook [Accessed: 2021-04-01].

OpenAI. (2024). Video generation models as world simulators [Accessed on 19 June 2024].

Parmar, G., Zhang, R., & Zhu, J.-Y. (2022). On Aliased Resizing and Surprising Subtleties in GAN Evaluation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pasarkar, A. P., & Dieng, A. B. (2024). Cousins Of The Vendi Score: A Family Of Similarity-Based Diversity Metrics For Science And Machine Learning. *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, 3808–3816.

Pease, A., & Colton, S. (2011). Computational Creativity Theory: Inspirations behind the FACE and the IDEA models. *Proc. ICCC*.

Pease, A., Winterstein, D., & Colton, S. (2001). Evaluating Machine Creativity. *Workshop on Creative Systems, 4th International Conference on Case Based Reasoning*.

Piovarči, M., Levin, D. I. W., Rebello, J., Chen, D., Ďurikovič, R., Pfister, H., Matusik, W., & Didyk, P. (2016). An Interaction-Aware, Perceptual Model for Non-Linear Elastic Objects. *ACM Trans. Graph.*, *35*(4).

Piovarči, M., Levin, D. I., Kaufman, D. M., & Didyk, P. (2018). Perception-aware modeling and fabrication of digital drawing tools. *ACM Transactions on Graphics (TOG)*, *37*(4), 1–15.

Pontius, R. G., & Millones, M. (2011). Death to Kappa: Birth of Quantity Disagreement and Allocation Disagreement for Accuracy Assessment. *International Journal of Remote Sensing*, *32*(15).

Preuss, M., & Wessing, S. (2013). Measuring Multimodal Optimization Solution Sets with a View to Multiobjective Techniques. In M. Emmerich, A. Deutz, O. Schuetze, T. Bäck, E. Tantar, A.-A. Tantar, P. D. Moral, P. Legrand, P. Bouvry & C. A. Coello (Eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV* (Vol. 227).

Pugh, J. K., Soros, L. B., Szerlip, P. A., & Stanley, K. O. (2015). Confronting the Challenge of Quality Diversity. *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*.

Pugh, J. K., Soros, L. B., & Stanley, K. O. (2016). Quality Diversity: A New Frontier for Evolutionary Computation. *Frontiers in Robotics and AI*, *3*.

Rabii, Y., & Cook, M. (2023). Why Oatmeal is Cheap: Kolmogorov Complexity and Procedural Generation. *Proceedings of the 18th International Conference on the Foundations of Digital Games*, 1–7.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th International Conference on Machine Learning* (pp. 8748–8763, Vol. 139). PMLR.

Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *International Conference on Learning Representations*.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021). Zero-Shot Text-to-Image Generation. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th International Conference on Machine Learning* (pp. 8821–8831, Vol. 139). PMLR.

Razavi, A., van den Oord, A., & Vinyals, O. (2019). Generating Diverse High-Fidelity Images with VQ-VAE-2. *Advances in Neural Information Processing Systems*, *32*, 14837–14847.

Razeghi, Y., Logan IV, R. L., Gardner, M., & Singh, S. (2022). *Impact of Pretraining Term Frequencies on Few-Shot Reasoning*. arXiv: 2202.07206v2.

Rhodes, M. (1961). An Analysis of Creativity. *The Phi Delta Kappan*, *42*(7), 305–310.

Ridler, A. (2017). Repeating and Mistranslating: the Associations of GANs in an Art Context. *NeurIPS Workshop on Machine Learning for Creativity and Design*.

Ritchie, G. (2007). Some Empirical Criteria for Attributing Creativity to a Computer Program. *Minds and Machines*, *17* (1).

Rodriguez Torrado, R., Khalifa, A., Cerny Green, M., Justesen, N., Risi, S., & Togelius, J. (2020). Bootstrapping Conditional GANs for Video Game Level Generation. *2020 IEEE Conference on Games (CoG)*, 41–48.

Rogowitz, B. E., Frese, T., Smith, J. R., Bouman, C. A., & Kalin, E. B. (1998). Perceptual image similarity experiments. *Human Vision and Electronic Imaging III*, *3299*, 576–590.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-Resolution Image Synthesis With Latent Diffusion Models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

Runco, M. A., & Jaeger, G. J. (2012). The Standard Definition of Creativity. *Creativity Research Journal*, *24* (1).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, *115* (3).

Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th). Prentice Hall Press.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Gontijo-Lopes, R., Ayan, B. K., Salimans, T., Ho, J., Fleet, D. J., & Norouzi, M. (2022). Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *Advances in Neural Information Processing Systems*, *35*.

Sajjadi, M. S., Bachem, O., Lucic, M., Bousquet, O., & Gelly, S. (2018). Assessing Generative Models via Precision and Recall. *Advances in Neural Information Processing Systems*, *31*, 5228–5237.

Salakhutdinov, R., & Hinton, G. (2009). Deep Boltzmann Machines. *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, 448–455.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved Techniques for Training GANs. *Advances in Neural Information Processing Systems*, *29*.

Sarkar, A., & Cooper, S. (2022). Tile2tile: Learning Game Filters for Platformer Style Transfer. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, *18* (1), 53–60.

Saunders, R. (2012). Towards Autonomous Creative Systems: A Computational Approach. *Cognitive Computation*, *4*(3), 216–225.

Saunders, R., Gemeinboeck, P., Lombard, A., Bourke, D., & Kocaballi, A. B. (2010). Curious Whispers: An Embodied Artificial Creative System. *Proceedings of the First International Conference on Computational Creativity (ICCC)*.

Saunders, R., & Gero, J. S. (2004). Curious agents and situated design evaluations. *AI EDAM*, *18*(2), 153–161.

Schultz, D. (2020). Make ML Art Datasets: Week 3 [Accessed: 2020-03-29].

Shaker, N., Nicolau, M., Yannakakis, G. N., Togelius, J., & O'Neill, M. (2012). Evolving Levels for Super Mario Bros Using Grammatical Evolution. *2012 IEEE Conference on Computational Intelligence and Games (CIG)*.

Shane, J. (2018). Machine learning failures - for art! [Accessed: 2020-03-29].

Shi, W., Wang, Z., Soler, C., & Rushmeier, H. (2021). A Low-Dimensional Perceptual Space for Intuitive BRDF Editing. *EGSR 2021-Eurographics Symposium on Rendering-DL-only Track*, 1–13.

Shimodaira, H. (2000). Improving Predictive Inference under Covariate Shift by Weighting the Log-Likelihood Function. *Journal of Statistical Planning and Inference*, *90*(2).

Shin, J.-E., Riesselman, A. J., Kollasch, A. W., McMahon, C., Simon, E., Sander, C., Manglik, A., Kruse, A. C., & Marks, D. S. (2021). Protein Design and Variant Prediction Using Autoregressive Generative Models. *Nature Communications*, *12*(1).

Shneiderman, B. (2002). Creativity Support Tools. *Communications of the ACM*, *45*(10), 116–120.

Sim, J., & Wright, C. C. (2005). The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements. *Physical Therapy*, *85*(3).

Simon, H. A. (1990). Bounded rationality. In *Utility and probability* (pp. 15–18). Springer.

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICLR)*.

Simpson, E. H. (1949). Measurement of Diversity. *Nature*, *163*(4148), 688–688. https://doi.org/10.1038/163688a0

Smith, G. (2017). Computational Creativity and Social Justice: Defining the Intellectual Landscape. *Proceedings of the Workshop on Computational Creativity and Social Justice at the Eighth International Conference on Computational Creativity (ICCC)*.

Smith, G., & Whitehead, J. (2010). Analyzing the Expressive Range of a Level Generator. *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*.

Sobol, I. M. (1967). On the Distribution of Points in a Cube and the Approximate Evaluation of Integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, *7* (4).

Solow, A. R., & Polasky, S. (1994). Measuring Biological Diversity. *Environmental and Ecological Statistics*, *1* (2).

Stålberg, O., Meredith, R., & Kvale, M. (2018). Bad North.

Stanley, K. O., & Lehman, J. (2015). *Why Greatness Cannot Be Planned: The Myth of the Objective*. Springer.

