Automating Generative Deep Learning for Artistic Purposes: Challenges and Opportunities

Sebastian Berns [1], **Terence Broad** [2,3], **Christian Guckelsberger** [4,5,1] and **Simon Colton** [1,6]

[1] School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[2] Department of Computing, Goldsmiths, University of London, UK
[3] Creative Computing Institute, University of the Arts London, UK
[4] Department of Computer Science, Aalto University, Espoo, Finland
[5] Finnish Center for Artificial Intelligence
[6] SensiLab, Faculty of IT, Monash University, Melbourne, Australia

Abstract

We present a framework for automating generative deep learning with a specific focus on artistic applications. The framework provides opportunities to hand over creative responsibilities to a generative system as *targets* for automation. For the definition of targets, we adopt core concepts from automated machine learning and an analysis of generative deep learning pipelines, both in standard and artistic settings. To motivate the framework, we argue that automation aligns well with the goal of increasing the *creative responsibility* of a generative system, a central theme in computational creativity research. We understand automation as the challenge of granting a generative system more *creative autonomy*, by framing the interaction between the user and the system as a co-creative process. The development of the framework is informed by our analysis of the relationship between automation and creative autonomy. An illustrative example shows how the framework can give inspiration and guidance in the process of handing over creative responsibility.

Introduction

The increasing demand in industry and academia for off-the-shelf machine learning (ML) methods has generated a high interest in automating the many tasks involved in the development and deployment of ML models. Such automated machine learning (AutoML) can make ML more widely accessible to non-experts, and decrease the workload in establishing ML pipelines, amongst other benefits.

AutoML is a very active area of research. The progress to date has been documented in several surveys (e.g. Truong et al., 2019; Tuggener et al., 2019; Chauhan et al., 2020; He, Zhao, and Chu, 2021). There exist a book (Hutter, Kotthoff, and Vanschoren, 2019), an AutoML challenge (Guyon et al., 2019) and a dedicated workshop at the International Conference on Machine Learning, currently in its seventh edition. Crucially though, the automation of generative deep learning (DL) as ML subdomain has received very little attention.

While AutoML is concerned with automating solutions for classification and regression, methods in generative DL deal with the task of distribution fitting, i.e. matching a model's probability distribution to the (unknown) distribution of the data. Neural architecture search (NAS), an important topic of research in AutoML, has been extended to generative adversarial networks (GANs) (Gong et al., 2019; Li et al., 2020; Gao et al., 2020; Fu et al., 2020), one prominent type of generative models. Moreover, evolutionary approaches have been applied to optimising the GAN training objective (Wang et al., 2019) and other training parameters (Costa et al., 2019). Even though certain aspects of the GAN training scheme have been automated, we highlight three gaps in existing research: (i) there exists no unified automation framework for generative DL more generally; (ii) existing work does not address the use of generative DL for artistic purposes; (iii) researchers have not sought to motivate the automation of DL systems with the goal to endow artificial systems with creative autonomy.

We propose a framework for the automation of generative deep learning that, on the one hand, adopts core concepts from AutoML, and on the other hand, is informed by the theory and practice of computational creativity (CC) research, the "philosophy, science and engineering of computational systems which, by taking on particular *responsibilities*, exhibit behaviours that unbiased observers would deem to be creative" (Colton and Wiggins, 2012, emphasis added). We can leverage insights from CC because automation in generative DL aligns with one of the field's central research goals: to endow computational systems with *creative responsibilities* (Colton, 2009), i.e. the ability to make specific decisions in a creative process. These decisions independently can be understood as *targets* for automation when framing the design of a generative DL pipeline as a form of *co-creativity* (Kantosalo et al., 2014). By virtue of this interpretation, we can inform the automation of generative DL more specifi-

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

cally with well-established, generic CC strategies to equip computational systems with creative responsibilities. Our framework differs from AutoML not only in its stronger focus on generative models, but also in the assumed goals of the generative DL pipeline. More specifically, we identify targets for automation based on the wide and successful application of generative DL in artistic work. In contrast to standard applications, artistic ML engineers and users aim to produce artefacts of high cultural value over perfectly generalised reproductions of the training data.

