

Chapter 10

Evolutionary Generative Models



João Correia, Francisco Baeta, and Tiago Martins

Abstract In the last decade, generative models have seen widespread use for their ability to generate diverse artefacts in an increasingly simple way. Historically, the use of evolutionary computation as a generative model approach was dominant, and recently, as a consequence of the rise in popularity and amount of research being conducted in artificial intelligence, the application of evolutionary computation to generative models has broadened its scope to encompass more complex machine learning approaches. Therefore, it is opportune to propose a term capable of accommodating all these models under the same umbrella. To address this, we propose the term *evolutionary generative models* to refer to generative approaches that employ any type of evolutionary algorithm, whether applied on its own or in conjunction with other methods. In particular, we present a literature review on this topic, identifying the main properties of evolutionary generative models and categorising them into four different categories: *evolutionary computation without machine learning*, *evolutionary computation aided by machine learning*, *machine learning aided by evolutionary computation* and *machine learning evolved by evolutionary computation*. Therefore, we systematically analyse a selection of prominent works concerning evolutionary generative models. We conclude by addressing the most relevant challenges and open problems faced by current evolutionary generative models and discussing where the topic's future is headed.

J. Correia (✉) · F. Baeta · T. Martins

Department of Informatics Engineering, Centre for Informatics and Systems of the University of Coimbra, University of Coimbra, 3004-531 Coimbra, Portugal

e-mail: jncor@dei.uc.pt

F. Baeta

e-mail: fjrbaeta@dei.uc.pt

T. Martins

e-mail: tiagofm@dei.uc.pt

10.1 Introduction

In recent years, the growth of interest in artificial intelligence research and development has sparked the development of increasingly more complex systems in many fields. In particular, generative models have seen widespread use in the fields of machine learning and evolutionary computation for their ability to generate new instances that follow the probabilistic distribution of a set of pre-existing ones. Due to their recent advancements and impressive results, generative models have become a hot topic. Their outputs can be diverse and applied to a broad range of application domains, e.g. image, music, text, engineering and game design.

Within generative machine learning, the proposal of generative adversarial neural networks (GANs) in 2014 by Goodfellow et al. [72] dramatically changed the landscape in research on generative models by proposing an effective adversarial way of training them. Consequently, GANs quickly gained acclamation to the extent of being coined as “the most interesting idea in the last 10 years in Machine Learning” by Yann LeCun. However, before the advent of GANs, evolutionary computation was the *de facto* approach employed in generative modelling [171]. With the rapid surge of interest around GANs and other deep learning generative approaches, most machine learning-based approaches surpassed what most evolutionary computation generative approaches and models have achieved. This was the case until the last few years when a paradigm shift happened propelled by the maturity of the field of evolutionary machine learning, where methods from evolutionary computation and machine learning are combined as a unified solution.

This chapter surveys the work on evolutionary generative models. We propose this term to identify generative models that use evolutionary computation in any part of the generative process. This definition of evolutionary generative model means that a substantial body of the surveyed work includes plain evolutionary computation approaches that date back to the 1980s, although authors such as Richard Dawkins and Karl Sims have never described their work as evolutionary generative models. Given the increasing adoption of evolutionary machine learning techniques by the research community as well as the rise of the popularity of generative models, the study of evolutionary generative models is ever more frequent in the literature. Despite this, to the best of our knowledge, there is still no comprehensive review focussed on the existing literature for evolutionary generative models. Therefore, in this chapter, we survey the existing literature related to this type of model and analyse existing work from both a historical and scientific point of view.

During the survey, we found that the evolutionary generative model can be divided into four different categories based on the role of machine learning in the generative process: *evolutionary computation without machine learning*, *evolutionary computation aided by machine learning*, *machine learning aided by evolutionary computation*, and *machine learning evolved by evolutionary computation*. The division along the four categories is central to the structure and narrative of this chapter.

The document outline is as follows. In Sect. 10.2, we introduce the core concepts and definitions regarding evolutionary generative models and outline the main properties of these models, thus providing a knowledge base for the chapter and

defining its scope. Then, in Sect. 10.3, we propose a taxonomy to classify evolutionary generative models. Next, in Sect. 10.4, we provide a brief historical overview to contextualise research on this type of generative model, describing the most relevant works and presenting a visual timeline. The following four Sects. 10.5, 10.6, 10.7 and 10.8, analyse the collected papers of the four defined categories. Then, in Sect. 10.9, we overview open problems in the area of evolutionary generative models and identify new research opportunities. Finally, Sect. 10.10 lays the closing remarks by enumerating the main contributions, opportunities and the conclusion of this chapter.

10.2 Fundamentals

We define an *evolutionary generative model* as a generative model that employs evolutionary computation in any part of the generative process. Although the *evolutionary* part of this definition is explicit, we believe that *generative model* may have different interpretations.

When we look at the broader concept of generative model and generative modelling, we can encounter different definitions. For instance, in machine learning, generative models are considered models capable of generating new data instances. More formally, given a set of data instances X and a set of labels Y , generative models capture the joint probability $p(X, Y)$, or just $p(X)$ if there are no labels. Therefore, a generative model describes how a dataset is generated in terms of a probabilistic model that can create instances from a distribution similar to the training data instances. In statistics, a generative model is a statistical model of the joint probability distribution $p(X, Y)$ of a given observable variable X and target variable Y . In graphic design, a generative process is concerned with the creation of an algorithm, or model that produces multiple designs that instantiate a concept. Considering these definitions and our analysis of varied works in this area, we can define a common ground in what concerns the specifications that characterise generative models.

A generative model typically implements an algorithmic process that generates outputs that serve a certain objective or concept. This way, the generated outputs tend to follow a given distribution within an objective class, thus forming a space where the outputs share common characteristics. In addition, the generative model can often receive input information that influences the generative process and consequently changes the outputs. This generative process can be implemented using evolutionary computation or enhanced by it.

During our analysis of existing evolutionary generative models, we identified five properties which we consider relevant to study in each model: application domain, evolutionary algorithm, population, representation and fitness. These 1 properties are outlined in Fig. 10.1 and described in the following paragraphs.

Domain corresponds to the application domain of the model being analysed: *image*, *music*, *text* or *other* applications such as engineering, architecture and even less common disciplines such as dance [59] and culinary recipes [152]. It can be

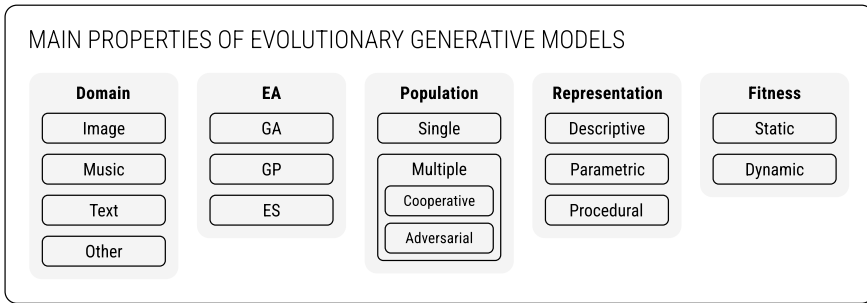


Fig. 10.1 Main properties of evolutionary generative models

argued that this property pertains to the applications of the model and not to the model itself. However, as far as evolutionary computation is concerned, we found that most algorithms follow specific representations and parameterisations according to the application domain.

Evolutionary Algorithm establishes which evolutionary paradigm is employed by the evolutionary generative model, namely, genetic algorithms, genetic programming or evolutionary strategies.

Population relates to the fact of the generative model having a *single* or *multiple* populations of individuals being evolved. Within approaches with *multiple* populations, we have two different types: *cooperative* and *adversarial*. In a *cooperative* setup, multiple populations collaborate to achieve a common goal, where typically each population specialises in a specific task or sub-problem, and their combined efforts contribute to the overall solution. On the other hand, in an *adversarial* setup, different populations are evolved and compete with each other.