Stock, P., & Cisse, M. (2018). ConvNets and ImageNet Beyond Accuracy: Understanding Mistakes and Uncovering Biases. *Proceedings of ECCV*.

Stokes, D. E. (2011). *Pasteur's quadrant: Basic science and technological innovation*. Brookings Institution Press.

Summerville, A. (2018). Expanding Expressive Range: Evaluation Methodologies for Procedural Content Generation. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, *14* (1), 116–122.

Summerville, A., Mariño, J. R. H., Snodgrass, S., Ontañón, S., & Lelis, L. H. S. (2017). Understanding Mario: An Evaluation of Design Metrics for Platformers. *Proceedings of the 12th International Conference on the Foundations of Digital Games*.

Summerville, A., Snodgrass, S., Guzdial, M., Holmgård, C., Hoover, A. K., Isaksen, A., Nealen, A., & Togelius, J. (2018). Procedural Content Generation via Machine Learning (PCGML). *IEEE Transactions on Games*, *10* (3), 257–270.

Summerville, A., Snodgrass, S., Mateas, M., & Ontañón, S. (2016). The VGLC: The Video Game Level Corpus. *Proceedings of the 7th Workshop on Procedural Content Generation*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Terveen, L. G. (1995). Overview of Human-Computer Collaboration. *Knowledge-Based Systems*, *8* (2-3).

Todd, G., Earle, S., Nasir, M. U., Green, M. C., & Togelius, J. (2023). Level Generation Through Large Language Models. *Proceedings of the 18th International Conference on the Foundations of Digital Games*.

Tommasi, T., Patricia, N., Caputo, B., & Tuytelaars, T. (2017). A Deeper Look at Dataset Bias. In G. Csurka (Ed.), *Domain Adaptation in Computer Vision Applications*. Springer International Publishing.

Torralba, A., & Efros, A. A. (2011). Unbiased Look at Dataset Bias. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Truong, A., Walters, A., Goodsitt, J., Hines, K., Bruss, C. B., & Farivar, R. (2019). Towards Automated Machine Learning: Evaluation and Comparison of AutoML Approaches and Tools. *Proceedings of the International Conference on Tools with Artificial Intelligence (ICTAI)*.

Tuggener, L., Amirian, M., Rombach, K., Lörwald, S., Varlet, A., Westermann, C., & Stadelmann, T. (2019). Automated Machine Learning in Practice: State of the Art and Recent Results. *Proceedings of the Swiss Conference on Data Science*.

Ulrich, T., Bader, J., & Thiele, L. (2010). Defining and Optimizing Indicator-Based Diversity Measures in Multiobjective Search. In R. Schaefer, C. Cotta, J. Kołodziej & G. Rudolph (Eds.), *Parallel Problem Solving from Nature, PPSN XI*.

van der Maaten, L., & Weinberger, K. (2012). Stochastic Triplet Embedding. *2012 IEEE International Workshop on Machine Learning for Signal Processing*.

van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*(86), 2579–2605.

Vassiliades, V., Chatzilygeroudis, K., & Mouret, J.-B. (2017). Using Centroidal Voronoi Tessellations to Scale Up the Multidimensional Archive of Phenotypic Elites Algorithm. *IEEE Transactions on Evolutionary Computation*, *22*(4).

Veale, T. (2013). Creativity as a Web Service: A Vision of Human and Computer Creativity in the Web Era. *AAAI Spring Symposium Series*.

Veale, T., Cardoso, F. A., & Pérez y Pérez, R. (2019). Systematizing Creativity: A Computational View. *Computational Creativity: The Philosophy and Engineering of Autonomously Creative Systems*, 1–19.

Ventura, D. (2016). Mere Generation: Essential Barometer or Dated Concept? *Proceedings of the Seventh International Conference on Computational Creativity (ICCC)*.

Villani, C., et al. (2009). *Optimal Transport: Old and New* (Vol. 338). Springer.

Vimpari, V., Kultima, A., Hämäläinen, P., & Guckelsberger, C. (2023). "An Adapt-or-Die Type of Situation": Perception, Adoption, and Use of Text-To-Image-Generation AI by Game Industry Professionals. *Proceedings of the ACM Annual Symposium on Computer-Human Interaction in Play*.

