Generative Search Engines: Initial Experiments

Simon Colton,[1,2] Amy Smith,[1] Sebastian Berns,[1] Ryan Murdock [3] and Michael Cook [1]
[1] School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[2] SensiLab, Faculty of IT, Monash University, Melbourne, Australia
[3] Department of Psychology, University of Utah, USA

Abstract

Text-to-image generation involves producing an image which somehow reflects a given text prompt. We investigate the creative potential of a novel approach to this task. This employs three neural models working in concert: a generative adversarial network producing images with input latent vectors chosen by a search guided by a pair of models able to assess the appropriateness of a generated image for a text prompt. For evaluation purposes, we re-frame the task to be analogous to Google-like image search and introduce notions of efficiency, fidelity, variety, sophistication and coherence in the generated images. We have found the approach remarkably successful and explore here its potential for various creative tasks. We propose two approaches to increase efficiency in the generative process, and we evaluate the approach in an experiment simulating commercial design usage. We further suggest ways in which a generative search engine could be used in videogame design via standard usages and via an always-on modality for continuous creativity.

Introduction

As in many areas of AI research and practice, deep learning is making advances for computational creativity applications that may have seemed impossible a few years ago. This is certainly the case in text-to-image synthesis tasks, where images are generated that somehow reflect a user-given text prompt [1]. The main application of this has so far been to produce images which reflect content prescribed by a text prompt, e.g,. "a yellow bird in a tree", aiming to produce photorealistic images which could have been photographed in physical reality. However, there has also been progress in generating images for more abstract prompts such as "the devil in my head", where images don't exist in reality and have to be imagined, and likewise artistic or visual prompts such as "a skyline in the style of Mondrian".

In the next section, we describe a new set of text-to-image techniques that have shown much promise for use in various creative projects. These employ a generative adversarial network (GAN), producing images with input latent vectors chosen by a search guided by a pair of models called CLIP. The CLIP models are able to assess the appropriateness of a generated image for a text prompt, hence act as a fitness function for search strategies. We provide some background to these techniques in the next subsection. To provide a context for benchmarking and evaluation, we re-frame the text-to-image task as akin to image retrieval, and so can propose measures based on internet-based search engines like Google image search, as described in the third section below. This enables us to propose five different kinds of creative task that text-to-image generation can currently help with, and we present a preliminary evaluation of the new methods for each kind of task.

Focusing on applications to design inspiration, we propose and validate two methods for improving the speed of the search when multiple images are returned for a single prompt. To further explore the creative potential of the new wave of text-to-image generators, we simulated a scenario whereby a trained artist employed the approach as if it were an internet image search engine, for inspiration in the commercial setting of tattoo design. We report the overall success of this simulation as further evidence of the potential for neural text-to-image generators to drive forward computational creativity research. To supplement this, we further speculate on how generative search engines could help in videogame design, through standard usages and an always-on approach as a team member. We conclude by describing some of the many future directions for this research.

Background

Generative models such as variational autoencoders [22] (VAEs) have made successful inroads into automated image production by synthesising new images (fakes) to look like they come from a given distribution [34]. VAEs work by compressing media such as images into a smaller *latent space* of vectors in such a way that they can be decompressed to produce an output that resembles the original media. This enables the direct production of latent vectors, e.g., through random means or a search process, that can be decompressed for generative purposes, and numerous artists have used and abused this approach, as described in [2].

A particularly successful technique for developing image generation models involves adversarial training of two models simultaneously, one to generate fake images and one to critique the images in terms of how realistic they look when compared to a corpus of real images [16]. Some of these *Generative Adversarial Networks* (GANs) have been trained for specific generation of a single type of image, e.g., faces [20], and others have been trained to generate multiple types of images. For instance, the BigGAN model [5] produces images in 1,000 categories, given a random latent vector and a one-hot class vector specifying a required category.

A technique called *contrastive learning* aims to train two compression models at once, so the latent representations

Proceedings of the 12th International Conference on Computational Creativity (ICCC '21)
ISBN: 978-989-54160-3-5

of two different, but related, media items are similar. In particular, OpenAI recently made available a pair of pre-trained models called CLIP [27], which encode images and text prompts as latent vectors so that the cosine distance between encodings for (image, text) pairs that are related is smaller than for pairs where the text and image are not related. CLIP was trained over 400 million image/text pairs scraped from the internet, and has captured a broad and deep understanding of correlations between text and images, covering visual elements such as content, mood, texture, pattern, lighting, emotion and genre, as well as individual objects, people (if numerous examples exist on the internet) and artistic styles. A similar effort to produce a contrastive model called ALIGN has been reported by Google in [19].

CLIP-Guided GAN Image Generation

Colab notebooks [3] are free interactive Python programming environments which are connected to a CPU or GPU provided by Google. To increase accessibility, notebooks can be configured to hide code and expose GUI elements such as text boxes and drop-down lists. In recent practical work, a community of deep learning artists and researchers have been sharing code repositories and Colab notebooks which enable users to type in prompts and generate images which reflect the text. These notebooks largely share the same structure, as follows. For a user-given text prompt P:

(i) A pre-trained generative adversarial network (GAN), G, is loaded into memory.

(ii) The CLIP pair of pre-trained models is likewise loaded.

(iii) A randomly-generated latent vector, V, is input into G, and an image, I, is generated.

(iv) The cosine distance between the CLIP-encoded representations of prompt P and image I is used in a calculation to estimate the fitness of I with respect to it reflecting P.

(v) The fitness is used as a loss function, to update V in a backpropagation search, producing a new image I.

Steps (iii) to (v) are repeated until the user stops the process.

