Latent spaces: a creative approach

Matthew Yee-King

Abstract This chapter explores the creative possibilities offered by latent spaces. Latent spaces are machine-learnt maps representing large media datasets such as images and sound. With a latent space, an artist can rapidly search for interesting places in the dataset, then generate new artefacts around and between places. These unique artefacts were not in the original dataset, but they relate to it. Readers will find a detailed explanation of what latent spaces are and how they fit into a series of developments that have taken place in digital media processing techniques such as content-based search and feature extraction. We will encounter four examples of machine learning systems that provide latent spaces suitable for creative work. The first example is Music-VAE which creates a latent space of millions of musical fragments represented in the symbolic MIDI format. The second example is Latent Timbre Synthesis (LTS). Unlike Music-VAE, which works in a symbolic musical domain, LTS works directly with audio fragments. The third example is StyleGAN which creates a latent space of images which has specific properties allowing for style transfers. The final example is VQGAN+CLIP which is a text phrase-to-image system which uses fine-tuning techniques to iteratively generate images. Finally, we consider examples of artists working with each of the four systems along with reflections on their creative processes.

1 Introduction

One might characterise the practice of creating visual art, music or sound as an exploratory process. An artist explores the space of possibilities in a given domain. Over time, they develop an understanding of this space and how they can effectively explore it and present it to others.

Matthew Yee-King
Department of Computing, Goldsmiths, University of London, e-mail: m.yee-king@gold.ac.uk

This chapter will consider how such a creative process might change if the artist has access to a latent space. We will dig into a deeper explanation of latent spaces later on, but for now, think of a latent space as a map of thousands or even millions of items. Taking the example of a latent space of images, you would start with a dataset containing thousands of images. A machine learning system would place all the images in positions relative to each other, just like a city map showing the various buildings and landmarks. It would attempt to place similar looking images in similar positions on the map. The space containing all the positions is known as the latent space. You can take an image that the map knows, and the map will tell you where that image is located.

A latent space is no ordinary map, though - you can also take an image that it has never seen, and it can give you its best guess as to where that image lies based on how it looks. You can place your finger in a random position on the map, which might lie in-between known images, and the system can generate you an image that is its best guess for what you should see there. This ability to place and create artefacts, not on the original map unlocks a wealth of creative possibilities. In this chapter, I consider the new opportunities for artistic practice presented by such technology and how creative practice changes to best exploit these possibilities.

Let us capture a flavour of the change by taking in a view from some contemporary creative practitioners who have worked extensively with latent spaces. Broad et al. describe the creation of latent spaces as 'learning to render entire distributions of complex high dimensional data with ever-increasing fidelity. They group together the various ways artists have exploited latent spaces under the term 'active divergence'. Active divergence involves 'optimising, hacking and rewriting [latent space models] to actively diverge from the training data' (Broad, Berns, et al. 2021). Referring to the example of a map of images, the training data would be the set of images used to create the original map. Optimising, hacking and rewriting are standard methods for computer artists, but how can they execute those in this machine learning-driven domain? We will consider this question through several detailed examples later on.

The ability to create and explore latent spaces can be seen as a computer-aided transition from considering a small number of positions in a creative space to having access to an interrogable model representing many positions. How can an artist cope with this abundance of choices? As it turns out, this is nothing new for artists. Going back to the early days of computer-aided creative work, we find that practitioners were quick to identify and discuss the 'over-abundance problem'. In an interview in 1968, John Cage characterised computerisation as a transition from a scarcity of ideas to an abundance:

> "The need to work as though decisions were scarce-as though you had to limit yourself to one idea-is no longer pressing. It's a change from the influences of scarcity or economy to the influences of abundance and-I'd be willing to say-waste." - Cage 1968 (Austin, Cage, and Hiller 1992)

A contemporary of Cage, Gottfried Michael Koenig goes a step further in considering what statistical distributions representing a range of possibilities might mean for composers:

> “the trouble taken by the composer with series and their permutations has been in vain; in the end it is the statistical distribution that determines the composition" - Koenig, 1971 (Ames 1987)

Another composer, Iannis Xenakis, famously also worked with statistical distributions. Writing in 1966, Xenakis saw it as a “musical necessity that the laws pertaining to the calculation of probabilities found their way into composition" (Xenakis 1966). This colourful quote embodies the feeling one might have when working with latent spaces:

> "With the aid of electronic computers the composer becomes a sort of pilot: he presses the buttons, introduces coordinates, and supervises the controls of a cosmic vessel sailing in the space of sound" Xenakis 1971 (ibid.)

These early statistical creation methods were probably a natural response to the constraints and formalisms of serialism that had come before. One might draw a parallel between the movement from serialism to stochastic composition and the rise of statistical machine learning after decades of formal and symbolic AI techniques. It is this statistical machine learning which makes creative latent spaces possible.