Representation refers to the structure of the genotype. Based on the categorisation of evolutionary art systems by Machado [123], we generalised the categorisation and divided the representations used in the analysed works into three main types: *descriptive*, *parametric* and *procedural*. In a *descriptive* representation, we evolve values that directly define the properties of each individual. In a *parametric* representation, we also evolve values, but they are used as input parameters for a model that generates each individual. This way, individuals are defined explicitly by the parametric model and the properties of these individuals are controlled by varying the parameter values of the model, which are encoded in the genotype. In a *procedural* representation, on the other hand, we evolve a sequence of procedures or rules, which can be seen as a model that is then used to generate the individual. In summary, while *descriptive* and *parametric* representations are data oriented, as they encode values, *procedural* representations are process oriented, since the evolutionary algorithm evolves, for instance, mathematical expressions, networks or rules.

Fitness evaluation in the generative process is crucial, and according to the literature, there are several ways of doing it and it depends on many factors. However, in this chapter, we are mainly interested in whether the fitness evaluation changes throughout the evolutionary process or not. Thus, we divided the surveyed works

into two categories: *static* and *dynamic*. In the case of *static* fitness, if the individual is the same, no change should occur to its fitness throughout the evolutionary process. As for *dynamic* fitness, we consider cases where the output of the evaluation may vary during the evolutionary process, i.e. when different fitness values can be assigned to the same individual in different moments of the evolutionary process. Among the approaches that implement dynamic fitness, we can find, for instance, those that employ user feedback to guide the evolutionary process, i.e. interactive evolutionary computation approaches. We consider that fitness assignment based on user feedback, i.e. interactive evolutionary computation, is *dynamic*.

All these properties helped to structure and sort the analysed works, as can be seen later in this chapter, in Tables 10.1 and 10.4, which contain the analysed evolutionary generative models.

Table 10.1 Examples of evolutionary generative models categorised as *evolutionary computation without machine learning*

Domain	Year	Authors	EA	Population	Represent.	Fitness
Image	1986	[52] Dawkins	GA	Single	Procedural	Dynamic
Image	1991	[185] Sims	GP	Single	Procedural	Dynamic
Image	1991	[187] Smith	GA	Single	Procedural	Dynamic
Image	1992	[199] Todd and Latham	GP	Single	Procedural	Dynamic
Image	1993	[11] Baker and Seltzer	GA	Single	Descriptive	Dynamic
Image	1993	[186] Sims	GP	Single	Procedural	Dynamic
Image	1995	[74] Graf and Banzhaf	GA	Single	Descriptive	Dynamic
Image	1996	[5] Angeline	GA	Single	Parametric	Dynamic
Image	1996	[214] World	GP	Single	Procedural	Dynamic
Image	1997	[157] Nishio et al.	GA	Single	Parametric	Dynamic
Image	1998	[76] Greenfield	GP	Single	Procedural	Dynamic
Image	1999	[14] Bentley and Kumar	GA/GP	Single	Descrip./Proced.	Static
Image	1999	[207] Unemi	GP	Single	Procedural	Dynamic
Image	2000	[77] Greenfield	GP	Single	Procedural	Dynamic
Image	2000	[104] Lewis	GA	Single	Procedural	Dynamic
Image	2000	[124] Machado and Cardoso	GP	Single	Procedural	Dynamic
Image	2001	[31] Chapuis and Lutton	GP	Single	Procedural	Dynamic
Image	2002	[125] Machado and Cardoso	GP	Single	Procedural	Dynamic
Image	2002	[172] Rooke	GP	Single	Procedural	Dynamic
Image	2003	[7] Aupetit et al.	GA	Single	Parametric	Dynamic
Image	2004	[78] Greenfield	GP	Multi. (advers.)	Procedural	Dynamic
Image	2005	[55] DiPaola	GP	Single	Procedural	Static
Image	2005	[58] Draves	GA	Single	Procedural	Dynamic
Image	2005	[79] Greenfield	GA	Single	Parametric	Static
Image	2006	[6] Ashlock	ES	Single	Parametric	Static
Image	2007	[56] DiPaola and Gabora	GP	Single	Procedural	Static
Image	2007	[80] Hart	GP	Single	Procedural	Dynamic/Static

(continued)

Table 10.1 (continued)

Domain	Year	Authors	EA	Population	Represent.	Fitness
Image	2008	[35] Collomosse	GA	Single	Parametric	Dynamic
Image	2008	[129] Machado and Graça	GP	Single	Procedural	Dynamic
Image	2008	[154] Neufeld et al.	GP	Single	Procedural	Static
Image	2008	[208] Ventrella	GA	Single	Procedural	Dynamic
Image	2010	[131] Machado et al.	GP	Single	Procedural	Dynamic/Static
Image	2011	[24] Brown et al.	GA	Single	Parametric	Static
Image	2011	[169] Reynolds	GP	Single	Procedural	Static
Image	2012	[15] Bergen and Ross	GA	Single	Descriptive	Dynamic/Static
Image	2012	[37] Colton	GA	Single	Parametric	Static
Image	2012	[54] den Heijer and Eiben	GP	Single	Procedural	Static
Image	2012	[75] Greenfield	GP	Single	Procedural	Static
Image	2012	[132] Machado and Pereira	GA	Single	Parametric	Dynamic
Image	2013	[16] Bergen and Ross	GP	Single	Procedural	Static
Image	2014	[126] Machado and Correia	GP	Single	Procedural	Dynamic/Static
Image	2016	[170] Rodrigues et al.	GA	Single	Parametric	Dynamic
Image	2018	[121] Maças et al.	GA	Single	Parametric	Dynamic
Image	2018	[217] Zoric and Gambäck	GA	Single	Procedural	Static
Image	2020	[50] Cunha et al.	GA	Single	Parametric	Dynamic
Image	2020	[161] Parente et al.	GA	Single	Parametric	Dynamic
Image	2020	[167] Rebelo et al.	GA	Single	Parametric	Static
Image	2021	[9] Baeta et al.	GP	Single	Procedural	Static
Music	1991	[91] Horner and Goldberg	GA	Single	Procedural	Static
Music	1994	[18] Biles	GA	Multi. (colab.)	Descriptive	Dynamic
Music	1994	[102] Laine and Kuuskankare	GP	Single	Procedural	Static
Music	1994	[92] Horowitz	GA	Single	Parametric	Dynamic
Music	1994	[188] Spector and Alpern	GP	Single	Procedural	Static
Music	1995	[95] Jacob	GA	Single	Descriptive	Dynamic
Music	1999	[196] Thywissen	GA	Single	Parametric	Dynamic
Music	2000	[151] Moroni et al.	GA	Single	Descriptive	Static
Music	2002	[53] de la Puente et al.	GP	Single	Procedural	Static
Music	2003	[69] Gartland-Jones	GA	Single	Parametric	Static
Music	2003	[138] Manaris et al.	GP	Single	Descriptive	Static
Music	2007	[19] Bilotta et al.	GA	Single	Parametric	Static
Music	2007	[98] Khalifa et al.	GA	Single	Descriptive	Static
Music	2009	[168] Reddin et al.	GP	Single	Procedural	Static
Music	2010	[165] Prisco et al.	GA	Single	Descriptive	Static
Music	2010	[184] Shao et al.	GP	Single	Procedural	Dynamic
Music	2011	[57] Donnelly and Sheppard	GA	Single	Descriptive	Static
Music	2011	[147] McDermott and O'Reilly	GP	Single	Procedural	Static
Music	2012	[61] Eigenfeldt and Pasquier	GA	Single	Descriptive	Dynamic
Music	2012	[97] Kaliakatsos-Papakostas et al.	GP	Single	Procedural	Dynamic

(continued)

Table 10.1 (continued)