The notebooks differ mostly in terms of the GAN model they use for image generation, and certain (often extensive) technical tweaks required for search success over the latent vectors in each case. Four examples notebooks that we have experimented with are: the *Big Sleep* [25] which uses BigGAN and was written by Ryan Murdock (fourth author); *Aleph* [24], which uses the DALL-E generator [28]; *Siren* [13], which uses the image generator described in [31]; and *Aphantasia* [12], which uses texture generation via the Lucent PyTorch library [21]. We have found that these notebooks are not particularly robust to change, i.e., while the notebook may still work, a change could mean that the images produced no longer reflect the prompt given, or indeed have any content at all. Note that a technical description of a similar approach called CLIP-GLaSS is given in [14].

We concentrate here on the *Big Sleep* implementation, where CLIP is used to guide the BigGAN generator. To utilise this in the context of a generative search engine (see below), we copied the Python code from the Colab notebook to a server at our disposal. We also exposed some parameters and made some improvements. In particular, in the original implementation, the fitness of a generated image is calculated as the average over 128 randomly chosen sub-images called *cuts*. The cuts are chosen according to a normal probability distribution with mean $0.8s$ and standard deviation $0.3s$, where s is the generated image size, namely 512×512 pixels. After some initial experimentation, we determined that the simpler, faster, approach of assessing only the entire image doesn't work, but the number of cuts can be reduced to improve efficiency. As described below, we exposed to experimentation the number of cuts, as well as the learning rate hyper-parameter for the backpropagation search.

Figure 1: First images generated by the *Big Sleep* Colab notebook for the prompt "A skyline in the style of Mondrian", and for the prompt "The inside of a black hole".

Also described below, we experimented with various halting mechanisms for the search, as the original *Big Sleep* implementation relies on users stopping the process when they are happy with the output (or want to discard it). We also implemented the ability for users to provide a target image, T, as well as a text prompt, P, and for the fitness of a generated image, I, to be estimated in terms of a weighted sum of the cosine distance between I and P and the cosine distance between I and T. We implemented the ability for users to specify a fixed category for the class vector input to BigGAN, which constrains the image generation, and a way to spawn up to four server processes for a single text prompt, so multiple images can be generated simultaneously.

In general, we have found the output from the implementation to be remarkably good for a range of different prompts requiring the generation of images ranging from mundane and photo-realistic to abstract and artistic. This has been reflected by hundreds of cherry-picked images shared by dozens of users on social networks like Twitter. Example images arising from our very first usage of two prompts (i.e., with no cherry picking) are given in figure 1, and numerous further examples are given in the remainder of the paper. We have been impressed by three elements of the *Big Sleep* process. Firstly, it was surprising that BigGAN has such enormous generative potential, as it was trained to produce photo-realistic images of hamburgers and dogs, so we didn't expect it to have points in the latent space corresponding to images appropriately reflecting prompts such as "A holographic skull", as described in [32] or "The inside of a black hole", as shown in figure 1.

Secondly, CLIP's understanding of how well a piece of text and an image match each other is surprisingly broad and deep. As an example of this, when using the name of one the first author of this paper (Simon Colton) as the prompt, it guided BigGAN to produce images of superheroes, because another person with the same name has a modest online presence including images of superheroes. Thirdly, we've been surprised by how quickly the backpropagation search can find appropriate latent vectors for BigGAN, starting from a random position, sometimes converging after 50 steps and usually before 200. In particular, researchers in the online art/technology community developing text-to-image Colab notebooks have also informally experimented with using evolutionary approaches to search for latent vectors. Unpublished results indicate that the process was up to three times slower and not nearly as successful – in terms of image quality – as the backpropagation approach.

Generative Search Engines

OpenAI is reserving for commercial exploitation a one-shot text-to-image generation system which has been used for their DALL-E project (openai.com/blog/dall-e). This can take a text prompt, encode it into a latent space and then decode it as an image, without any search required, using the CLIP encoding methods. This, and the pace at which generative deep learning is advancing, points to future one-shot implementations that will be reliable and fast. Hence we can make a meaningful comparison with Google-like internet image retrieval engines, i.e., we can imagine in the near future that standard online image retrieval searches are supplemented with **generative search engines** which make, rather than retrieve, images. The latter will complement the former by being able to produce images that don't, or couldn't, exist in reality, including images that would normally require imagination in people to produce.

In this context, it would seem sensible to evaluate the *Big Sleep* and other generators in terms of image retrieval [30]. We draw from this literature and from assumptions about normal usage of web search engines to propose the following measures for the success of a generative search engine. In the scenario where a user-given prompt has returned multiple images, we can evaluate the following properties:

- **Efficiency:** how fast the full set of images are generated.
- **Fidelity:** how well the images reflect the prompt in the subjective view of the user.
- **Variety:** how visually or conceptually varied the set of generated images are, in the subjective view of the user.

In addition, while we would expect a Google search to return relatively sophisticated images, we have found that image generation can result in failures because the images are (a) devoid of detail, i.e., blank or roughly patterned images or (b) detailed but too noisy to be interpretable. Moreover, in cases where the image is detailed and interpretable with respect to general impression of the given prompt, sometimes the images are too incoherent to be of any value. Hence, we suggest also considering the following two measures of value for individual generated images:

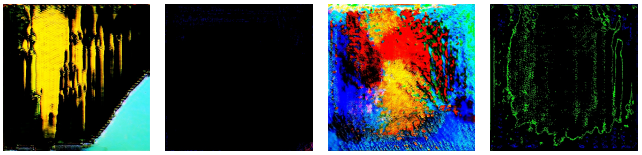

Figure 2: Images rejected for lack of sophistication

- **Coherence:** how well the images stand up to scrutiny on detailed inspection.
- **Sophistication:** how detailed, information-rich and interpretable the images are.

Fig. 2 shows images rejected for lack of sophistication, and we return to the question of coherence in the next section.