Reflecting on the quotes above, these early pioneers of statistical, creative spaces were already transitioning to conceptualising the substrate of their work not as small numbers of items and simple processes but large sets and complex processes. A full exploration of the history of artistic practice using statistical distributions and related techniques is beyond the scope of this writing. We have seen a to-ing and fro-ing between formalisms and statistical models in computational creative work and machine learning, with increasingly sophisticated models.

The most recent development in this story started in the mid-2000s, and it makes the rendering of large datasets into high fidelity latent spaces of images and sounds possible. LeCun et al. refer to this development as representation learning (LeCun, Bengio, and Hinton 2015). This is where a machine learning model learns how to extract and represent the most pertinent information in a dataset of images, sounds or texts. These machine-designed latent spaces and their capabilities are the subjects of this chapter.

1.1 Structure of this chapter

In the next section, I will explain what a latent space is and how we can go about creating them. In section 3, we will consider some examples of recent systems which enable artists to work creatively in latent spaces. Following that, we will look at some creative work that has been produced using the example systems we presented in section 4.

2 What is a latent space?

Now we will work towards an understanding of what a latent space is, why it is necessary and how we can go about creating one.

The avant-garde composers quoted in the previous section observed that the computerisation of creative practice is associated with a transition from a sparsity of ideas and content to an abundance. Instead of labouring with a single musical score, we work within a dataset of scores. Instead of working with a single image, we work with a dataset of thousands or millions of images. This leads us to wonder how we might adapt our practice to engage with this abundance of material.

We can start with two actions that we might want to take as creative practitioners given such a dataset: *search* and *generate*. Search allows us to find items of interest. Generate allows us to somehow generate new items using the existing items. These actions are shown in figure 1. One search approach is to use meta-data such as filenames, dates, geo-tags and camera settings. For example, you might search for all images taken in 2010. This approach depends on somebody or something having already correctly tagged the items, and the search is limited to the available tags.

Once we have located our items of interest in the dataset, we can move to the generation of new items. For example, you might load the images retrieved from your search into an image editor and create a collage. The problem is that this approach takes us back to the 'sparse' scenario as only the search part of the process takes account of the abundance in the archive. Generating like this ignores anything except the selected items.

Content-based search is another option. Content-based search involves examining the actual data in the archive. For images, the lowest level of data is the raw colour values for the pixels; for sound, it is the raw waveforms. Content-based search is potentially much more potent than meta-data search as it does not rely on accurate data being added to the items in an archive. Instead, it goes directly to the actual content in the archive.

But there are a few reasons why working with raw media data is problematic. It is very large; for example, the data for an image with a resolution of 1024x768 has one number for each red, green and blue channel in its 786,432 pixels, so it is 2,359,296 dimensional. Raw pixel data is also sensitive to image transformations, such as translations and rotations. If you rotate an image, the pixel data might change completely. So with raw pixel data, you would not be able to know if one image was

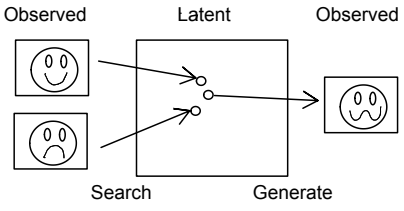

Fig. 1 Observed space contains the raw data representing the images. Latent space is a more compact space which places similar images close to each-other. A small movement in latent space should lead to a small movement in observed space.

very similar to another, just rotated. The two images would appear very far away in raw pixel space.

So why are humans able to judge two rotated images to be similar? It turns out that the raw data coming out of the back of the retina is processed through several stages before meaningful perception takes place. This processing can be called *feature extraction*, and algorithmic (as opposed to biological) feature extraction is what makes content-based search possible. To achieve content-based search, we can specify a desired feature as a search term then find items with similar features in our dataset. We might even pass a raw media item such as an image as our search term, then extract the feature for searching.

Researchers have designed many different features to enable searches for different things. For example, features that help detect which instrument is playing in a piece of music or features that help detect the position and orientation of faces in images. There are many software libraries available that make it possible to extract these features, for example, the OpenCV library for image features and the librosa library for audio features (Bradski and Kaehler 2000; McFee et al. 2015).

2.1 Latent spaces

Now we are familiar with the ideas of content-based search and feature extraction, we are ready to talk about latent spaces. The features that we use to carry out content-based search form a latent space. We can refer to the original files as the 'observable' and the features we extract as the 'latent'. Therefore, the latent space is the space that contains the features of everything in our dataset. Consider an 'average brightness' feature for images as a simple example. We can represent this feature as a single value, perhaps between 0 and 1. To search, we can specify our desired brightness range and retrieve all images falling in that range. The latent space for this feature illustrated in figure 2 is a simple, one-dimensional line along which we place all of the images in our dataset. If we move through the latent space, viewing the images as we go, we will see them increase in brightness.