Domain	Year	Authors	EA	Population	Represent.	Fitness
Music	2012	[122] MacCallum et al.	GP	Single	Procedural	Dynamic
Music	2012	[164] Pirnia and McCormack	GP	Single	Procedural	Static
Music	2015	[85] Hofmann	GP	Single	Procedural	Static
Music	2015	[101] Kunimatsu et al.	GP	Single	Procedural	Static
Music	2015	[115] Loughran et al.	GP	Single	Procedural	Static
Music	2015	[193] Sulyok et al.	GP	Single	Procedural	Static
Music	2016	[84] Hickinbotham and Stepney	GP	Single	Procedural	Dynamic
Music	2016	[116] Loughran et al.	GP	Single	Procedural	Static
Music	2016	[119] Loughran and O'Neill	GP	Single	Procedural	Static
Music	2016	[180] Scirea et al.	GA	Multi. (colab.)	Descriptive	Static
Music	2017	[117] Loughran and O'Neill	GP	Single	Procedural	Dynamic
Music	2017	[197] Ting et al.	GA	Single	Descriptive	Static
Music	2018	[159] Olseng and Gambäck	GA	Single	Descriptive	Static
Music	2020	[4] Alvarado et al.	GA	Single	Descriptive	Static
Music	2020	[166] Prisco et al.	GA	Single	Descriptive	Static
Music	2021	[3] Albarracín-Molina et al.	GP	Single	Procedural	Static
Music	2021	[174] Santos et al.	GA	Single	Parametric	Static
Music	2021	[93] Hui Yap et al.	GA	Single	Descriptive	Static
Music	2022	[142] Martínez-Rodríguez	ES	Single	Descriptive	Static
Text	2000	[140] Manurung et al.	GP	Single	Procedural	Static
Text	2003	[139] Manurung	GP	Single	Procedural	Static
Text	2006	[150] Montero and Araki	GA	Single	Descriptive	Dynamic
Text	2007	[82] Hervás et al.	GA	Single	Procedural	Static
Text	2012	[141] Manurung et al.	GA	Single	Procedural	Static
Text	2013	[70] Gervás	GP	Single	Procedural	Static
Text	2015	[202] Tomasic et al.	GA	Single	Procedural	Static
Other	1996	[177] Schnier and Gero	GA	Single	Procedural	Static
Other	2008	[62] Ekárt	GP	Single	Procedural	Static
Other	2008	[66] Frade et al.	GP	Single	Procedural	Dynamic
Other	2010	[26] Browne and Maire	GP	Single	Procedural	Static
Other	2010	[28] Byrne et al.	GP	Single	Procedural	Dynamic
Other	2012	[183] Shaker et al.	GP	Single	Procedural	Static
Other	2014	[29] Cardona et al.	ES	Single	Descriptive	Static
Other	2015	[87] Hoover et al.	GA	Single	Procedural	Dynamic
Other	2016	[114] Lopes et al.	GA	Single	Descriptive	Static
Other	2017	[40] Cook et al.	GA/GP	Multi. (colab.)	Descrip./Proced.	Static
Other	2017	[120] Lucas and Martinho	GA	Single	Descriptive	Static
Other	2017	[153] Muehlbauer et al.	GP	Single	Procedural	Static
Other	2018	[39] Cook and Colton	GA/GP	Multi. (colab.)	Descrip./Proced.	Static

10.3 Taxonomy

We propose a taxonomy to classify evolutionary generative models into four categories according to the existence and role of machine learning in the generative model. The proposed categories are described next.

Evolutionary computation without machine learning—systems where the generative model only uses evolutionary computation with no intervention of any kind of machine learning technique. The modelling is purely based on evolutionary computation and the evaluation typically involves user guidance, hardwired fitness functions or a set of rules and conditions.

Evolutionary computation aided by machine learning—systems that combine evolutionary computation and machine learning, where the utilisation of machine learning enhances the capability of the evolutionary generative model. For instance, using the activation of a machine learning classifier to assign fitness and guide evolution towards outputs that maximise the classifier activation or using a clustering technique to aid in the selection of evolved individuals.

Machine learning aided by evolutionary computation—systems which are machine learning-based and where evolutionary computation is employed to enhance the capability of the machine learning part. For instance, neuroevolution approaches fall into this category; the same applies to approaches that explore the latent space of a GAN to generate suitable outputs or use evolutionary computation instead of gradient optimisation to train or fine-tune generative machine learning models.

Machine learning evolved by evolutionary computation—systems in which evolutionary computation evolves entire machine learning generative models or entire machine learning components. For instance, neuroevolution of augmenting topologies (NEAT) evolves artificial neural networks and their components using a genetic algorithm or the evolution of populations of GANs.

10.4 Historical Overview

This section provides a brief historical overview of key publications in each of the four proposed categories for evolutionary generative models. In particular, in each of the following subsections, we summarise the publication timeline of each category while presenting a table containing several examples and their main properties. Figure 10.2 presents a timeline for the four proposed categories.

10.4.1 Evolutionary Computation Without Machine Learning

This category of evolutionary generative models is the one with the highest volume of publications (please refer to Fig. 10.2), which is explained by the fact that before the popularisation of GANs and neuroevolution, evolutionary computation approaches represented the early instances of evolutionary generative models.

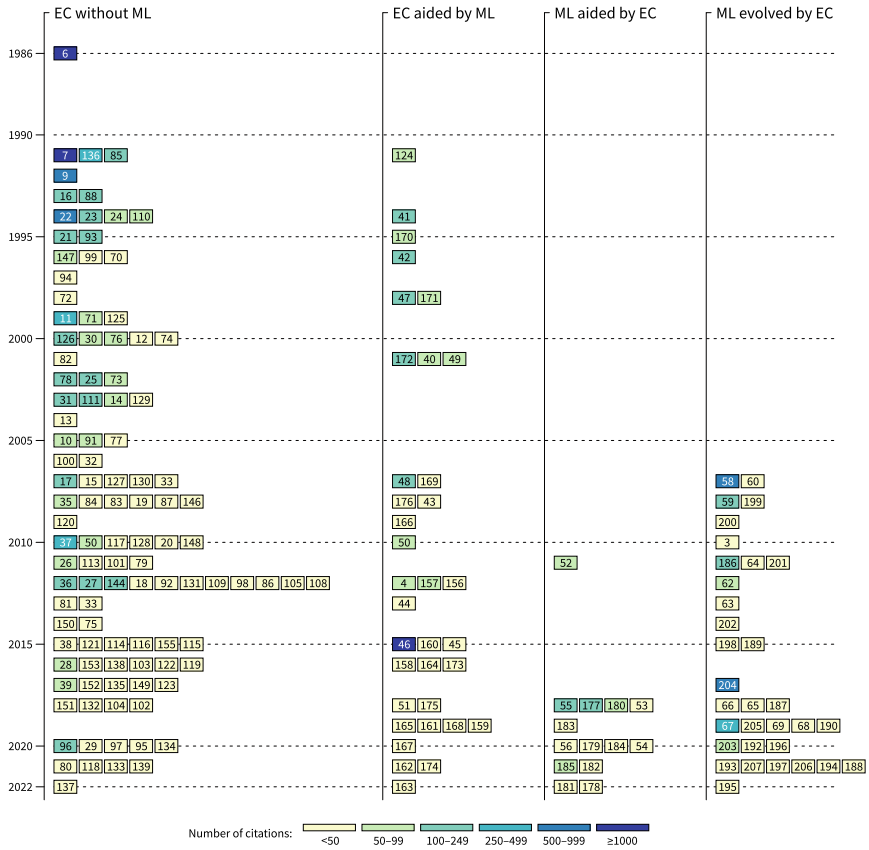


Fig. 10.2 Timeline showing all publications on evolutionary generative models cited in this chapter. Publications are ordered chronologically (oldest at the top and most recent at the bottom) and divided horizontally by category (from left to right: *evolutionary computation without machine learning*, *evolutionary computation aided by machine learning*, *machine learning aided by evolutionary computation* and *machine learning evolved by evolutionary computation*). Each publication is represented with a rectangle containing the corresponding reference number. The colour of each rectangle indicates the number of citations of the corresponding publication according to the scale shown at the bottom of the figure (darker colours indicate a greater number of citations). The number of citations was queried to Google Scholar on June 7, 2023