Creative Usage Scenarios

It is impossible to characterise all the ways a generative search engine might be used in creative projects. To begin to understand the potential, we propose below five different scenarios where a generative search engine might feasibly be employed, followed by a preliminary evaluation.

Artistic exploration

The majority of people using the Colab notebooks have so far done so for artistic and entertainment purposes, largely posting generated images on social media and blog posts, e.g., [29]. There has been much activity and cherry picking of the results, from which an overly successful view of the approach might be gained. That said, we have successfully used a combination of the four notebooks mentioned above for an art project [32], and found the process rewarding. Each different art project engaging the notebooks has differed in terms of the image requirements, and the same will likely be true for generative search engines. In our usage, a high variety of images over the same prompt was always valuable, along with sophistication of the images. However, we were often less concerned with fidelity to the prompt and coherence of the images, as we wanted to encourage imaginative interpretation of the results by audience members. In many cases, we were more interested in how the content of the images portrayed mood and style rather than coherence. Two example images in this vein are given in figure 3.

Figure 3: Images generated by the Big Sleep Colab notebook for prompts: "A galaxy funfair ride" and "A painting of a cauldron of magic". Note the word 'galaxy' added to image 1, which is not uncommon.

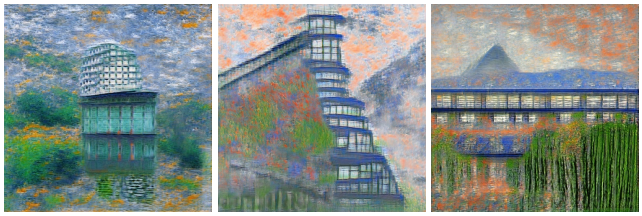

Figure 4: Images generated for the prompt "A modernist building in the style of Claude Monet".

Automated artistic treatment

Building on much work in graphics-based pastiche generation, neural style transfer [15] enables visual answers to questions such as: "If Claude Monet painted exactly this scene, what might it look like?" A next logical step is to ask questions like: "If Claude Monet painted this subject matter, what might it look like?" This is enabled by CLIP-guided GAN image generation, which, due to its random start, can produce multiple treatments of the subject. For instance, to explore an artistic treatment that couldn't have happened historically, using the prompt "A modernist building in the style of Claude Monet" for our *Big Sleep* approach, the images in figure 4 were produced. We see there appropriate, novel, buildings in a style/setting which reflects the artist.

Image manipulation

As mentioned above, we enabled the process to take a target image and a text prompt – as pioneered by artist Mario Klingemann[1] – in order to generate images which look like the target, but also reflect the text. As examples, in figure 5, target images of two faces were given, along with three text prompts. Equal weighting of similarity to the image and the prompt was specified. We see that each generated image resembles the subject matter (person) and reflects the text prompt. Also given in figure 5 are some generated images starting from a photograph with text prompts "In the style of X" where X is a well known artist. This is another application of automated artistic treatment as above, and we see that, unlike with standard style transfer, while the subject matter is per the original digital photo, the scene portrayed varies from it. This helps to extend the generation of pastiche images in an artist's style to include content and scene arrangement, rather than just transferring textures, colours and abstractions onto a given digital photo as in [15].

Imaginative idea visualisation

While stock photography and internet searches will satisfy many design needs in terms of photos of scenes from reality and artistic images of imagined realities, they will never be able to satisfy every visual scene that someone might want. For instance, the internet search prompt: "Church in an eyeball" produces no useful results from a Google image search. Text-to-image generative search engines can be used for these kinds of tasks, and many highly imaginative scenes have been visualised using the Colab notebooks and shared on social media. As an example, using our version of the *Big*

[1] twitter.com/quasimondo/status/1353300845266411521

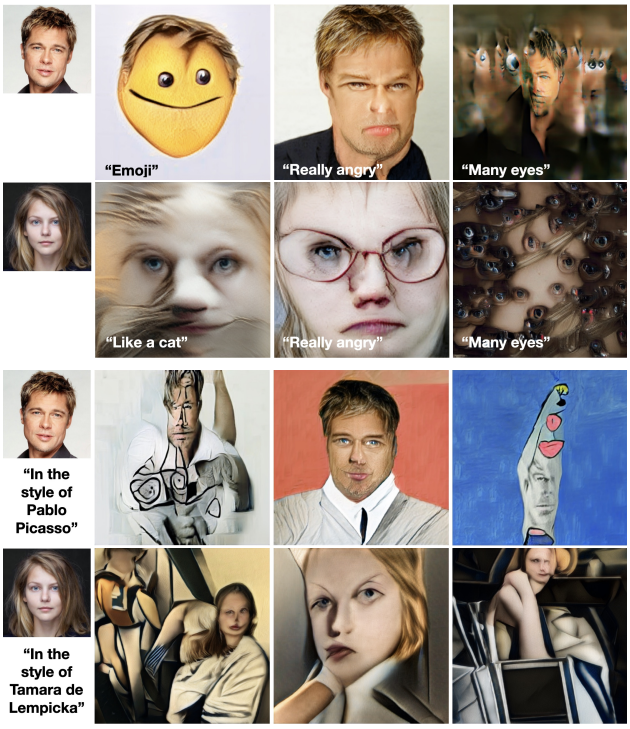

Figure 5: Top two lines: Target image, text prompts and generated images. Bottom two lines: target image, artistic style prompt and generated images.

Sleep, the images in figure 6 were returned for "Church in an eyeball". As another example, the prompt "An iPhone in the snow" produced an image where the iPhone was stood up, as if the snow was a dock for it. Such visualisation techniques could be highly valuable during creative ideation stages in, for example, advertising and other creative industries.