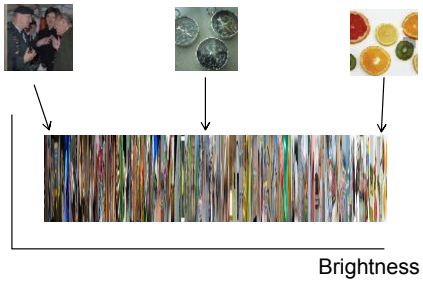

Fig. 2 Visualisation of a latent space based on a one dimensional image brightness feature. 100 images from the imagenet dataset are organised from darkest to brightest (Deng et al. 2009). Additionally, three zoomed in images are shown, which are the darkest, middle brightest and brightest.

What about generation? Operating with the content of media archives intuitively seems to lend itself to generative tasks, as we are working more directly in the domain we wish to generate. We could manually generate new content from the found content, as we did for meta-data search. But perhaps there is a more powerful option, one which maintains a sense of the abundance in the archive. If we can reverse the feature extraction process somehow, we can actually move from latent feature space back to the raw data of observed space. With such a feature reversing technique, we can actually move from any point in latent space back to observed space. This opens up possibilities such as choosing a point in the middle of two images, and generating from there. An example of that is shown in figure 1.

But reversing from latent feature space back to observed space is not so straightforward. Consider the analogy of a fruit juicing machine. You place your fruit in the machine, and it extracts the most desirable part of the fruit. But it is not easy to get back to the original fruit because you have thrown away 'information' in the juicing process. My simple image brightness feature loses a lot of information going from two million numbers down to one single number, and we could not possibly reconstruct an image from that. Unfortunately, many existing features developed for image and sound analysis tasks are not reversible. They were not designed for generative purposes, so reversing was not a requirement.

Fortunately, there is a new generation of reversible features based on machine learning. These are machine-designed features used in the representation learning systems described by LeCun et al. (LeCun, Bengio, and Hinton 2015). Instead of creating a latent space using human-engineered features, machine learning systems learn their own features suitable for representing the pertinent information in the dataset. In parallel, the systems learn methods for feature reversal. I will spend the rest of the chapter discussing these features and some AI-creativity systems built on top of them.

Before that, I will summarise the key points from this section: large datasets of media are now available; creative practitioners wanting to meaningfully exploit the abundance in these datasets need to be able to search in and generate from the dataset; content-based search provides the most powerful way to search using feature extraction; the extracted features form a latent space containing the dataset; unfortunately many features are not designed for generative purposes and they are therefore not reversible meaning you cannot generate by moving from latent space back to raw data in observed space; recent developments in machine learning provide reversible features which allow for 'fully abundant' search and generation.

3 Examples of latent spaces

Now we have a grasp of what a latent space is and what it means to search and generate, we will consider some examples of latent spaces in creative domains. I have selected four examples: two for sound and two for image. For each of image and sound, I present a symbolic approach and a sub-symbolic approach. To clarify those

terms, in the musical domain, Briot et al. define symbolic as “dealing with high-level symbolic representations (e.g., chords, harmony)" and sub-symbolic as “dealing with low-level representations (e.g., sound, timbre)", along with the associated processes for each domain (Briot, Hadjeres, and F.-D. Pachet 2020). An equivalent definition for images would have symbolic as the contents of the image (e.g. contains a dog) and sub-symbolic as describing the raw image data (RGB values, brightness).

Before we take in the examples, we should highlight some terms that we will encounter. Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) are high-level terms describing approaches to designing and training neural networks, especially for generative purposes. VAE involves training an encoder to encode to a latent space and a decoder to decode back out to observable space such that the error between original and decoded data is minimised. GANs involve a generative network learning to generate data similar to the training data where the similarity is judged by a second, ‘critic’ network which is also learning. So with GANs the error is dictated by the ever-learning critic, not a normal metric as in VAEs. Below the VAE or GAN method is the actual neural network architecture, which in both cases will consist of multiple layers of different types such as long, short term memory (LSTM), convolutional, fully connected and so on.

3.1 A latent space for symbolic music data: Music-VAE

Roberts et al. reported the Music-VAE system in 2018 (Roberts et al. 2018). Music-VAE works with MIDI files which are a standard data format for representing sequences of musical events such as notes. Music-VAE uses a Variational Autoencoder method in combination with a recurrent, LSTM network for its encoder and decoder. LSTMs are useful for time series such as sequences of musical events. Music-VAE can encode musical sequences to a latent space and it can decode from latent space back to musical sequences.