The biomorphs presented by Dawkins [52] in his book *The Blind Watchmaker*, in 1986, were one of the earliest examples of using evolutionary computation to create images using user interaction, basically setting the start for interactive evolutionary computation. Afterwards, in 1991, Sims [185] employed user-guided evolution to create images, solid textures and animations encoded as Lisp-based expressions of mathematical functions, variables and constants, also referred to as symbolic expressions. The idea of using symbolic expressions in evolutionary algorithms had previously been explored by Koza [100] in 1990 and would eventually give rise to

the nowadays prolific field of genetic programming. Despite gaining popularity later on, the approach of using evolutionary computation for image generation was not yet common in the computer science community at this point in time (see Fig. 10.2). That would change, however, in 1992 with the work by Todd and Latham [199], who popularised evolutionary art in general with their work on the generation of 2D and 3D images and by making it available to the general public. In line with this idea, in 1999, Draves [58] created the collaborative abstract artwork “Electric Sheep” that evolved fractal flames, a generalisation and refinement of the iterative function systems. Several researchers were experimenting with varied representations for evolutionary algorithms. Bentley and Kumar [14] explored the importance of different embryogenesis processes, i.e. the mapping from the genotype representation to phenotype, in the same generative design problem demonstrating the impact of having a direct genotype on phenotype mapping on the final solutions and evolution with distinct solutions using a genetic algorithm or genetic programming.

Given the impact and amount of publications regarding evolutionary computation for image generation, we can conclude that the first half of the 1990s sparked more interest in evolutionary art in general following its appearance. This interest was eventually picked back up around the turn of the millennium, with many works in evolutionary computation being published (see Fig. 10.2). Within this period, it is worth mentioning the work by Lewis [104] concerning aesthetic evolutionary design with network data flows, a direct acyclic graph representation of transformations on geometric primitives for generating specific types of images such as cartoon faces and human figure geometry. Greenfield [78] used the image generation approach of Sims [185] with a bio-inspired coevolutionary fitness scheme, inspired by hosts and parasites dynamics, to instantiate an evolutionary simulation for the generation of aesthetic image tiles. Aupetit et al. [7] created an interactive genetic algorithm to evolve parameters for an ant colony optimisation approach to generate paintings. Hart [80] proposed new mutation operators for expression-based evolutionary image generation and tree alignments schemes for improving the morphing between two individuals, thus enabling a smoother animation process and expanding the possible solutions for the cross-dissolve process between individuals [186]. Genetic programming was used extensively around this time, with a few more notable examples to be mentioned. Di Paola and Gabora [56] tried to evolve a portrait of Charles Darwin using genetic programming, an endeavour that presented many hurdles, such as handling fitness values plateaus which proved difficult to overcome. The work by den Heijer and Eiben [54] used a scalable vector graphics format as a genetic representation of their genetic programming approach for the evolution of abstract images. Outside of expression-based image generation, we should mention the work by Colomesse [35], which performed non-photorealistic rendering by generating images with pre-processed and parameterised brush strokes based on a starting image. To the best of our knowledge, although many genetic programming approaches have done it before [185], this work represents one of the first efforts in using genetic algorithms to evolve filters from existing images. In 2010, Machado et al. [131] presented an evolutionary engine that allows the evolution of context-free grammars using a graph representation and operators.

The early use of genetic algorithms in the generation and evolution of music was explored by Jacob [95], who employed an interactive system to assign fitness in music composition, and by Biles [18], who introduced the framework Genjam to generate Jazz solos. It is important to note that these first studies in music generation mainly focussed on the evolution of melodic progressions. One of the first studies specifically targeting metric and rhythm was performed by Horowitz [92] in 1994. Furthermore, around this period, the use of expression-based methods to create music was introduced by Spector and Alpern [188] with a system to create Bebop music using genetic programming. These first efforts were later picked up and extended upon, which saw the introduction of new frameworks and techniques. It is also worth mentioning the work by de la Puente et al. [53] in the early 2000s, which, to the best of our knowledge, corresponds to the first application of a grammatical evolution algorithm to music composition. Moreover, Donnelly and Sheppard [57] proposed the use of a genetic algorithm to evolve four-part melodic harmonies, and frameworks such as DarwinTunes [122] were introduced using genetic programming tree-like representation to represent full songs. In addition to this publication, it is important to mention MetaCompose [180], a framework still used nowadays, consisting of a graph-based chord sequence generator, as well as a melody and accompaniment generator. More recently, a system that has been gaining interest is Evocomposer [166]. The novelty of this framework is the hybrid evaluation function capable of multi-objective optimisation to solve a four-voice harmonisation problem, i.e. given a music line as input, the algorithm must generate the remaining three accompanying voices.

Concerning the domain of text generation, the body of existing research seems to be more sparse. At the beginning of the millennium, Manurung et al. [140] presented the first efforts in applying evolutionary techniques to the development of an English poetry generation system using genetic programming. Manurung [139] would later extend upon this research in poetry generation, this time employing grammatical evolution. Although poetry seems to be the preferred text type for generative systems, it is worth highlighting pioneers in other genres, such as the work by Montero and Araki [150] in 2006, who has worked on the generation of dialogue in natural language. Also in the mid-2000s, Hervás et al. [82] proposed an evolutionary approach to assist in the generation of alliterative text that preserves the meaning of a given input sentence. Gervás [70] used an evolutionary algorithm coupled with his “wishful automatic Spanish poet” (WASP) system for poetry generation.

Lastly, beyond the image and music domains, we should also note the integration of evolutionary techniques with other types of domains of applications throughout the timeline. It is the case of the work by Frade et al. [66], which uses genetic programming for the evolution of realistic video game terrain. Additionally, the work of Shaker et al. [183] used grammatical evolution for the generation of *Super Mario Bros.* levels. Aside from the efforts of creating gaming assets, Browne and Maire [26] presented an approach to evolve interesting game designs from a human perspective with the Ludi framework. Still, in the domain of game design, the work by Hoover et al. [87] represents one of the first systems using evolutionary techniques

to develop multiple aspects of a level, including visuals, audio and gameplay, along with the work by Cook et al. [40] with the ANGELINA framework.

10.4.2 Evolutionary Computation Aided by Machine Learning

After summarising the timeline of evolutionary generative models that use solely evolutionary computation, we now move on to evolutionary generative models that, in addition to evolutionary computation, also use machine learning on any part of the generative process.

The machine learning approaches have aided evolutionary generative models mostly in two aspects: fitness assignment or auxiliary representations for selecting individuals. Starting with the latter, Saunders and Gero [176], in the early 2000s, proposed a system that used a self-organising map to categorise each generated image and acted as a short-term memory of their agent-based and evolutionary generative model. This is one of the first examples of using an auxiliary representation to help the evolutionary process. Nevertheless, based on our collection of works, most of the work done on evolutionary generative models aided by machine learning mostly uses machine learning to evaluate or assist in evaluating generated individuals. Until the early 1990s, the evolution of images had been done by resorting to user interaction to assign fitness to individuals, i.e. an interactive evolutionary computation approach. Albeit effective, interactive fitness assignment is tied to problems such as user fatigue or inconsistent evaluation, and it is hard to scale. The work by Baluja et al. [12] aimed to change this paradigm by proposing an automatic fitness assignment scheme based on the evaluation of images using machine learning, more precisely, an artificial neural network, to automate the evolutionary process. However, this approach only had partial success, the main problem being the employed artificial neural network was not able to generalise and distinguish between different visually striking image features and properly guide the evolutionary process. Most works at that time tended to use interactive fitness assignment schemes or opted for a semi-automatic form with the help of machine learning approaches, as is the case of the incorporation of an artificial neural network to aid fitness assignment in the enhanced GenJam framework proposed by Biles et al. [17].