Volz, V. (2019). *Uncertainty Handling in Surrogate Assisted Optimisation of Games* [Doctoral dissertation, Technische Universität Dortmund].

Volz, V., Justesen, N., Snodgrass, S., Asadi, S., Purmonen, S., Holmgård, C., Togelius, J., & Risi, S. (2020). Capturing Local and Global Patterns in Procedural Content Generation via Machine Learning. *2020 IEEE Conference on Games (CoG)*.

Volz, V., Schrum, J., Liu, J., Lucas, S. M., Smith, A., & Risi, S. (2018). Evolving Mario Levels in the Latent Space of a Deep Convolutional Generative Adversarial Network. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 221–228.

Vornhagen, J. B., Tyack, A., & Mekler, E. D. (2020). Statistical Significance Testing at CHI PLAY: Challenges and Opportunities for More Transparency. *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, 4–18.

Wang, C., Xu, C., Yao, X., & Tao, D. (2019). Evolutionary Generative Adversarial Networks. *IEEE Transactions on Evolutionary Computation*, *23*(6).

Wang, H., Jin, Y., & Yao, X. (2017). Diversity Assessment in Many-Objective Optimization. *IEEE Transactions on Cybernetics*, *47*(6).

Weitzman, M. L. (1992). On Diversity. *The Quarterly Journal of Economics*, *107*(2), 363–405.

Wiggins, G. A. (2006a). A Preliminary Framework for Description, Analysis and Comparison of Creative Systems. *Knowledge-Based Systems*, *19*(7), 449–458. https://doi.org/10.1016/j.knosys.2006.04.009

Wiggins, G. A. (2006b). Searching for Computational Creativity. *New Generation Computing*, *24*(3).

Wills, J., Agarwal, S., Kriegman, D., & Belongie, S. (2009). Toward a perceptual space for gloss. *ACM Transactions on graphics (TOG)*, *28*(4), 1–15.

Withington, O., & Tokarchuk, L. (2023). The Right Variety: Improving Expressive Range Analysis with Metric Selection Methods. *Proceedings of the 18th International Conference on the Foundations of Digital Games*.

Wong, B. (2011). Points of View: Color Blindness. *Nature Methods*, *8*(6).

Yannakakis, G. N., & Togelius, J. (2011). Experience-Driven Procedural Content Generation. *IEEE Transactions on Affective Computing*, *2*(3), 147–161.

Yu, N., Li, K., Zhou, P., Malik, J., Davis, L., & Fritz, M. (2020). Inclusive GAN: Improving Data and Minority Coverage in Generative Models. In A. Vedaldi, H. Bischof, T. Brox & J.-M. Frahm (Eds.), *Proceedings of ECCV*. Springer International Publishing.

Zadrozny, B. (2004). Learning and Evaluating Classifiers under Sample Selection Bias. In R. Greiner & D. Schuurmans (Eds.), *Proceedings of the 21st International Conference on Machine Learning*. ACM Press.

Zafar, M. B., Valera, I., Gomez Rodriguez, M., & Gummadi, K. P. (2017). Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. *Proceedings of the 26th International Conference on World Wide Web*.

Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-Attention Generative Adversarial Networks. *Proceedings of the 36th International Conference on Machine Learning*, 7354–7363.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 586–595.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K.-W. (2017). Men Also Like Shopping: Reducing Gender Bias Amplification Using Corpus-Level Constraints. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Zhong, P., Mo, Y., Xiao, C., Chen, P., & Zheng, C. (2019). Rethinking Generative Mode Coverage: A Pointwise Guaranteed Approach. *Advances in Neural Information Processing Systems*, *32*.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Zitzler, E., Knowles, J., & Thiele, L. (2008). Quality Assessment of Pareto Set Approximations. In J. Branke, K. Deb, K. Miettinen & R. Słowiński (Eds.), *Multiobjective Optimization: Interactive and Evolutionary Approaches*.

## COLOPHON

This thesis was designed by Sebastian Berns and is typeset in *Signifier* by Klim Type Foundry. The design is based on the *classicthesis* template developed by André Miede and Ivo Pletikosić.

https://sebastianberns.com

*Final Version* as of 18th March 2025 (Public).