Design inspiration

In certain scenarios, the generative search reliably produces highly varied, coherent and sophisticated images with high fidelity. In such cases with high reliability, the generated images could be mined for design inspiration, with ideas being extracted directly from the imagery produced. Design inspiration tasks undertake imaginative idea visualisation as above, hence design inspiration extends idea visualisation. The difference comes in terms of the coherence of the images produced: if the images are coherent enough then they can be examined and may provide design inspiration; if not, then only a general impression can be gained.

Figure 6: Generated images for "Church in an eyeball."

A Preliminary Evaluation

We plan to investigate fully the potential of generative search engines in each of the above usage scenarios and more, but we concentrate in the next two sections on design inspiration projects. As an indication of the value of the approach for design inspiration, the high-quality images in figure 7 were all produced in a batch of 100 using the same text prompt.

For the artistic exploration usage, it is clear that the new methods work and are having an impact, as evidenced by the sharing of hundreds of generated artistic images in blog posts, NFT platforms and social media, and in research papers like [32] since the notebooks were published in early 2021. In these cases, images failing all but the sophistication measure could be considered useful in art projects. That is, incoherent images and those not matching the prompt very well could be presented as more abstract pieces requiring interpretation, and users seem prepared to cherry pick from what may be a small yield with low variety.

For the automated artistic treatment usage, image manipulation and idea visualisation scenarios, we tested 50 different text prompts and 50 target-image/prompt pairs which we felt were representative of the kind of things searched for, with examples given in figure 5. For each prompt, we generated 25 images using 300 search steps and the standard *Big Sleep* setup. We recorded a *curation coefficient* – defined informally in [7] as the proportion of images we were prepared to show people – and reasons for failures. We found:

• When producing artistic treatment images, we tried prompts covering objects, buildings and people, and a dozen visual artists with distinctive styles, such as Picasso, Monet, de Lempicka and O'Keefe. We found that the process often failed because the design problem was too hard, and report a curation coefficient of only around 20%. The process is most successful for famous artists like Picasso, who has many different styles including abstract and cubist treatments. This is because (a) the fame (hence much coverage in the 400 million internet training examples) and variety of styles means that CLIP has multiple options for guiding the search to a local minima, depending on the random starting point for the latent vector, and (b) unusual scene constructions – in the sense of representing physical reality – are associated with cubism and abstraction, hence jumbled and incoherent images, which are often generated, are fine.

In other cases, with higher realism in the scenery, for instance with Tamara de Lempicka, work less well, as the jumbled generated content lowers the values of the images, even if the painting style is transferred. To optimise the fidelity of this approach, further study of how CLIP has entangled subject matter and style (e.g., portraits and cars versus rich colours in the art of de Lempicka), when learning about the visual properties of artists' works, will be required. This may lead to automated *prompt engineering* of the text that is used to guide the search, in order to improve matters. Another approach is to split the generation into two parts, for instance first generating a content image with a suitable prompt and then using the latent vector for this as the starting point, rather than a random vector, with the prompt "In the style of ..." for the artist. If the second search uses a much smaller learning rate, it can tweak the generated image and often produces an appropriately stylised image.

• For imaginative idea visualisation, there is much variation in the fidelity of the output, depending on the nature of the prompt. In particular, there are sweet spots for specificity, coherence and constrainedness of the text prompt, and more study of user expectations for prompt types will be needed to make generative search engines satisfying. For instance, while a prompt such as "a white blackness" is over-constrained in the sense of being logically unachievable, users know this and may be more amenable to any appropriate output. This may similarly be true for prompts such as "Venus in love" that can be satisfied appropriately in numerous, sometimes abstract ways, as the user will have an expectation that the result will require interpretation. On the other hand, for highly specific prompts such as "an orange on top of a lemon on top of a table" there will likely be lower expectation that the results will need interpretation, and users may be more critical of poorly composed results. We found best cases with a 25% curation coefficient, e.g., for the prompt "A church in a desert" and "A necklace made out of coal", to worst cases with 0% curation coefficient, e.g., for "A palace with no windows". The average curation coefficient was around 5%, with most images being rejected due to lack of fidelity and/or lack of coherence.

• The image manipulation scenario was the least satisfying, with an average curation coefficient of around 2%. This is not surprising given that the constrainedness of the problem is doubled when both a text prompt and a target image are used. We only rarely found outputs which looked like the target and reflected the prompt, although when this did happen, the images were surprisingly good. Fortunately, there are other approaches to GAN-based image manipulation which could be incorporated into a generative search engine, e.g., using StyleGAN in [26].

Figure 7: Images generated in the same batch of 100 images by our version of the *Big Sleep* generator for the prompt: "Any architectural work that does not express serenity is an error" (a quote from architect Luis Barrigan).

Improving Efficiency for Design Inspiration

In porting the *Big Sleep* code from a Colab notebook to a server at our disposal, we took the opportunity to implement some improvements. In particular, as BigGAN was trained on the ImageNet competition images, a random latent vector coupled with a random class vector usually produces an image of a dog or a bird. After the search process, many generated images retained the look of a dog or bird, which becomes tedious very quickly. To improve matters, while keeping the randomness, which is a key to the success of the process, we enabled the search to start with 20 random pairs of latent/class vectors and choose the one with CLIP encoding closest to the CLIP encoding of the text prompt, yet furthest from the CLIP encodings of "dog" and "bird".

One of the most successful series of prompts have been related to architecture, where generated images have been reliably high in fidelity, coherence, sophistication and variety, as evidenced with images for the same architecture prompt in figure 7. We see that all the images portray serene architecture, yet they are quite varied, and – while not perfect – they have enough coherence and sophistication that architectural elements could potentially be extracted for design inspiration. Given this, we were able to concentrate on improving the generative search engine efficiency in this domain.