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

To leverage insights from CC in the development of our framework, we first clarify the relationship between automating generative DL and endowing artificial systems with creative responsibility. We then outline a standard non-automated pipeline for the development and deployment of generative deep learning models, and show how applications in artistic settings differ from this standard pipeline. Drawing from these two sources, we lay out the automated generative deep learning pipeline, describe several targets for automation therein and suggest ways in which automation could be achieved. We continue with an illustrative example to demonstrate how our framework can give inspiration and guidance in the process of gradually handing over creative responsibility to a generative system. We analyse the relationship between automation and creative autonomy in the context of our framework. We conclude the paper by discussing the limitations of our framework and suggest directions for future work.

Automated, Artistic Deep Learning as Co-Creation

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

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

To this end, it is useful to frame this interaction in the process of automation as a *co-creative* act. We adopt Kantosalo et al.'s (2014) working definition of *human-computer co-creativity* as "collaborative creativity where both the human and the computer take creative responsibility for the generation of a creative artefact". To qualify as a collaborative activity, both human and system must achieve *shared goals* (Kantosalo et al., 2014, drawing on Terveen, 1995).

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

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

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

We believe that providing the computational system with creative responsibility in the form of automating certain targets does not constrain, but rather expands the shared creative process. The *person* or *producer* has, due to their personality and cognitive characteristics, a strong impact on the creative *process*, *product*, and the creative environment, i.e. the *press* (Rhodes, 1961; Jordanous, 2016). However, human creativity is also limited, e.g. due to our bounded rationality (Simon, 1990). A computational system can complement human shortcomings, e.g. via its higher information processing or memory capacity, enabling creativity on larger search spaces (Boden, 2003; Wiggins, 2006).

A Standard Generative Pipeline and Artistic Deviations

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

Data Acquisition

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

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

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

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

Training

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

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

Evaluation

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

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

Iterative Improvements of Outputs

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

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

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

Deployment

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

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

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

Post-training Modifications

Having looked previously at the curation of a model's output in an artistic setting, i.e. the act of identifying the few artefacts of interest in a large set of output samples, we now turn to active divergence techniques (Berns and Colton, 2020) which aim at consistently producing results that diverge from the training data. These strategies, specifically developed in creative contexts for the purpose of art production, include hacks, tricks and modifications to the model parameters, as well as the daisy-chaining of several models.

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

Other approaches make modifications to the model in order to have artefacts completely diverge from any training data. An existing pre-trained model can be fine-tuned using a loss function that maximises the likelihood over the training data (Broad, Leymarie, and Grierson, 2020). Other techniques intelligently combine learned features across various models (Guzdial and Riedl, 2018), or rewrite the weights of the model (Bau et al., 2020), re-configuring them to represent novel data categories or semantic relationships. In contrast, *network bending* does not require any changes to the weights of the model (Broad, Leymarie, and Grierson, 2021). An analysis of the model is performed to determine which features are responsible for generating different semantic properties in the generated output. Deterministically controlled filters are then inserted as new layers into a model and applied to the activation maps of features.

An Automation Framework

We build our framework drawing on the standard generative DL pipeline and its artistic deviations, as previously described. We first discuss automation as a search problem and the various techniques it can be approached. We then go on to list the targets for automation in a generative deep learning pipeline for artistic purposes. Some decisions have implications for other targets further down the line, e.g. the number and type of hyper-parameters depend in part on the kind of network architecture and optimisation