Generative music systems are not new - there is a long history of methods for generating symbolic music data. Aside from formal methods such as grammars, Markov models were a common technique in the literature prior to the dominance of deep learning. Fernandez et al. survey examples of this work (Fernández and Vico

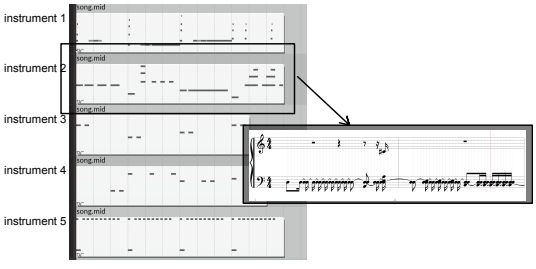

Fig. 3 Example output from a single point in the Music-VAE multi-instrument model latent space. The different instrument tracks are shown with a piano roll view, and there is a zoomed in view of one of the instruments shown with traditional notation.

2013). Pachet and Roy's work with statistical models and constraints is perhaps the pinnacle here (Roy, Papadopoulos, and F. Pachet 2017). There is limited work which deals explicitly with latent spaces of musical corpora but an example is Ellis and Arroyo's Eigenrhythms (Ellis and Arroyo 2004). Music-VAE does have some deep network precursors: the Ragtime generating DeepHear system by Sun from 2015 is perhaps the earliest example[1]. For further examples, we refer the reader to Briot et al. (Briot, Hadjeres, and F.-D. Pachet 2020).

Returning to our description of Music-VAE, using a dataset of 1.5 million MIDI files, the researchers trained different Music-VAE models on various inputs including 16 bar monophonic melody and bass lines, and drum patterns. The latent vectors therefore represent melodies, basslines or drum patterns. A later version of Music-VAE encoded complete musical arrangements, including instrument selection (Simon et al. 2018).

Music-VAE has been designed with the aim of creating a latent space which is suitable for creative exploration. The Music-VAE space places similar inputs into similar places, and it is a smooth space. Smooth means interpolations from one point to another in latent space produce a gradual series of outputs. In other words, similar melodies should encode to similar positions in latent space and moving between two melodies in latent space should sound like a musically smooth transition.

These two features are crucial to the creative possibilities of Music-VAE. To achieve the smooth transitions, Music-VAE uses a variational autoencoder as opposed to just an autoencoder. The variational autoencoder works with distributions rather than single points which encourages a better mapping of the space.

Music-VAE's creators have endeavoured to provide a range of components to help creatives work with the system. A variety of pre-trained models are available, including melody, drum, trio and full arrangement models [2]. The Magenta.js library provides ready made helper classes and functions[3]. Magenta.js allows users to sample latent space in near realtime, generate MIDI files and play them in the web browser.

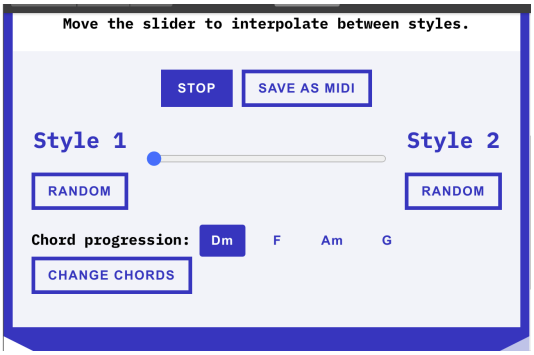

Fig. 4 The Music-VAE interpolation demo which uses the tensorflow.js implementation of Music-VAE. The user can generate two random points in latent space and interpolate between them, listening to the results. They can also condition the generative model using a chord sequence.

[1] https://fephsun.github.io/2015/09/01/neural-music.html

[2] https://goo.gl/magenta/js-checkpoints

[3] https://github.com/magenta/magenta-js

The researchers also provide a range of examples, especially based around the idea of interpolating between positions in latent space. Figure 4 shows one such demo.

3.2 A latent space for sub-symbolic audio data: Latent Timbre Synthesis

In 2021, Tatar et al. described Latent Timbre Synthesis (LTS) (Tatar, Bisig, and Pasquier 2021). LTS involves the creation of a latent space representing a corpus of audio frames. Therefore LTS is a sound synthesis technique, where a stream of latent vectors are converted to a stream of audio frames. There are several related systems which create latent spaces of audio corpora for creative purposes. Casey's 2005 work represents an early example which addressed the problem of efficient content-based search in audio documents for resynthesis purposes via the creation of a compact latent space (Casey 2005). Wavenet is a more recent and more closely related example (Oord et al. 2016). Wavenet has been through multiple iterations and is a well established structure for sound and especially voice synthesis. One of the limitations of Wavenet noted by Tatar et al. and addressed with LTS is its computational complexity, preventing realtime synthesis. Aside from Wavenet, other work has involved training networks on raw audio, for example Collins et al.'s work on 'brAIn swapping' (Collins, Ruzicka, and Grierson 2020) and Zukowski and Carr's work generating infinite death metal with SampleRNN (Zukowski and Carr 2018; Mehri et al. 2016).

We have selected the Latent Timbre Synthesis system because it explicitly deals with latent spaces and their application to interactive, creative work in the sound design and sound art domains. LTS is different from Wavenet and SampleRNN in that it generates spectral frames instead of raw audio samples, at professional CD quality as opposed to lower sample and bit rates.