In 2008, Machado et al. [133] experimented with automatic fitness assignment using artificial neural networks, similar to the work by Baluja et al. but promoting an adversarial scenario between an artificial neural network and the evolutionary computation approach. The artificial neural network was trained to discriminate among images previously generated by the evolutionary approach and famous paintings. Then the evolutionary approach was used to generate images that the artificial neural network classifies as paintings. The images created throughout the evolutionary run were added to the training set, and the process was repeated. The iterative process led to the refinement of the classifier forcing the evolutionary approach to explore different types of images after each refinement. Based on this work, Correia et al. [41] used

a machine learning approach trained with multiple object detectors to evolve figurative images consistently, i.e. images that contain a certain object according to the machine learning approach. Although different types of images generated resemble the object, some of the images evaluated by the machine learning approaches yielded high confidence but did not resemble it or were exploits of the machine learning approaches that did not correspond to the intended object. The work was later extended by Machado et al. [134], generating images that were ambiguous from a human and machine learning approach perspective. Following up on the idea of having fitness assignment schemes with machine learning approaches and their exploits, the work by Nguyen et al. [156] demonstrated how easily state-of-the-art machine learning approaches based on deep artificial neural networks can be fooled, i.e. classify images with high confidence of belonging to a certain class, when in reality the images do not resemble that class from a human perspective. The authors have shown that it is possible while using a simple genetic algorithm with a direct encoding or using compositional pattern-producing networks (CPPNs) to generate images that were unrecognisable by humans but recognised by deep neural networks with high confidence. Despite its exploits, machine learning intervention in fitness assignment schemes proliferated and is still one of the most used approaches for automating the fitness assignment process.

The idea of using machine learning to assist with fitness assignments was also used in other types of applications. For sound generation, the work by Johanson and Poli [96] presented the generation of music with automatic fitness using an artificial neural network to rate the generated music. Manaris et al. [137] proposed a hybrid approach that saw the combination of evolutionary techniques along with machine learning to compose and analyse music. In terms of text generation, akin to Manurung et al. [140], Levy [103] explored evolutionary techniques coupled with machine learning for poetry generation, generation of stories with plot induction by McIntyre and Lapata [148] and Al-Najjar and Hämäläinen [2] worked with the generation of culturally satirical movie titles. As an example of an evolutionary system employing machine learning for other applications, we include the work by Morris et al. [152] concerning evolutionary food recipes using a genetic algorithm and an artificial neural network trained on a starting set of recipes.

10.4.3 Machine Learning Aided by Evolutionary Computation

In this subsection, we shift our focus to evolutionary generative models that are machine learning-based and where evolutionary computation is employed to improve the capability of the machine learning part.

As can be seen in Fig. 10.2, this is the section with the least entries among all the categories. This can be mostly related to the fact that the generative models based on machine learning gained traction around 2014 with the advent of GANs, and most of the researchers working on evolutionary approaches started to explore models that

combined the machine learning models with the evolutionary ones. Before 2014, the work by Manaris et al. [136] introduced in 2011 the Monterey Mirror, a system that uses a genetic algorithm to evolve musical phrases provided by a Markov model.

After 2014, in this category of evolutionary generative models, most publications pertain to latent variable evolution, a term initially coined by Bontrager et al. [22] to refer to the evolutionary exploration of a given latent space. Latent spaces in generative machine learning are intermediate compressed versions of the original inputs, e.g. bottleneck layer of auto-encoders. Thus, these compressed versions, which are typically composed of numerical vectors, encode original samples and by sampling the latent space and decoding via the generative model, we can explore samples that were learnt by the generative model. This exploration by sampling and decoding vectors of the latent space holds potential that has been researched until today. For instance, in the work of Bontrager et al. [22], the authors trained a GAN to generate fingerprints capable of matching a high percentage of users and afterwards explored the generated latent space of the model using covariance matrix adaptation evolution strategy (CMA-ES). In a different setting, Fernandes et al. [64] explored the latent space to evolve groups of diverse images using genetic algorithms and multi-dimensional archive of phenotypic elites (MAP-elites). In the context of a procedural content generation scenario and following similar procedures, Volz et al. [210] generated game levels using latent variable evolution with CMA-ES and a GAN. Moreover, in the work by Schrum et al. [178], instead of directly evolving latent vectors, it evolves parameters for CPPNs that will, in turn, generate latent vectors for GANs.

10.4.4 Machine Learning Evolved by Evolutionary Computation

Lastly, we overview the timeline of the last category of evolutionary generative models, in which evolutionary computation is used to evolve entire machine learning models or entire machine learning components.

A significant instance in the literature of machine learning models evolved by evolutionary computation for generative purposes would be the NEAT framework in 2002 by Stanley and Miikkulainen [191]. NEAT represented one of the first neuroevolution approaches capable of automatically evolving and adjusting the topology and weights of an artificial neural network. Stanley [190] would later extend upon his work by proposing CPPNs and evolving these networks with the NEAT method. Thus, in 2008, the CPPN-NEAT approach was made publicly available in the form of an online evolutionary tool named Picbreeder [181], capable of guiding evolution using an interactive fitness assignment scheme. Furthermore, Dubbin and Stanley [59] generated dance moves using the CPPN-NEAT model.

Around the same time, Togelius et al. [200] explored a competitive coevolutionary generative model for racing car controllers. These early studies regarding controllers led the way to the evolutionary exploration of game design features, which was

mostly set in motion by Togelius et al. [201] with the evolution of controllers to play *Super Mario Bros*. Other works followed, such as the generation of shapes in video games by Liapis et al. [111] and the evolution of gameplay in the work of Cardona et al. [30], using competitive coevolution to evolve agents for both the player and the non-player characters in the classic game *Pac-Man*.

In the realm of music, we mention the work by Bell [13] in the generation of musical compositions by evolving populations of Markov models with a genetic algorithm. Despite this, and perhaps motivated by a saturation of research concerning traditional machine learning methods to improve the performance of GANs, the end of the last decade saw an increase in evolutionary machine learning research mostly based on coevolution. Aside from neuroevolution being central in the same timeline, evolutionary approaches that operate to evolve entire generative models, such as GANs, have a high computational cost. Here the advantages of the optimisation aspect of evolutionary computation were to be explored while minimising the computational cost of an already demanding training process of the state-of-the-art generative machine learning models.

Extending upon the GAN scheme, in 2018, Garciarena et al. [68] presented a method for generating Pareto set approximations of generators and discriminators. In this method, the authors resort to the coevolution of generators and discriminators where a single individual represents both components. In the same year, the Lippizanner framework was proposed by Al-Dujaili et al. [1] as the first system encompassing a spatial approach to coevolution where a grid of cells was used for evolving groups of GANs. Moreover, in 2019, Wang et al. [211] proposed the evolutionary GAN (E-GAN) model, which consisted of the evolution of a population of generators within a GAN model. This approach was later enhanced by Costa et al. [49] with the proposal of coevolutionary GAN, a model based on coevolution where an additional population of discriminators is evolved alongside the population of generators. Additionally, Toutouh et al. [203] extended the Lippizanner framework by incorporating the generator training method from the E-GAN model, thus effectively combining variation operators and population-level approaches to improve the diversity of GANs.

10.5 Evolutionary Computation Without Machine Learning

In the last section, we provided a historical overview and timeline of the works in each category. In this section, as well as in the three following ones, we will cover the collected works on evolutionary generative models. We start by analysing the works that only use evolutionary algorithms. Table 10.1 enumerates a series of evolutionary generative models which we consider to belong to the category *evolutionary computation without machine learning*. The listed publications are grouped by application domain (first column) and then sorted by year (second column) and by authors (third column).

Our analysis begins with applications in the image domain, including shape generation, 3D modelling and video generation, among others. Afterwards, we delve into sound applications, covering everything from sound effects to full-fledged music composition. Next, the most relevant works in text generation are analysed, and we finish the section with other applications which do not fit into the mentioned application domains. Each of the works analysed in each domain is further subdivided by the type of evolutionary algorithm used.