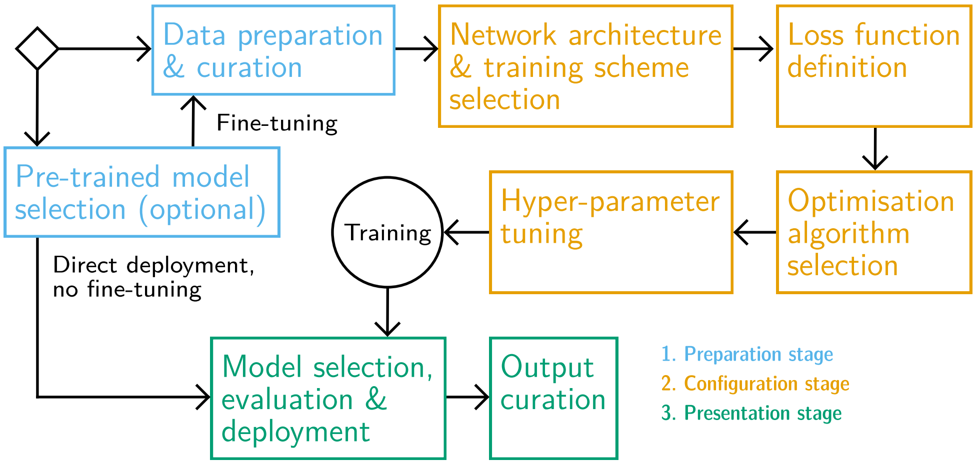

Figure 1: Automated generative deep learning framework. Three stages: preparation, configuration and presentation. Targets for automation are illustrated as individual steps and are further discussed in the section *Targets for Automation* below.

algorithm. While the process is presented as a sequence of consecutive steps from input to output, it should be understood that all steps are optional and flexibility is required. Improving a system's output works best as an iterative loop in which we might go back and adjust or intervene at any given prior step.

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

Automation as a Search Problem

A generative pipeline is automated by assigning responsibilities over individual targets to either the user or the system. While those retained by a person will have to be tuned manually, all other targets require the system to determine a configuration independently. This problem is analogous to the search problem over hyper-parameters in AutoML. The possible values of each automated target effectively construct a search space over possible system configurations. The number of total permutations, and the resulting search space, can grow rapidly with every independent target added.

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

The extensive work on search problems provides numerous approaches to constrain this search. Strategies range from complete, to informed, to random methods. While exhaustive search can yield an optimal solution, it can be impractical and often infeasible for large search spaces. Random sampling, on the other extreme, can be a surprisingly effective strategy at low cost and with potentially surprising results. While Jennings (2010) requires a system to meet the *non-randomness* criterion in order to be considered creatively autonomous, this definition does not rule out all uses of randomness and allows for testing random perturbations to a system's standards. AI-based search methods can benefit from meaningful heuristics and leverage both exploration and exploitation (e.g. evolutionary search). Gradient-based methods have seen a lot of progress in recent years. Other approaches include rule-based selection and expert systems, with drawbacks including that they require manual construction and expert knowledge.

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

Automation vs. Autonomy

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

We further assume the choice of generated media (image, audio, text, video, etc.) to be made by a person prior to building a system. Naturally, it is not difficult to imagine a setup in which this choice, too, becomes part of the pipeline. Going one step further in autonomous automation, our framework and its targets make it possible to devise a generative system which produces automated generative pipelines. In fact, it might be possible for a generative system to generate itself, much like a general-purpose compiler that compiles its own source code. This self-referential generation has similarly been proposed in work on automated process invention (Charnley, Colton, and Llano, 2014).

Targets for Automation

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

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

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

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

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

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

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

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

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

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

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

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

An Illustrative Example

In early 2021, a generative deep learning Colab notebook (Bisong, 2019) called the *Big Sleep* was shared online (Murdock, 2021). It allows for text-to-image generation (Agnese et al., 2020), effectively visualising a user-given text prompt, often with innovative content and design choices, as per the example in figure 2. This is an instance of an artistic deviation from the standard pipeline, where CLIP (Radford et al., 2021) is used to evaluate a generated image w.r.t. a given text prompt, driving a gradient-based search for latent vector inputs to a generative model called BigGAN (Brock, Donahue, and Simonyan, 2019). We use this setup as an example to identify the following places where automation could be introduced, based on our framework. We highlight concrete techniques and references for automation from the literature.

• In terms of **pre-trained model selection**, numerous people have substituted BigGAN with other GAN generators. This creative responsibility could be automated, with the system choosing from a database of GANs and installing new ones into the notebook.

Figure 2: Image generated by the *Big Sleep* Colab notebook for the prompt "The Melbourne skyline in pastel colours". Note the appropriate presentation of content and style, and additional pastel strokes in the sky as an unprompted innovation.

• In terms of **data preparation and curation**, users often choose imaginative text prompts, as the notebook often produces high quality, surprising results for these. This could be substituted, for example, with automated fictional ideation techniques (Llano et al., 2016).