To create its latent space, LTS cuts the raw audio signal into frames then applies the Constant Q Transform (CQT) to create spectrograms for each frame. CQT is a

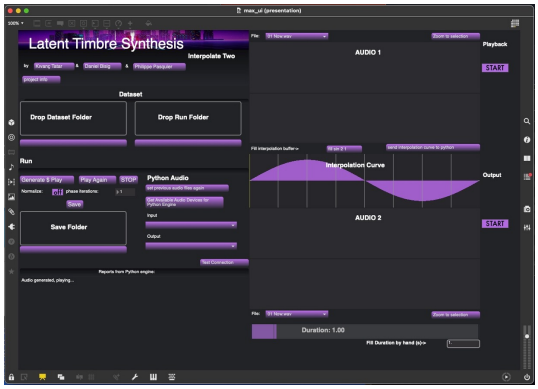

Fig. 5 User interface for the Latent Timbre Synthesis engine. The user can add a training dataset and train then they can generate audio sequences by interpolating between positions in the latent space. The interface is implemented in Max/MSP but it talks via OSC to a Python back-end.

common feature in music information retrieval tasks such as instrument detection. Next, LTS trains a variational autoencoder to encode and decode the CQT spectrograms to and from latent space. The VAE neural network architecture uses densely connected andconvolutional layers to encode and decode spectral frames.

Convolutional layers are often used for digital signal processing tasks as they are a kind of trainable filter where the training adjusts the filter so it can extract the most useful information from the signal. Convolutional layers are suited to spectral frame data if the spectral frames are not treated as a sequence. Sequential data would generally require some sort of recurrent network, such as LSTM. Since it does not use LSTM-like layers, we can say that LTS does not model the sequence in which the frames occur.

Working with latent spaces of spectral frames as opposed to raw audio samples is common amongst deep synthesis systems which aim to operate in near realtime on regular hardware. This performance level is an important feature for accessibility and interaction (Grierson et al. 2019). Tatar et al. report synthesis running at twice realtime with a commonly available consumer gaming GPU (GTX 2080). This is similar to the performance reported for other realtime tools for deep synthesis, e.g. the DDSP system (Engel et al. 2020; Yee-King and McCallum 2021)

Tatar et al. provide the Python code comprising the LTS system[4]. They also include a user interface created using Max/MSP which communicates with the Python system via Open Sound Control messages (see figure 5). With this UI, users can supply a dataset of audio files and train the system, or they can explore the latent space by providing two audio files and interpolating between them. This interpolation activity is similar to the interpolation activity in the Music-VAE system, except the output is a stream of audio frames instead of discrete MIDI arrangements.

3.3 A latent space for sub-symbolic image data: StyleGAN

Kerras et al. reported StyleGAN in 2019 (Karras, Laine, and Aila 2019). StyleGAN is an image synthesizer which is trained using generative adversarial methods. This means there are two models: the image synthesis model and a discriminator model which evaluates the synthesis model. Both models are trained at the same time - as the synthesis model improves, the discriminator model gets better at finding its flaws. StyleGAN creates a latent space of a dataset of images (70,000 in the original article). Therefore you can generate a latent vector and StyleGAN can convert that into an image, or vice-versa.

Similarly to Music-VAE, users can move around in StyleGAN's latent space and see how the resulting images change. The difference is that StyleGAN allows for more control over how users do that. StyleGAN takes the latent vector and creates several transformed versions of it (we might call these 'sub-latents'). To synthesize an image, the sub-latents are passed as control inputs to different layers in the image

[4] https://gitlab.com/ktatar/latent-timbre-synthesis

synthesis network. StyleGAN's user can manipulate the raw latent vector, or they can manipulate one or more of the sub-latents before they go into the generator's layers. Users can even take latent vectors of two images, and mix and match which sub-latents they pass into the network, leading to mixtures between the two images. This is much more subtle than simply interpolating between the latent vectors of the two images.

The different layers of StyleGAN tend to represent semantically meaningful aspects of the training dataset, e.g. for faces, presence or absence of sunglasses, hair length, face angle and so on. That means that 'sub-latent mixing' allows for a technique called style transfer where semantic characteristics of one image can be applied to another. For example, given an image of a young person and an older person, it is possible to transform the older person's image such that they look young.

Earlier I described how there has been a to-ing and for-ing between formal and statistical methods in machine learning. Often, formal methods are associated with more explicit modelling of semantic features. It is interesting to note that the statistical structure learned by StyleGAN allows it to perform semantic style transfer between images.