Curtailing Search

The fitness of a generated image is calculated as the average of the cosine distances of the CLIP encodings of n sub-image *cuts* of it from the CLIP encoding of the text prompt, with $n = 128$ in the original implementation. Cosine distance is computed along dimension -1 then scaled by -100, and high quality images tend to range in fitness from around 20 to 50. We've also found that 300 iterations improving the latent vector is almost always enough for the search to converge onto an image, and further steps will likely not change matters. Envisioning a situation where natural language processing could be used to analyse and improve prompts, and to choose specialist procedures, we experimented to test such a procedure. In particular, we've found that any generated image with architectural content that has fitness 29 or above will be almost guaranteed to have high sophistication and fidelity, and sufficient coherence around 80% of the time. Examples of generated architectural images that we judged to be too incoherent are given in figure 8 – we see that on close inspection, the buildings are too confusing to easily extract design ideas from.

Given the reliability mentioned above, it is possible to test strategies which curtail a search when a generated image reaches the fitness bar of 29. In the setting of a generative search where users expect high quality images returned quickly, we can hypothesise that a fail-fast approach – where searches that are clearly not working are abandoned early – would have benefits. To test this, we took 10 well known quotes from architects, such as "Architecture aims at eternity" (by Christopher Wren) verbatim as text prompts for generative sessions producing 25 images each. We varied the backpropagation learning rate over the values $\{0.07, 0.09, 0.12, 0.15\}$ and the number of cuts ranged over $\{32, 50, 64, 80, 100, 128\}$, producing $50 \times 4 \times 6 = 1200$ batches of 25 images. We ran each search for a full 300 iterations, which enabled us to simulate different search-halting strategies to find the most efficient ways to generate a certain number of quality images (on average).

In simulating a generative search engine, we modelled the usage of 1, 4 and 16 GPUs with load-balancing, meaning that each of the 25 image generation processes are run on the next available GPU. We specified that simulated searches should stop immediately and return any image with fitness ≥ 29. We simulated a total limit of number of iterations from the range $\{150, 200, 250, 300\}$, at which point the best ever (in terms of fitness) image is returned. Moreover, to test fail-fast strategies, we further specified that searches should stop and return no image if the fitness is below f after s iterations with (s, f) ranging over $\{(10, 15), (30, 20), (50, 25), (70, 27), (90, 28)\}$. These pairings were chosen by examining failed generations leading to clearly low sophistication images.

Depending on the search requirements, it is possible to optimise the search for number of quality ($f \geq 29$) images returned or speed, or a balance of both. With the parameters exposed during the generation sessions, along with those for the simulation, we were able to select from the 1,920 search setups only those which yielded at least 1, 5 and 10 quality images in every batch. These were then sorted in increasing time taken, and the fastest setups given in table 1. We see that, if speed is the most important factor in the generative search engine, then reducing the cuts and increasing the learning rate causes fails quickly and the session ends in 442s, 118s or 53s for the entire session, run on 1, 4 and 16 GPU setups respectively. The down-side to this is, of course, an expected yield of only 4.2 quality hits per batch. On the other hand, if quality yield is the priority, then increasing the cuts, decreasing the learning rate and scrapping the early abandonment of searches can produce an expected yield of 10 quality images, albeit in slower times of 752s, 214s and 87s. As shown in table 1, the compromise of producing an

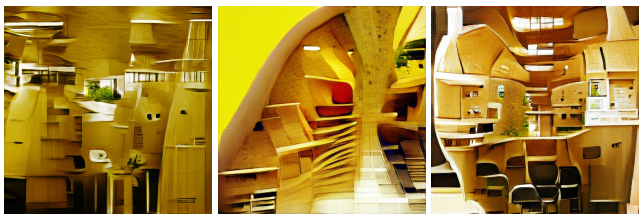

Figure 8: Generated images failing on coherence.

R	End	Cuts	LR	(s, f)	Hits	1/4/16 GPUs (s)
1	200	64	0.12	(50, 25)	1 / 4.2 / 10	442 / 118 / 53
5	300	80	0.09	(30, 20)	7 / 9.5 / 10	659 / 109 / 74
10	300	100	0.09	None	10 / 10 / 10	752 / 214 / 87

Table 1: (R)equired guaranteed quality results in each batch; example search setups, including total steps allowed (End), number of sub-image cuts, learning rate (LR) and ((s)tep, (f)itness) threshold; min/av/max quality results (Hits) and durations for 1, 4 and 16 GPUs with load-balancing.

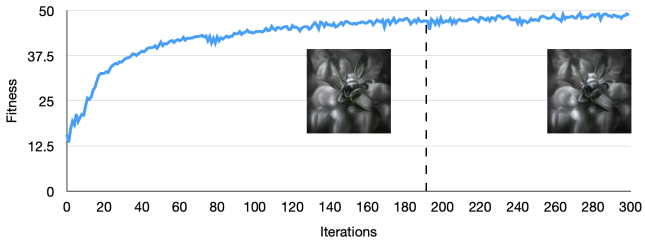

Figure 9: Fitness increase during search. First image: from start of the plateau (at dotted line). Second image: best seen.

expected yield of 9.5 quality images in 659s, 109s and 74s is possibly the best setup for the generative search engine.

Plateau Detection

As depicted in figure 9, the fitness of a generated image tends to plateau at some stage during the search (or it plummets to negative numbers). In the situation where it's not possible to know in advance what a suitable fitness threshold would be, this plateau can be detected and used to end the search. After some experimentation, we found the following routine for this to be satisfactory: using the numpy *polyfit* package, a line fitting the fitnesses recorded over a window of the previous 30 iterations is calculated, and if the gradient of this line becomes negative, the search stops and the best ever recorded image is returned. Note that the search is not stopped if a plateau is detected before 50 steps, as we found that this occasionally ended the process too soon. In figure 9, the search would have been terminated using this routine at step 192, saving 108 steps, and reducing the time taken from 113 to 78 seconds.