When it comes to using evolutionary computation as an image generative model, the two most common approaches are by far genetic programming and genetic algorithm, with genetic programming taking a slight preference within the literature analysed (see Table 10.1). Most of the approaches that employ genetic programming do so by evolving symbolic expressions that generate images through the process of feeding coordinates as input. This way, to evaluate an individual, each pixel coordinate is fed into the expression that will generate a colour (typically a tuple of values representing RGB channels) of which the output image, i.e. the individual, will be composed of. These approaches have been applied to the generation of a diverse range of image types, including aesthetic imagery [76–78, 80, 126, 172, 185, 186, 207, 214], fractals as iterated function systems [58] or the generation of specific image classes [9, 16, 55, 56, 124, 125, 169]. Other works define the features of an output image iteratively: vector graphics [54], stochastic sequencing of points to generate iterated function systems [31], evolving image filters [154] and the evolution of 2D renders of 3D compositions [129]. Regarding the type of fitness function, for the purposes of image generation with genetic programming, we observe that there are more static than dynamic (see Table 10.1), mostly due to automation of the generative model via hardwired fitness functions.

Akin to genetic programming, genetic algorithms were also used to generate abstract imagery [37, 52, 187]. Aside from image generation, genetic algorithm-based generative systems were used to generate drawings of a specific category by evolving properties of primitives such as coordinates, rotation and size. Line drawings are perhaps the simpler examples of these approaches. Within fractals, Ventrella [208] used a genetic algorithm to explore different configurations of the Mandelbrot set. Baker and Seltzer [11] evolved lines either using line drawings coordinates or by evolving agents that live on a canvas and spread their raster until they die. This aspect of agents having an energy limit to operate was a typical aspect of agent-based approaches [79, 145, 175].

The evolution of vector graphics is also explored in genetic algorithms and evolutionary strategies for image generation [15]. Aside from vectors, similar works made use of other primitives. Examples include examples of evolving images directly bitmaps via transformations and distortions [74], the generation of cartoon faces [104, 157], emojis [50], letterings [161], posters [167], the evolution of renderings based on existing images (non-photorealistic rendering) [35, 132] and the evolution of context-free grammars [131]. Animation is another field where genetic algorithms stand out, with some works evolving fractals [5, 6, 24] and screensavers [58].

Other works demonstrated the interplay between the image and music domains by generating images based on musical input directly [217] or based on systems evolved

using music [170]. Other authors developed meta-models by evolving the parameters of these algorithms [7, 121] or resorted to an indirect way of generating the outputs, e.g. Greenfield [75], although not concerning image generation directly, produced controllers for drawing robots that, in turn, created visually appealing imagery.

Readers interested in a thorough analysis of evolutionary computation for the purposes of image generation are encouraged to read the works by Galanter [67], Romero and Machado [171] as well as Lewis [105].

Akin to image generation, genetic algorithms and genetic programming are again the most prolific techniques in the generation of sound, with genetic algorithms taking a slight preference within the literature reviewed. Concerning genetic programming, many works deal with the generation of musical melodies [97, 164], some of them of a particular style such as Jazz [188]. However, the largest body of research includes both the generation of rhythm along with accompanying melodies [51, 85, 101, 102, 122, 138, 147, 193]. Some of these are music guided towards specific properties of sound and musical score [51], while others focus on the methods behind the compositional process [193]. Other works resort to grammatical evolution for the generation of music that has been applied mainly to musical composition [3, 184], including melody formation [53, 115, 116, 119, 168] as well as live coding [117].

Regarding genetic algorithms, a number of works dealt with the generation of specific parts of music such as the melody [19, 71, 151, 196] and the rhythm [92]. Some studies delved into the problem of harmony production, i.e. given a musical line/voice, the system is tasked with the creation of accompanying lines/voices to have a complete musical piece with chords for each input note [57, 165, 166], with the harmonisation of bass lines being a recurring theme, in particular [165].

However, most works in this section tackle the broader problem of automated musical composition [61, 69, 93, 98, 159, 180], some of these deal with the generation of responses to given musical phrases instead of generating sound from the ground up [18]. It is worth noting that in musical composition, many approaches tackle different problems, such as generating variations from existing pieces, thematic bridging and live coding. The generation of variations deals with the composition of pieces given an existing source [4, 95, 197]. On the other hand, thematic bridging concerns the generation of lines capable of seamlessly transitioning between two given musical phrases [91]. More recently, thematic bridging was also explored using evolutionary strategies in the work by Martínez-Rodríguez [142], who proposed fitness as a weighted neighbourhood average distance metric between a solution and the target. More applications include live coding, which tackles the use of domain-specific languages for generating new musical patterns in a live concert setting [84].

Generally, we can observe that most systems for sound generation based on genetic algorithms and genetic programming implement static fitness assignments, with user-defined fitness being a minority. Still, in music composition, more recent works try to gather inspiration from different domains. An example of this is the work by Santos et al. [174], which bridges the fields of image and music generation by using images as a source of inspiration for the compositional process.

The above-mentioned works illustrate examples from the literature for evolutionary computation in music. Readers who seek a complete analysis of the appli-

cation of evolutionary computation to music should refer to the following surveys by Todd and Werner [198], McCormack [146], Loughran and O'Neill [118] and Eigenfeldt et al. [60].

The body of work analysed regarding the application of evolutionary computation for text generation is rather lacking compared to the domains of image and sound. Nevertheless, the genetic algorithm is by far the most used approach, with applications to the generation of poetry [70, 139–141], dialogue (i.e. natural language generation) [150], as well as alliterative text [82]. In particular, it is worth mentioning the work by Manurung [139], where the evolution of grammars was used to generate poetry. For the more curious reader, a survey by Oliveira [158] analyses automated poetry, which goes beyond the scope of evolutionary computation.

Finally, we list works that one way or another concern the generation of artefacts that do not fit the categories of either images, sound or text. Here it is worth mentioning the automatic evolution of designs for Lace Knitting Stitch Patterns by Ekárt [62], and the modelling of terrain maps by Frade et al. [66], both resorting to genetic programming techniques. Moreover, the works dealing with 3D [28, 177] and shape design [153] are worth noting. Similar to what was verified for sound, text and other applications, most of these applications implement fixed fitness assignment as well. Lastly, another avenue where evolutionary computation has been applied is to aid game design. In this context, we include Ludi, a general game system introduced by Browne and Maire [26] to synthesise and evaluate new games capable of captivating the interest of human players, as well as the use of a (1 + 1) evolutionary strategy for the generation of decks of cards by Cardona et al. [29]. Regarding genetic algorithms, we can mention the works by Cook et al. [40] and Cook and Colton [39] in the automated and cooperative coevolution of game design. Still pertaining to game design, some other examples consist in the co-generation of game levels alongside a human designer [120] as well as the evolution of level design to increase the feeling of in-game tension [114]. Refer to the survey of Liapis [109] for a thorough analysis of the applications of evolutionary computation and procedural content generation, among other techniques to the generation of game design.

In terms of representation used in the analysed evolutionary generative models, we found instances of the different types of representation, with a clear predominance of procedural representations (70 procedural, 23 descriptive and 19 parametric representations).