3.4 A latent space for combined symbolic and sub-symbolic image data: VQGAN + CLIP

In 2021, the Internet witnessed a storm of creative image generation using a novel text to image converter called VQGAN + CLIP[5]. The text to image system relies

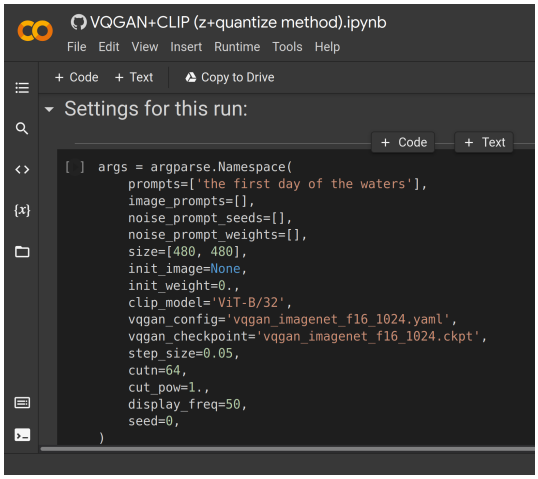

Fig. 6 The editable parameters for Katherine Crowson's VQGAN notebook viewed in Google colab.

[5] https://www.vice.com/en/article/n7bqj7/ai-generated-art-scene-explodes-as-hackers-create-groundbreaking-new-tools

Fig. 7 Response of the VQGAN network to the phrase ”the space of possible worlds”, iteration 0 (top left) then iteration 100 (top right), 250 (bottom left) and 500 (bottom right)

on two networks, CLIP and VQGAN (Razavi, Van den Oord, and Vinyals 2019; Esser, Rombach, and Ommer 2021). VQGAN can generate images from latent space similarly to StyleGAN. CLIP can evaluate an image as to how well it matches a text phrase. The method begins with the user selecting a textual input phrase and a ‘noise vector’ or what we know as a latent vector in VQGAN’s latent space. Then the system iteratively generates images from VQGAN using the latent vector, evaluating the images for how well they match the phrase with CLIP. Each iteration, the text to image matching error is used to fine-tune the weights in VQGAN so it produces better matches.

In more detail, the CLIP network learns a latent space of paired text and image features. Effectively, you can pass it text and an image and it will tell you how closely they match. It knows what the expected image features should be for a given text input, and it can compare those expected image features to the ones from the image

you pass it. That gives you an error between the expected image for a given phrase and the image you passed. On the other hand, the VQGAN network has learnt a latent space of a large number of images and it can then generate back out from that latent space to images. VQGAN is similar to StyleGAN in that sense. In fact, it is possible to swap out the image generator - an earlier iteration of this technique used a relative of StyleGAN called BigGAN as the image generator.

To use the two networks together, we start with a text phrase. Then we select a position in VQGAN's latent space. We reverse from VQGAN latent space to an image. Then we pass that image and the text phrase to CLIP. CLIP computes the features of the text and the image. Then it computes the distance between the text and image features. This provides an error value, which we use to fine-tune the weights on VQGAN using normal neural network training techniques. Then we try generating the image again, and so on.

The technique produces interesting results because the 'search' can start anywhere in VQGAN's latent space. The fine-tuning then warps the space so that the selected position moves to improve the CLIP rating. The warping is iterated, so you can extract images periodically as the warping is taking place. For example, you might happen to start at the position representing dogs in latent space, but want the phrase 'big green houses'. The result would be a dog like image iteratively warped into a big green house. The selection of the starting position and the target phrase, and the images that occur along the way provide for a huge range of creative possibilities.

The provision of this 'CLIP leading a generator' method in an accessible form is credited to machine learning engineers/ artists Katherine Crowson and Ryan Murdoch. They provided easily modified and executed Google colab notebooks containing the code for the implementation. Google colab makes it possible to run blocks of Python code on Google's compute infrastructure using just a web browser. Colab notebooks became something of a currency for digital artists in the early 2020s.

4 Creating with latent space systems

In this section, we will revisit each of the generative systems we discussed in the previous section and consider some examples of work that has been created using them.

4.1 Composers working with Music-VAE

In 2019, the band 'Young Americans Challenging High Technology', consisting of musicians Claire L. Evans & Ross Goodwin used Music-VAE in a music project. At a

Google I/O talk in 2019[6], Evans explained their creative process when they worked with Music-VAE in collaboration with the researchers who created it. Evans and Goodwin manually annotated their back catalogue of music into MIDI and extracted melodies, basslines and drum patterns. They trained a Music-VAE model on this corpus. According to Evans, Music-VAE "Allowed us to find melodies hidden in between songs from our own back catalogue". Thus they made use of Music-VAE's interpolation capabilities to find new melodies which carried the essence of YACHT music. In order to move from there to a full piece of music, they applied certain human workflow rules such as 'only use melodies generated from the model, do not improvise'. They used another model to generate the lyrics in a similar fashion, by training it on a corpus of their lyrics then generating from that latent space. To create the final track 'Loud Light', they took the MIDI data and lyrics from the machine learning models and created the final music using standard production techniques.