• Murdock (2021), the notebook programmer, innovated in **loss function definition**, employing patches from generated images rather than the entire image to evaluate its fit to the prompt. Various image manipulation routines could be automatically tested within loss function calculations from a library, with the system automatically altering the notebook at code level.

• As described in Colton et al. (2021), in some circumstances where multiple images are being generated simultaneously, increasing the learning rate can help searches fail quickly. Such **hyper-parameter tuning** could be automated using standard AutoML techniques, guided by requirements on acceptable search successes and output image quality.

• In terms of **model selection and deployment**, we can imagine models being used as creative web-services (Veale, 2013), with higher-level CC systems accessing text-to-image generators in a variety of projects.

• There has been an explosion of human usage of notebooks like the *Big Sleep*, with attendant **output curation** via cherry picking results for posting on social networks and in blogs. This would be an ideal target for automation with systems using CLIP and other techniques to evaluate images, also possibly inventing new aesthetic measures (Colton, 2008a).

Discussion

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

Automation and Creative Autonomy

We have previously analysed the close relationship between the *automation* of generative DL systems and the central CC goal to increase a system's *creative autonomy* (Jennings, 2010; McCormack, Gifford, and Hutchings, 2019; Guckelsberger, Salge, and Colton, 2017) by granting it more *creative responsibilities* (Colton, 2008b). Here, we complement this *a priori* analysis based on knowledge of our concrete automation pipeline. The aim is to understand to which extent our proposed pipeline already enables facets of creative autonomy, and how CC insights on creative autonomy could be used to advance it in future work.

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

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

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

Linkola et al. (2017) follow a more constrained approach and, as part of a larger agenda to realise meta-creativity in CC, propose that creative autonomy requires *artefact-*, *goal-* and potentially *generator-awareness*, realised through operators of *(self-) reflection* and *(self-) control* which closely match Jennings' (2010) requirements for evaluation and change. Whether a system built within our framework satisfies these definitions depends on the extent to which it is granted responsibilities in the form of automating decision making for targets identified in the framework. We demonstrate this based on extensions to a non-automated generative DL system. Such a system can be considered to have some generator-awareness due to the role of its loss function (self-reflection), and its adjustments of own parameters through error correction methods like back-propagation (weak self-control). A system's control over changes to its generator can be increased from weak to strong within our framework, through the automated manipulation of network architecture or selection of a pre-trained model. Further putting a system in charge of its loss function within our framework (strong control) affords it goal-awareness and consideration as autonomously creative, if it is capable of modifying the loss function in response to its evaluation of generated output.

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

Future Work

In this formulation of our framework we have only briefly mentioned automation of creative responsibilities via the usage of ML models. The possibility of training or deploying multiple models in the same system enables the addition of organisational structures to our framework, in which we think of individual models as agents in a multi-agent system.

To use our framework in co-creative applications, augmenting a system with the ability to communicate its adjustments and intentions would be especially beneficial. Moreover, to address our framework's limitations, further work is needed to consider applications which use generative DL but are not artistically focused. This could potentially inform a more general automated ML framework, which would further benefit from more formal definitions.

We plan further study of the ways in which deep learning researchers, practitioners and artists work with generative systems, in particular where they have, and could, add levels of automation, via analyses such as the illustrative example above. Some of the techniques that artists apply, such as data set curation and iteration, as well as the selection of generated outputs, are promising avenues for automation and require further investigation. We further plan to put our framework to use in applied projects. Through this, we aim to provide demonstrative examples of how some of the challenges in automation can be tackled and to show the surprising results that automation can afford. For the evaluation of such demonstrative examples we plan to draw from the FACE descriptive model of creative acts (Colton, Charnley, and Pease, 2011; Pease and Colton, 2011).

Acknowledgements

We thank our reviewers for their helpful comments. Sebastian Berns and Terence Broad are funded by the EPSRC Centre for Doctoral Training in Intelligent Games & Games Intelligence (IGGI) [EP/L015846/1, EP/S022325/1]. Christian Guckelsberger is supported by the Academy of Finland Flagship programme "Finnish Center for Artificial Intelligence" (FCAI).