10.6 Evolutionary Computation Aided by Machine Learning

This section includes approaches where models are inherently evolutionary, but their functionality or performance is improved with machine learning approaches. A common example is an approach where the evolutionary model uses machine learning in

their fitness assignment or one of the components of the evolutionary computation model is replaced by machine learning. Table 10.2 presents examples of evolutionary generative models, which we consider to belong to the category *evolutionary computation aided by machine learning*. The listed publications are grouped by application

Table 10.2 Examples of evolutionary generative models categorised as *evolutionary computation aided by machine learning*

Domain	Year	Authors	EA	Population	Represent.	Fitness
Image	1994	[12] Baluja et al.	GP	Single	Procedural	Static
Image	1994	[176] Saunders and Gero	GP	Multi. (colab.)	Procedural	Dynamic/Static
Image	2008	[36] Colton	GA	Single	Parametric	Dynamic
Image	2008	[133] Machado et al.	GP	Single	Procedural	Dynamic
Image	2009	[32] Chen et al.	GA	Single	Parametric	Static
Image	2012	[127] Machado et al.	GP	Single	Procedural	Static
Image	2012	[128] Machado et al.	GP	Single	Procedural	Dynamic
Image	2013	[41] Correia et al.	GP	Single	Procedural	Static
Image	2015	[134] Machado et al.	GP	Single	Procedural	Static
Image	2015	[143] Martins et al.	GA	Multi. (colab.)	Descriptive	Static
Image	2015	[156] Nguyen et al.	GA	Single	Procedural	Static
Image	2016	[44] Correia et al.	GA	Single	Parametric	Static
Image	2016	[209] Vinhas et al.	GP	Single	Procedural	Static
Image	2019	[42] Correia et al.	GP	Single	Procedural	Dynamic
Image	2019	[43] Correia et al.	GA	Single	Parametric	Static
Image	2019	[144] Martins et al.	GA	Single	Procedural	Static
Image	2019	[195] Tanjil and Ross	GP	Single	Procedural	Static
Image	2020	[113] Lopes et al.	GA	Single	Procedural	Dynamic/Static
Image	2021	[45] Correia et al.	GP	Single	Procedural	Static
Image	2022	[10] Baeta et al.	GP	Single	Procedural	Dynamic
Music	1991	[71] Gibson and Byrne	GA	Single	Descriptive	Static
Music	1995	[189] Spector and Alpern	GP	Single	Procedural	Static
Music	1996	[17] Biles et al.	GA	Single	Descriptive	Static
Music	1998	[27] Burton and Vladimirova	GA	Single	Descriptive	Dynamic
Music	1998	[96] Johanson and Poli	GP	Single	Procedural	Dynamic/Static
Music	2001	[162] Pearce and Wiggins	GA	Single	Descriptive	Static
Music	2007	[137] Manaris et al.	GP	Single	Procedural	Dynamic
Music	2007	[163] Phon-Amnuaisuk et al.	GP	Single	Procedural	Static
Music	2016	[94] Ianigro and Bown	GA	Single	Procedural	Dynamic
Text	2001	[103] Levy	GA	Single	Descriptive	Static
Text	2010	[148] McIntyre and Lapata	GA	Single	Procedural	Static
Text	2018	[2] Al-Najjar and Hämäläinen	ES	Single	Descriptive	Static
Text	2021	[213] Winters and Delobelle	GA	Single	Descriptive	Static
Other	2012	[152] Morris et al.	GA	Single	Procedural	Static
Other	2018	[108] Liapis	GA	Single	Parametric	Static

domain (first column) and then sorted by year (second column) and by authors (third column).

Similar to Sect. 10.5, the use of genetic programming in this section mainly pertains to the generation of specific image classes/types [41, 127, 128, 134, 143, 195, 209] or to aesthetic imagery [12, 42]. In addition to this, genetic programming methods for the transformation of existing images have also been presented, such as the evolution of image pipelines by Correia et al. [45]. Machado et al. [133] and Correia et al. [42] explored the stylistic change in evolutionary art in a framework similar to GANs using an adaptive classifier along with a genetic programming engine. Similarly, Baeta et al. [10] proposed another example of a system having a competition between isolated components, where the generator of a standard deep convolutional GAN is replaced by a genetic programming approach.

Regarding genetic algorithms for the image domain, examples of machine learning for the fitness assignment phase include the use of artificial neural networks either by evolving false positive images [156], false negatives [44] or even to perform data augmentation [32, 43]. Lopes et al. [113] and Martins et al. [144] also employed a genetic algorithm guided by a machine learning model to evolve vector graphics, and stencils for typefaces, respectively. Similar to the work by Bergen and Ross [15] (see previous section), both modalities of fitness function (static and dynamic) were explored, the latter as a separate experiment and the former by mixing a static target function with user-defined options. Graph evolution is another example of the application of genetic programming to image generation, as exemplified by Vinhas et al. [209].

Concerning genetic programming, there are examples for the generation of melodies [96, 163], including Jazz [188], and the generation of rhythm with accompanying melodic tracks, explored by Manaris et al. [137] as well as Spector and Alpern [189]. Using genetic algorithms, many works dealt with the generation of specific parts of music such as melody [71] and rhythm [27]. Other works focussed solely on harmony generation [162], i.e. musical responses, instead of generating sound from scratch. Like in the category *evolutionary computation without machine learning*, automated music composition is also addressed in this section with the work of Biles et al. [17], who used artificial neural networks for fitness assignment. Although almost all examples of the application of genetic algorithms to music generation dealt with music, the broader exploration of audio, in general, was addressed by Ianigro and Bown [94].

Relating to text generation, a genetic algorithm was used by Levy [103] for the purposes of poetry generation with an artificial neural network as a measure of adaptive fitness. Other examples include the study of satire generation by Winters and Delobelle [213] and by Al-Najjar and Hämäläinen [2]. Finally, examples of other applications include the works by Liapis [108] for the evolution of colour palettes, the evolution of scene elements by Colton [36] and the introduction of a system for generating novel culinary recipes by Morris et al. [152].

In this section, most of the analysed works in terms of representation have representativity of the three types of representation but with a prevalence of the procedural ones (20 procedural, 9 descriptive and 6 parametric representations).

10.7 Machine Learning Aided by Evolutionary Computation

In this section, we include the approaches where evolutionary computation is used to enhance the functionality or performance of the machine learning model. As an example, latent variable evolution belongs in this section because these approaches use evolutionary computation to explore a latent space previously built using a machine learning model. Table 10.3 contains examples of evolutionary generative models which we consider to belong to the category *machine learning aided by evolutionary computation*. The listed publications are grouped by application domain (first column) and then sorted by year (second column) and by authors (third column).

The idea of latent variable evolution was introduced by Bontrager et al. [22, 23] to explore the latent space of inputs to the generator network of a GAN using a CMA-ES approach to generate fingerprints that maximise the number of imposter matches. Concerning genetic algorithms, Schrum et al. [178] evolved parameters for CPPNs, which in turn generated a variety of different latent vectors for GANs.

Other examples concerning the application of genetic algorithms to latent variable evolution include the work by Grabe et al. [73], who used Gaussian mixture models to explore possible latent vectors to evolve distinct images, along with interactive methods to perform the evolutionary search by Zaltron et al. [215], among others [21, 64, 135]. Outside of latent variable evolution but still pertaining to image generation, Colton [38] also proposed a system to generate image filters using a genetic algorithm with MAP-elites or the work by Korde et al. [99], which uses an evolutionary algorithm to train in the initial iterations to stabilise the weights before using normal optimisation techniques to train GANs.

Table 10.3 Examples of evolutionary generative models categorised as *machine learning aided by evolutionary computation*

Domain	Year	Authors	EA	Population	Represent.	Fitness
Image	2018	[22] Bontrager et al.	ES	Single	Parametric	Static
Image	2018	[21] Bontrager et al.	GA	Single	Parametric	Dynamic
Image	2018	[23] Bontrager et al.	ES	Single	Parametric	Static
Image	2019	[99] Korde et al.	GA	Single	Parametric	Dynamic
Image	2020	[64] Fernandes et al.	GA	Single	Parametric	Static
Image	2020	[178] Schrum et al.	GA	Single	Parametric	Static
Image	2020	[215] Zaltron et al.	GA	Single	Parametric	Dynamic
Image	2021	[38] Colton	GA	Single	Parametric	Static
Image	2021	[110] Liapis et al.	GA	Multi. (colab.)	Procedural	Static
Image	2022	[73] Grabe et al.	GA	Single	Parametric	Static
Image	2022	[135] Machín et al.	GA	Single	Parametric	Static
Music	2011	[136] Manaris et al.	GA	Single	Procedural	Static
Other	2018	[210] Volz et al.	ES	Single	Parametric	Dynamic/Static
Other	2020	[160] O'Reilly et al.	GP	Single	Procedural	Dynamic

Concerning other applications, an approach combining genetic algorithms and Markov chains was proposed by Manaris et al. [136] to evolve musical phrases. Volz et al. [210] worked on the generation of game levels using latent variable evolution with adversarial training. Moreover, the work by O’Reilly et al. [160] in cyberattacks and defence mechanisms using genetic programming in an adversarial setting, among other techniques, explored generating and evolving executable adversarial programs. Lastly, we mention the work of Liapis et al. [110] for the creation of suitable spaceships for arcade-style games using CPPN-NEAT and novelty search.