The best image generated (for the prompt "A painting of a realistic greyscale bee", as per the experiment described below) at this early stopping point and the best over 300 steps are both given in figure 9, and we see that they are nearly identical. Sampling 500 such pairs, we found that only around 10% of pre-plateau best images differed significantly from the best for 300 steps, and many were not always worse. We felt this was acceptable in return for an average speed-up of around 30-40%, depending on the search setup.

An Illustrative Design Project

To further explore the potential of text-to-image generative models for design inspiration and idea visualisation, we ran an experiment with a participant who has studied fine art to degree level, and who has had a career as a professional tattoo artist. The participant provided 8 client specifications for tattoos from past projects, summarised into short phrases:

A wicca-related colourful neo-traditional witch
A realistic greyscale bee
An anatomical heart with atmosphere
Alice in Wonderland with the Cheshire Cat
A greyscale geometric wolf head
A colourful slice of pizza
A colourful and fierce female gypsy wearing a headscarf
A deer head with flowers in a Victorian style frame

Each specification was taken verbatim as a text prompt for image generation using the plateau detection approach described above, and 25 images were generated. We also prefixed each of the following nine phrases to produce more text prompts: "A graphic design of", "A tattoo of", "An emoji of", "An illustration of", "A painting of", "An icon of", "A line drawing of", "A photo of" and "A sketch of". For each of the prefixed prompts, we generated 25 images, hence $8 \times 10 \times 25 = 2000$ images were produced in total. For each of the 80 batches of 25 images reflecting a prompt, the participant was then asked to assess (a) the number of images to reject immediately for reasons of fidelity, coherence or sophistication (b) of the remaining, how many provided some inspiration for a potential tattoo design, and (c) a score for diversity from 1 (low diversity) to 10 (high diversity). The participant also highlighted the kinds of visual and conceptual qualities in the images that might be useful for the task of tattoo design. Finally, the participant chose an image and produced a tattoo design inspired by it.

Results

On average, the participant rejected immediately 10.0 images per batch of 25, ranging from just 1 (for prompts including "A painting of Alice in Wonderland with the Cheshire Cat") to 22 (for the prompt "A graphic design of a greyscale geometric wolf head"). The participant reported that the main reason for rejection was lack of fidelity to the text prompt. Of the non-rejected images, on average 3.4 per batch provided some design inspiration. Only 3 of the 80 batches contained no inspiring image, e.g., for "A photo of a colourful and fierce female gypsy wearing a headscarf".

In some cases, as many as 10 of the 25 images provided design inspiration, e.g., for the prompt "A graphic design of an anatomical heart with atmosphere". An example image for this prompt, chosen by the participant, is given in figure 10. The participant reported the following qualities as providing inspiration: recognisable shapes (as a whole but also within the heart); appropriate use of colour (muted darker blues next to shades of a visceral crimson red); a serene white background (contrasting with the foreground heart adding atmosphere); an innovative angle and perspec-

Figure 10: Images with suitable qualities for design inspiration. First prompt: "A graphic design of an anatomical heart with atmosphere". Second prompt: "An illustration of a wicca-related colourful neo-traditional witch."

Proceedings of the 12th International Conference on Computational Creativity (ICCC '21)
ISBN: 978-989-54160-3-5

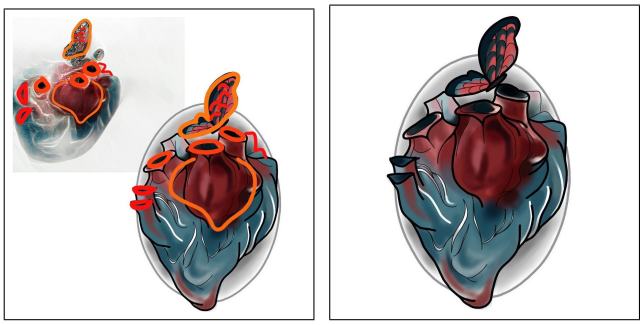

Figure 11: Design sketch indicating the influence of the generated image and the final tattoo design.

tive for the heart; the overall composition of the image (with the focal element taking up roughly two thirds of the space, adhering well to the rule of thirds); and the unprompted addition of a butterfly wing at the top of the image.

The participant chose this image to produce a final tattoo design for, as portrayed in figure 11, and we see that the butterfly wing has been incorporated. We expect such serendipitous effect to be common with generative search engines. The second image in figure 10 was generated for the prompt "An illustration of a wicca-related colourful neo-traditional witch". The participant reported favourably: an interesting composition (witch's pose, ratio of ground to sky), saturated fantasy themed colours, appropriate background (a suggestion of being outside in nature), texture and contrast.

Adding prefixes to the prompts is a form of *prompt engineering* which could in principle be done automatically to improve the quality of outputs. We found that the average number of rejects for the unadulterated prefixes was 4.125 and the average number of images providing design inspiration was 9.4. This baseline was improved upon by four of the prefixes, namely "graphic design", "illustration", "painting" and "sketch". The highest average yield of inspiring images (11.6 per 25) came from the "graphic design" prefix, and the lowest average number of rejects (4 per 25) came from "painting". The worst performing prefix was "line drawing", with, on average, 1.6 inspiring and 15.4 rejects per 25 images. The participant reported overall satisfaction with the process, highlighting the biggest differentiator for success/failure being the nature of the scene specified by the text prompt. Their impression was that too many images were too low in coherence for tattoo design, but when they were coherent, they were often genuinely inspiring.