4.2 Composers using Latent Timbre Synthesis

Tatar et al. worked with nine composers to create a compilation album using the LTS system [7] (Tatar, Bisig, and Pasquier 2021). They report on the results of interviews with the participating composers wherein they queried many areas of the composers' experiences. Considering comments made by the composers with particular relevance to latent spaces, one "found the wide range of sound output possibilities of LTS rather exhausting". Other composers discussed different phases in their workflows. Tatar et al. characterised this as wider, exploration search followed by narrower, exploitation search. These concepts relate to the divergent and convergent strategies described by Tubb and Dixon (Tubb and Dixon 2014). The compositions and interview transcripts are publicly available[8].

4.3 Artwork using StyleGAN

Terence Broad is a practitioner in the creative visual domain using machine learning techniques. In the article 'Amplifying the uncanny', Broad et al. explore Mori's uncanny valley using StyleGAN (Mori 1970; Broad, Leymarie, and Grierson 2020). The researchers acknowledge that a criticism of generative image systems is that "the endless generation of samples from a given model, while initially mesmerising and transfixing, can quickly become banal, monotonous repetitions for the sake of overwhelming the viewer". Broad exploits this overwhelming feeling by attempting to create large palettes of uncanny faces. Broad also exploits the nature of the GAN training method, manipulating how the discriminator decides on the credibility of

[6] https://www.youtube.com/watch?v=pM9u9xcM_cs

[7] https://medienarchiv.zhdk.ch/entries/376e81a2-a6b9-4b74-91a3-9144c192f8e1

[8] https://medienarchiv.zhdk.ch/entries/376e81a2-a6b9-4b74-91a3-9144c192f8e1

the images from the generator. Thus he 'fools' the discriminator into guiding the warping of the manifold of latent space towards his own ends. That means training towards maximally uncanny faces instead of maximally realistic ones.

4.4 Artwork using VQGAN/ CLIP

As mentioned above, artist/ engineers Katherine Crowson[9] and Ryan Murdoch are credited with creating Google colab notebooks which allowed people to experiment with state-of-the-art text to image technology by editing some simple parameters in a text field and executing GPU accelerated neural network models for free on the Google compute infrastructure. Figure 6 shows a screenshot of the parameter panel in Katherine Crowson's VQGAN-CLIP notebook. These colab notebooks made high resolution, pre-trained image generators available to artists with minimal technical training.

Creating accessible technology which regular artists could use meant that in 2021, the currencies for some digital visual artists suddenly became pre-trained image processing neural network with acronymic names, colab notebooks and non-fungible tokens. Artists could simply edit a text field such as that shown in Figure 6 in their web browser, press a series of play buttons, and the notebook code and Google's compute infrastructure would do the rest of the work. Once created, artists could use NFT technology to place the images on the blockchain and place them onto the art market.

The work often combines multiple visual elements in dream-like, brightly coloured images. The artists sell the images as sets containing tens or hundreds of images. For further reading and examples of neural visual art using CLIP and GANs from 2021, we refer the reader to an article by Luba Elliott[10].

Having considered four examples where creative practitioners worked with machine learning tools and latent spaces, we will now leave this discussion with a quote from Memo Atkin. Atkin is a digital artist, researcher and long-time user of deep networks for image and video generation. Here is what Atkin had to say about generative systems in his 2021 PhD thesis:

> It is incredibly valuable that a person has the ability to freely explore such a massive space, so that they may embark on an goal-less, purely inquisitive and creative exploration, to build an understanding of the extents of such a system's creative capacity (Akten 2021).

5 Conclusion

In this article, I have considered how it is possible to use machine learning techniques to create and explore latent spaces for different creative domains. The latent

[9] https://kath.io/,https://twitter.com/RiversHaveWings

[10] https://www.rightclicksave.com/article/clip-art-and-the-new-aesthetics-of-ai

spaces might be symbolic, e.g. Music-VAE or they might be sub-symbolic, as in Latent Timbre Synthesis. The spaces can even combine symbolic and sub-symbolic elements, as in the VQGAN + CLIP text to image system. I have presented examples of creative practitioners working with each of the example systems, illustrating how creatives might go about exploiting their capabilities. Some used the systems to learn and explore latent spaces of their own previous work, others exploited the techniques to deliberately create 'non-optimal', uncanny output. I also explained the importance of accessibility of the technology. Providing artists with pre-trained models, easy to hack code and easy access to compute infrastructure lead to a fantastic explosion of new visual work in the early 2020s. I look forward to the future of creative exploration of latent spaces and expect to see fantastical images and hear unheard sounds.

References

Akten, Memo (May 2021). "Deep Visual Instruments: Realtime Continuous, Meaningful Human Control over Deep Neural Networks for Creative Expression". PhD thesis. Goldsmiths, University of London. URL: https://research.gold.ac.uk/id/eprint/30191/.

Ames, Charles (1987). "Automated composition in retrospect: 1956-1986". In: *Leonardo*, pp. 169–185.