References

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

Azadi, S.; Olsson, C.; Darrell, T.; Goodfellow, I.; and Odena, A. 2019. Discriminator Rejection Sampling. In *Proc. ICLR*.

Bau, D.; Liu, S.; Wang, T.; Zhu, J.-Y.; and Torralba, A. 2020. Rewriting a Deep Generative Model. In *Proc. ECCV*.

Berns, S., and Colton, S. 2020. Bridging Generative Deep Learning and Computational Creativity. In *Proc. ICCC*.

Bisong, E. 2019. Google Colaboratory. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Springer.

Boden, M. A. 2003. *The Creative Mind: Myths and Mechanisms*. Routledge, 2 edition.

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

Broad, T.; Berns, S.; Colton, S.; and Grierson, M. 2021. Active Divergence with Generative Deep Learning - A Survey and Taxonomy. In *Proc. ICCC*.

Broad, T.; Leymarie, F. F.; and Grierson, M. 2020. Amplifying the Uncanny. *Proc. xCoAx*.

Broad, T.; Leymarie, F. F.; and Grierson, M. 2021. Network Bending: Expressive Manipulation of Deep Generative Models. *Proc. EvoMUSART*.

Brock, A.; Donahue, J.; and Simonyan, K. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *Proc. ICLR*.

Charnley, J. W.; Colton, S.; and Llano, M. T. 2014. The FloWr Framework: Automated Flowchart Construction, Optimisation and Alteration for Creative Systems. In *Proc. ICCC*.

Chauhan, K.; Jani, S.; Thakkar, D.; Dave, R.; Bhatia, J.; Tanwar, S.; and Obaidat, M. S. 2020. Automated Machine Learning: The New Wave of Machine Learning. In *Proc. Int. Conf. on Innovative Mechanisms for Industry Applications*.

Colton, S., and Wiggins, G. A. 2012. Computational Creativity: The Final Frontier? In *Proc. ECAI*.

Colton, S.; Smith, A.; Berns, S.; Murdock, R.; and Cook, M. 2021. Generative Search Engines: First Experiments. In *Proc. ICCC*.

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

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

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

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

Costa, V.; Lourenço, N.; Correia, J.; and Machado, P. 2019. COEGAN: Evaluating the Coevolution Effect in Generative Adversarial Networks. In *Proc. GECCO*.

Fu, Y.; Chen, W.; Wang, H.; Li, H.; Lin, Y.; and Wang, Z. 2020. AutoGAN-Distiller: Searching to Compress Generative Adversarial Networks. In *Proc. of the ICML*.

Gao, C.; Chen, Y.; Liu, S.; Tan, Z.; and Yan, S. 2020. AdversarialNAS: Adversarial Neural Architecture Search for GANs. In *Proc. CVPR*.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image Style Transfer Using Convolutional Neural Networks. In *Proc. CVPR*.

Gong, X.; Chang, S.; Jiang, Y.; and Wang, Z. 2019. AutoGAN: Neural Architecture Search for Generative Adversarial Networks. In *Proc. ICCV*.

Guckelsberger, C.; Salge, C.; and Colton, S. 2017. Addressing the "Why?" in Computational Creativity: A Non-Anthropocentric, Minimal Model of Intentional Creative Agency. In *Proc. ICCC*.

Guyon, I.; Sun-Hosoya, L.; Boullé, M.; Escalante, H. J.; Escalera, S.; Liu, Z.; Jajetic, D.; Ray, B.; Saeed, M.; Sebag, M.; Statnikov, A.; Tu, W.; and Viegas, E. 2019. Analysis of the AutoML Challenge series 2015-2018. In *AutoML*, Springer series on Challenges in Machine Learning.

Guzdial, M., and Riedl, M. O. 2018. Combinets: Creativity via Recombination of Neural Networks. *Proc. ICCC*.

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

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

Hutter, F.; Kotthoff, L.; and Vanschoren, J. 2019. *Automated Machine Learning: Methods, Systems, Challenges*. Springer Nature.

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

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

Kantosalo, A., and Toivonen, H. 2016. Modes for Creative Human-Computer Collaboration: Alternating and Task-Divided Co-Creativity. In *Proc. ICCC*.