Potential Applications to Videogame Design

The games industry represents a variety of exciting application areas for generative search engines. Games are complex to design and develop, requiring a wide variety of skills which leaves many smaller teams or individuals in need of creativity support tools. At the same time, large development studios need tools to help mitigate risk, support experimentation, and help large creative teams convey ideas to one another. We believe generative search engines can contribute to solving all of these problems.

Figure 12: Left: An image generated for the prompt "a mountain-top castle in the style of dark souls". Right: Concept art for the game *Dark Souls*.

Generative search engines can be applied to many of these use cases through standard prompt-and-curate processes similar to those we have described earlier in this paper. For instance, it is common to produce concept art and explore visual styles during a game's early pre-production. This can draw inspiration from many sources, and is usually highly freeform. Generative search engines could play a traditional role here in *imaginative idea visualisation* or even *design inspiration*, using prompts based on early design goals. As an example, we could imagine a design team working on a game that requires imagery inspired by the game *Dark Souls*, and trying the prompt "a mountain-top castle in the style of dark souls". Given that Dark Souls imagery was probably in the 400 million images used to train it, CLIP was able to guide BigGAN to produce a striking image when we used this prompt, as portrayed in figure 12. The compare, the second image in figure 12 is a real piece of concept art produced in pre-production for Dark Souls.

Many games are often made available to players while in development, as so-called 'early access' or beta versions. Such games often use placeholder art to make the experience feel more complete while the game content and design are finalised. This happened with games such as *DOTA 2* or *Slay The Spire*. In experiments using suitable prompts, our generative search engine was able to produce images which match or exceed the quality of such placeholder content, and would

Figure 13: A card depicting a flaming sword from the game *Slay The Spire*. Left: placeholder art. Centre: a generated mock-up created using the prompt "A fantasy sword enchanted with fire magic". Right: final in-game art.

be ideal for smaller art assets for things such as skill icons, user interface elements, or even character portraits. Figure 13 shows an example of placeholder art from the game *Slay The Spire*, along with an alternative placeholder using generated art as contrast, shown in the centre of the figure. Given our early impressions producing 100 images for each of 16 game-related prompts, we estimate curation rates for such applications would vary depending on the type of content required, from between 20% to close to 80% acceptable.

These straightforward applications of generative search engines may miss an opportunity to encourage different ways of engaging with these systems. We believe that such generative search engines could be set up to act as *continuously creative* AI agents, that function less like tools and more like co-workers [10]. In this setting, a generative search engine would be set up within a studio to continuously create new artworks based on prompts it receives from multiple sources. Some of these sources might be explicit – allowing any employee to add a prompt to a queue, for example. Other sources could be sought out by the system itself, through means such as studying changes made to game design documents or new features added to the game.

As an example, the art team working on pre-production of a new game might begin discussing possible directions for the art style and record notes on their meeting discussions citing sources of inspiration, or descriptions of styles they are interested in. Once these notes are uploaded and shared to the studio's servers, the always-on generative search engine identifies potential prompts from within this document, such as "a coloring book world full of vibrant characters", and produce some sample images that the art team can see and include in their planning and concept work. Figure 14 shows a series of example images created from this prompt, which is based on the top-level description of the game *Chicory*. These images are inspirational, suggesting colour palettes as well as form, texture and linework styles.

We believe that using generative search engines in such a way could transform our relationship with them. By reframing the system from a time-constrained tool to a creative collaborator, we shift expectations of the system as well as increasing its ability to surprise people and contribute meaningfully to active discussions. Such an approach may also alleviate problems such as curation, as since the system is working passively, its unused output can more easily be skimmed or ignored.

Figure 14: Image generated for prompts related to *Chicory*, a vibrant game about colouring in the world, e.g,. ""a coloring book world full of vibrant characters".

Conclusions and Future Work

The CLIP pre-trained models were released in early 2021, and there has already been an explosion of uses, including in text-to-image projects such as those described here. We have attempted to assess the potential value and impact of CLIP-guided GAN image generation, by looking at an implementation using CLIP to guide BigGAN. To the best of our knowledge we are the first to suggest switching context from text-to-image Colab notebooks to a generative search engine. This has enabled us to introduce appropriate measures of value, and increased efficiency with fail-fast and plateau-based stopping mechanisms. In an experiment designed to estimate the potential of generative search engines for design inspiration, a trained artist reported overall satisfaction and many benefits to the approach for creative design projects. We also highlighted some potential use-cases and interaction modalities for videogame design.

Visual imagination models [4], visual blending [11] and text-to-image approaches have been studied from a computational creativity perspective, e.g., text-to-emoji [33], newspaper text-to-collage [9] and poetry text-to-visual interpretations [17] have been investigated. With the increase in quality and fidelity afforded by automated search for GAN inputs, we expect them to be used in other computational creativity projects. One such area is fictional ideation [23], where automated invention of fictional ideas could be extended with automated visualisation of those ideas. In general, we expect to see much research into automated prompt engineering which can take a user's raw text and produce versions that increase the value of the images generated.

As with many deep learning techniques, such as human-steerable GANs [18], the methods described here can highlight biases inherent in pre-trained models such as CLIP, which must be catered for in generative search engines. For instance, Gustave Coral posted images of faces originally generated by CLIP-guided StyleGAN [20]. The prompt "Poor Man" produced the face of a person of colour, while "Rich Man" produced a white face (twitter.com/gustavecortal/status/1349826749404749824). We have experienced this ourselves, e.g., in figure 5, an image of a female face with glasses was produced in response to the prompt "really angry", which may highlight a bias.