Austin, Larry, John Cage, and Lejaren Hiller (1992). "An Interview with John Cage and Lejaren Hiller". In: *Computer Music Journal* 16.4, pp. 15–29.

Bradski, Gary and Adrian Kaehler (2000). "OpenCV". In: *Dr. Dobb's journal of software tools* 3, p. 2.

Briot, Jean-Pierre, Gaëtan Hadjeres, and François-David Pachet (2020). *Deep learning techniques for music generation*. Vol. 1. Springer.

Broad, Terence, Sebastian Berns, et al. (2021). "Active Divergence with Generative Deep Learning - A Survey and Taxonomy". In: *CoRR* abs/2107.05599. arXiv: 2107.05599. URL: https://arxiv.org/abs/2107.05599.

Broad, Terence, Frederic Fol Leymarie, and Mick Grierson (2020). "Amplifying the uncanny". In: *arXiv preprint arXiv:2002.06890*.

Casey, Michael A (2005). "Acoustic lexemes for organizing internet audio". In: *Contemporary Music Review* 24.6, pp. 489–508.

Collins, Nick, V Ruzicka, and Mick Grierson (2020). "Remixing AIs: mind swaps, hybrainity, and splicing musical models". In: *Proc. The Joint Conference on AI Music Creativity*.

Deng, Jia et al. (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Ellis, Daniel PW and John Arroyo (2004). "Eigenrhythms: Drum pattern basis sets for classification and generation". In: *ISMIR 2004: 5th International Conference on*

Music Information Retrieval: Proceedings: Universitat Pompeu Fabra, October 10-14, 2004. Ed. by Ramon Loureiro and Claudia Lomeli Buyoli.

Engel, Jesse et al. (2020). "DDSP: Differentiable digital signal processing". In: *arXiv preprint arXiv:2001.04643*.

Esser, Patrick, Robin Rombach, and Bjorn Ommer (2021). "Taming transformers for high-resolution image synthesis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883.

Fernández, Jose D and Francisco Vico (2013). "AI methods in algorithmic composition: A comprehensive survey". In: *Journal of Artificial Intelligence Research* 48, pp. 513–582.

Grierson, Mick et al. (2019). "Contemporary Machine Learning for Audio and Music Generation on the Web: Current Challenges and Potential Solutions". In: *45th International Computer Music Conference, ICMC 2019 and International Computer Music Conference New York City Electroacoustic Music Festival, NYCEMF 2019*. International Computer Music Association.

Karras, Tero, Samuli Laine, and Timo Aila (2019). "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *nature* 521.7553, pp. 436–444.

McFee, Brian et al. (2015). "librosa: Audio and music signal analysis in python". In: *Proceedings of the 14th python in science conference*. Vol. 8. Citeseer, pp. 18–25.

Mehri, Soroush et al. (2016). "SampleRNN: An unconditional end-to-end neural audio generation model". In: *arXiv preprint arXiv:1612.07837*.

Mori, Masahiro (1970). "Bukimi no tani [the uncanny valley]". In: *Energy* 7, pp. 33–35.

Oord, Aaron van den et al. (2016). "Wavenet: A generative model for raw audio". In: *arXiv preprint arXiv:1609.03499*.

Razavi, Ali, Aaron Van den Oord, and Oriol Vinyals (2019). "Generating diverse high-fidelity images with vq-vae-2". In: *Advances in neural information processing systems* 32.

Roberts, Adam et al. (2018). "A hierarchical latent vector model for learning long-term structure in music". In: *International conference on machine learning*. PMLR, pp. 4364–4373.

Roy, Pierre, Alexandre Papadopoulos, and François Pachet (2017). "Sampling variations of lead sheets". In: *arXiv preprint arXiv:1703.00760*.

Simon, Ian et al. (2018). "Learning a latent space of multitrack measures". In: *arXiv preprint arXiv:1806.00195*.

Tatar, Kıvanç, Daniel Bisig, and Philippe Pasquier (2021). "Latent timbre synthesis". In: *Neural Computing and Applications* 33.1, pp. 67–84.

Tubb, Robert and Simon Dixon (2014). "The Divergent Interface: Supporting Creative Exploration of Parameter Spaces." In: *NIME*, pp. 227–232.

Xenakis, Iannis (1966). "The Origins of Stochastic Music 1". In: *Tempo* 78, pp. 9–12.

Yee-King, Matthew and Louis McCallum (2021). "Studio report: Sound synthesis with DDSP and network bending techniques". In: *2nd Conference on AI Music*

Creativity (MuMe + CSMC). Graz, Austria 18 - 22 July 2021. Ed. by Artemi-Maria Gioti and Gerhard Eckel.

Zukowski, Zack and CJ Carr (2018). "Generating black metal and math rock: Beyond bach, beethoven, and beatles". In: *arXiv preprint arXiv:1811.06639*.