Kantosalo, A.; Toivanen, J. M.; Xiao, P.; and Toivonen, H. 2014. From Isolation to Involvement: Adapting Machine Creativity Software to Support Human-Computer Co-Creation. In *Proc. ICCC*.

Karras, T.; Aittala, M.; Hellsten, J.; Laine, S.; Lehtinen, J.; and Aila, T. 2020. Training Generative Adversarial Networks with Limited Data. In *Adv. NeurIPS*.

Li, M.; Lin, J.; Ding, Y.; Liu, Z.; Zhu, J.-Y.; and Han, S. 2020. GAN Compression: Efficient Architectures for Interactive Conditional GANs. In *Proc. CVPR*.

Linkola, S.; Kantosalo, A.; Männistö, T.; and Toivonen, H. 2017. Aspects of Self-Awareness: An Anatomy of Metacreative Systems. In *Proc. ICCC*.

Llano, M. T.; Colton, S.; Hepworth, R.; and Gow, J. 2016. Automated Fictional Ideation via Knowledge Base Manipulation. *Cognitive Computation* 8(2).

McCormack, J.; Gifford, T.; and Hutchings, P. 2019. Autonomy, Authenticity, Authorship and Intention in Computer Generated Art. In *Proc. EvoMUSART*.

Murdock, R. 2021. The Big Sleep colab notebook. `https://colab.research.google.com/drive/1NCceX2mbiKOSlAd_o7IU7nA9UskKN5WR`. Accessed: 2021-04-01.

Pease, A., and Colton, S. 2011. Computational Creativity Theory: Inspirations behind the FACE and the IDEA models. In *Proc. ICCC*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning Transferable Visual Models From Natural Language Supervision. *arXiv preprint arXiv:2103.00020*.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot Text-to-image Generation. *arXiv preprint arXiv:2102.12092*.

Rhodes, M. 1961. An Analysis of Creativity. *The Phi Delta Kappan* 42(7):305–310.

Ridler, A. 2017. Repeating and Mistranslating: the Associations of GANs in an Art Context. In *Proc. NeurIPS Workshop on ML for Creativity and Design*.

Ritchie, G. 2007. Some Empirical Criteria for Attributing Creativity to a Computer Program. *Minds and Machines* 17(1).

Runco, M. A., and Jaeger, G. J. 2012. The Standard Definition of Creativity. *Creativity Research Journal* 24(1).

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved Techniques for Training GANs. In *Adv. NIPS*.

Schultz, D. 2020. Make ML Art Datasets: Week 3. `https://www.youtube.com/watch?v=0StKJxx_Dfc`. Accessed: 2020-03-29.

Shane, J. 2018. Machine learning failures - for art! `https://www.youtube.com/watch?v=yneJIxOdMX4`. Accessed: 2020-03-29.

Simon, H. A. 1990. Bounded rationality. In *Utility and probability*. Springer. 15–18.

Terveen, L. G. 1995. Overview of Human-Computer Collaboration. *Knowledge-Based Systems* 8(2-3).

Truong, A.; Walters, A.; Goodsitt, J.; Hines, K.; Bruss, C. B.; and Farivar, R. 2019. Towards Automated Machine Learning: Evaluation and Comparison of AutoML Approaches and Tools. In *Proc. Intl. Conf. on Tools with Artificial Intelligence*.

Tuggener, L.; Amirian, M.; Rombach, K.; Lörwald, S.; Varlet, A.; Westermann, C.; and Stadelmann, T. 2019. Automated Machine Learning in Practice: State of the Art and Recent Results. In *Proc. Swiss Conf. on Data Science*.

Veale, T. 2013. Creativity as a Web Service: A Vision of Human and Computer Creativity in the Web Era. In *AAAI Spring Symposium Series*.

Ventura, D. 2016. Mere Generation: Essential Barometer or Dated Concept? In *Proc. ICCC*.

Wang, C.; Xu, C.; Yao, X.; and Tao, D. 2019. Evolutionary Generative Adversarial Networks. *IEEE Transactions on Evolutionary Computation* 23(6).

Wiggins, G. A. 2006. Searching for Computational Creativity. *New Generation Computing* 24(3).

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired Image-to-image Translation Using Cycle-consistent Adversarial Networks. In *Proc. ICCV*.