We fully expect generative search engines to complement stock photography and internet image retrieval in creative industry practice in the short to medium term. To achieve this, improvements in the underlying generative processes will be needed, for instance training GANs specifically for the purpose of outputting imaginative visual imagery. Improvements in contrastive image/text learning and in one-shot image generation will also be required. It will eventually become as common to search for images that don't exist yet, as it currently is to search for images that do exist, and to expect high quality, on-point images to be returned. We believe that this will revolutionise the creative industries and provide many avenues for research in computational creativity. In particular, we are planning for The Painting Fool computational creativity system [6] to use a generative search engine in order to express aspects of its daily existence, as per the notion of the machine condition, described in [8].

Acknowledgements

We would like to thank the anonymous reviewers for their very helpful and thorough comments. We would also like to thank the ICCC'21 organisers for allowing camera ready versions of papers to have two extra pages, which helped a lot. Amy Smith and Sebastian Berns are funded by the EPSRC Centre for Doctoral Training in Intelligent Games and Games Intelligence (IGGI) [EP/S022325/1].

References

[1] J Agnese, J Herrera, H Tao, and X Zhu. A survey and taxonomy of adversarial neural networks for text-to-image synthesis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(4), 2019.

[2] S Berns and S Colton. Bridging generative deep learning and computational creativity. In *Proceedings of ICCC*, 2020.

[3] E Bisong. Google colaborator. In *Building Deep Learning Models on Google Cloud Platform*. Springer, 2019.

[4] V Breault, S Ouellet, S Somers and J Davies. SOILIE: A computational model of 2D imagination. In *Proc. 12th Int. Conference on Cognitive Modeling*, 2013.

[5] A Brock, J Donahue, and K Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv:1809.11096*, 2019.

[6] S Colton. The Painting Fool: Stories from building an automated painter. In *McCormack, J., & d'Inverno, M., eds., Computers and Creativity*, 3–38. Springer, 2012.

[7] S Colton and G Wiggins. Computational Creativity: The final frontier? In *Proceedings of ECAI*, 2012.

[8] S Colton, A Pease, C Guckelsberger, J McCormack, T Llano and M Cook. On the Machine Condition and its Creative Expression. In *Proceedings of ICCC*, 2020.

[9] M Cook and S Colton. Automated collage generation: With more intent. In *Proceedings of ICCC*, 2011.

[10] M Cook and S Colton. Redesigning computationally creative systems for continuous creation. In *Proceedings of ICCC*, 2018.

[11] J Cunha, N Lourenco, P Martins and P Machado. Visual Blending for Concept Representation: A Case Study on Emoji Generation. *Next Generation Computing*, 38, 739–771, 2020.

[12] V Epstein. The Aphantasia colab notebook, 2021. colab/github/eps696/aphantasia/blob/master.

[13] V Epstein. The Siren colab notebook, 2021. colab/drive/1l14q4to5rmk8q2e6whoibqbnpnvbrj_7.

[14] F Galatolo, M Cimino, and G Vaglini. Generating images from captions via CLIP-guided generative latent space search. *arXiv:2102.01645*, 2021.

[15] L Gatys, A Ecker, and M Bethge. Image style transfer using convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[16] I Goodfellow, J Pouget-Abadie, M Mirza, B Xu, D Ward-Farley, S Ozair, A Courville, and Y Bengio. Generative adversarial networks. In *Proceedings of NIPS*, 2014.

[17] E Gruss, A Sapirshtein, and V Heruti. Text to J** girls in synthesis. In *Proceedings ICCC*, 2019.

[18] A Jahanian, L Chai, and P Isola. On the "steerability" of generative adversarial networks. In *Proc. of ICLR*, 2020.

[19] C Jia et al. Scaling up vision-language representation learning with noisy text supervision. *arXiv:2102.05918*, 2021.

[20] T Karras, S Laine, and T Aila. A style-based generator architecture for generative adversarial networks. *arXiv:1812.04948*, 2018.

[21] L Kiat. Lucent library, 2021. github.com/greentfrapp/lucent.

[22] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *arXiv:1906.02691*, 2019.

[23] T Llano, S Colton, R Hepworth, and J Gow. Automated fictional ideation via knowledge base manipulation. *Cognitive computation*, 8(2):153–174, 2016.

[24] R Murdock. The Aleph colab notebook, 2021. colab/drive/1oa1fzp7n1upbxwbgivoexbtsq2ora9vb.

[25] R Murdock. The Big Sleep colab notebook, 2021. colab/drive/1nccex2mbikoslado7iu7na9uskkn5w.

[26] O Patashnik, Z Wu, E Shechtman, E Cohen-Or, and D Lischinski. StyleCLIP: Text-driven manipulation of StyleGAN imagery. *arXiv:2103.17249*, 2021.

[27] A Radford et al. Learning transferable visual models from natural language supervision. *arXiv:2103.00020*, 2021.

[28] A Ramesh, M Pavlov, G Goh, S Gray, C Voss, A Radford, M Chen, and I Sutskever. Zero-shot text-to-image generation. *arXiv:2102.12092*, 2021.

[29] J Shane. Sea shanty surrealism, 2021. aiweirdness.com/post/645282704595795968/sea-shanty-surrealism.

[30] N Shirahatti and K Barnard. Evaluating image retrieval. In *Proc. IEEE Conference on CVPR*, 2005.

[31] V Sitzmann, J Martel, A Bergman, D Lindell, and G Wetzstein. Implicit neural representations with periodic activation functions. *arXiv:2006.09661*, 2020.

[32] A Smith and S Colton. CLIP-Guided GAN Image Generation: An Artistic Exploration. In *Proceedings of the EvoMusArt conference*, 2021.

[33] P Wicke and J Cunha. An approach for text-to-emoji translation. In *Proceedings of ICCC*, 2020.

[34] X. Wu, K. Xu, and P. Hall. A survey of image synthesis and editing with generative adversarial networks. *Tsinghua Science and Technology*, 22(6):660–674, 2017.

colab is an abbreviation for: colab.research.google.com