One of the most noticeable aspects of this section lies in the fact that, when compared to other ones, the body of work concerned with aiding machine learning with evolutionary computation is significantly less, especially when compared to Sect. 10.5. Moreover, in terms of representation, we can observe that most approaches are parametric, with just a few being procedural (in total, there are 3 procedural, 11 parametric and no descriptive representations). This is because the analysed works mostly use latent variable evolution, which evolves parameters to generative machine learning models.

10.8 Machine Learning Evolved by Evolutionary Computation

This section addresses approaches where evolutionary computation is used to evolve one or more populations of generative machine learning models. We present direct applications of evolutionary computation to machine learning or parts thereof. These applications are direct in the sense that the evolutionary computation techniques are applied as is, i.e. without modifications, over the set of machine learning individuals. As an example, here we include the NEAT-like approaches because the NEAT algorithm evolves a population of artificial neural networks. Table 10.4 presents examples of evolutionary generative models which we consider to belong to the category *machine learning evolved by evolutionary computation*. The listed publications are grouped by application domain (first column) and then sorted by year (second column) and by authors (third column).

Most of the research within the literature concerning machine learning models evolved by evolutionary computation has to do with the evolution of artificial neural networks and CPPNs [190], and with the use of neuroevolution techniques such as NEAT [191]. Because most fixed-topology neuroevolution models typically generate artificial neural networks to be used for classification tasks, such approaches are not within the scope of this section. For literature regarding the use of evolutionary approaches in supervised and unsupervised learning, see Sects. 1.6 and 2.5, respectively.

There have been a number of works inspired by the CPPN-NEAT framework proposed by Stanley [190] to generate images by augmenting and evolving a CPPN [63, 181, 182, 206, 216]. It is worth noting that some of the mentioned works have been applied to the GAN model, as is the case with the replacement of a standard

Table 10.4 Examples of evolutionary generative models categorised as *machine learning evolved by evolutionary computation*

Domain	Year	Authors	EA	Population	Represent.	Fitness
Image	2007	[190] Stanley	GA	Single	Procedural	Dynamic
Image	2008	[181] Secretan et al.	GA	Single	Procedural	Dynamic
Image	2011	[182] Secretan et al.	GA	Single	Procedural	Dynamic
Image	2015	[216] Zhang et al.	GA	Single	Procedural	Dynamic
Image	2017	[192] Suganuma et al.	GP	Single	Procedural	Dynamic
Image	2018	[1] Al-Dujaili et al.	GA	Multi. (advers.)	Procedural	Dynamic
Image	2018	[68] Garcarena et al.	GA	Single	Procedural	Dynamic
Image	2018	[206] Turhan and Bilge	GA	Single	Procedural	Dynamic
Image	2019	[34] Cho and Kim	GA	Single	Parametric	Dynamic
Image	2019	[46] Costa et al.	GA	Multi. (colab.)	Procedural	Dynamic
Image	2019	[49] Costa et al.	GA	Multi. (colab.)	Procedural	Dynamic
Image	2019	[203] Toutouh et al.	GA	Multi. (advers.)	Procedural	Dynamic
Image	2019	[211] Wang et al.	GA	Single	Procedural	Dynamic
Image	2020	[47] Costa et al.	GA	Multi. (colab.)	Procedural	Dynamic
Image	2020	[204] Toutouh et al.	GA	Multi. (advers.)	Procedural	Dynamic
Image	2021	[33] Chen et al.	GA	Multi. (colab.)	Parametric	Dynamic
Image	2021	[63] Ekern and Gambäck	GA	Single	Procedural	Dynamic
Image	2021	[81] Hemberg et al.	GA	Multi. (advers.)	Procedural	Dynamic
Image	2021	[107] Li et al.	GA	Single	Procedural	Dynamic
Image	2021	[205] Toutouh and O'Reilly	GA	Multi. (advers.)	Procedural	Dynamic
Image	2022	[65] Flores et al.	GA	Multi. (advers.)	Procedural	Dynamic
Music	2008	[88] Hoover et al.	GA	Single	Procedural	Dynamic
Music	2009	[89] Hoover and Stanley	GA	Single	Procedural	Dynamic
Music	2011	[13] Bell	GA	Single	Procedural	Dynamic
Music	2011	[86] Hoover et al.	GA	Single	Procedural	Dynamic
Music	2014	[90] Hoover et al.	GA	Single	Procedural	Dynamic
Music	2015	[179] Scirea et al.	GA	Single	Procedural	Dynamic
Text	2020	[112] Liu et al.	GA	Single	Procedural	Dynamic
Other	2007	[200] Togelius et al.	ES	Multi. (advers.)	Procedural	Dynamic
Other	2009	[201] Togelius et al.	GP	Single	Procedural	Static
Other	2010	[59] Dubbin and Stanley	GA	Single	Procedural	Dynamic
Other	2012	[111] Liapis et al.	GA	Single	Procedural	Dynamic
Other	2013	[30] Cardona et al.	ES	Multi. (advers.)	Procedural	Dynamic

deep convolutional GAN generator with a CPPN evolved with NEAT, proposed by Ekern and Gambäck [63]. Another example related to GANs is the work by Turhan and Bilge [206], which combined a variational auto-encoder with the pixel-wise generative capability of CPPNs to create a GAN-variational auto-encoder model, i.e. a GAN with an inference mechanism in the form of an extra encoder network. Similar to Sect. 10.6, despite most models in this section not exhibiting competition,

these last two studies can be identified as having a competition between isolated components [63, 206].

Apart from NEAT approaches, but still within the context of genetic algorithms, Wang et al. [211] proposed E-GANs, which consisted in the evolution of populations of generators. Similar works also used GANs to evolve samples from the generator [34]. Costa et al. [49] worked on the coevolution of sub-populations of generators and discriminators based on CoDeepNEAT [149] with multiple variants, e.g. one based on novelty search with local competition [47]. The technique of evolving both the generator and the discriminator was applied in other works, both with different sub-populations for each component [33] and in a single population where each individual represents a tuple consisting of both components [68].

Regarding coevolutionary approaches, we can identify a sub-category of models dealing with spatial coevolution. This idea was first proposed by Al-Dujaili et al. [1] with the introduction of the Lipizzaner framework using a grid of cells where GANs are allowed to evolve and communicate. Other works expanded on Lipizzaner by hybridising it with E-GAN and Lipizzaner to increase solution diversity [203] and proposing a ring topology for the spatial cell grid instead of a toroidal one [205]. Other works apply the same idea by scaling the framework in the context of a high-performance computing setting for the medical domain [65] or by further exploring the features of Lipizzaner [81, 204]. Lastly, although during our literature review, we did not find a substantial body of work in this section outside of the image domain, it is interesting to note the evolution of Markov chains in the realm of music using genetic algorithms [13], an approach later explored further by Scirea [179] for the generation of music from scientific papers.

In the realm of music generation, many of the studies analysed here also employ NEAT to evolve CPPNs. These approaches typically use functional scaffolding, a technique that exploits the fact that a pattern of notes and rhythms in different instrumental parts of the same song is functionally related. These studies were initially performed by Hoover et al. [88] and Hoover and Stanley [89], who started by presenting a framework called NEAT Drummer for the generation of drum patterns, which was later extended to the application of full-fledged music composition [86, 90]. Concerning text generation, we point out the work of Liu et al. [112] pertaining to the creation of a category-aware model employing a genetic algorithm to evolve a population of GANs.

Albeit not many, there are a few noteworthy works in the scope of this section that are concerned with the generation of artefacts not pertaining to imagery or music. It is the case with the generation of shapes in video games by Liapis et al. [111], as well as other work exploring the coevolution of controllers [30, 200]. Furthermore, Dubbin and Stanley [59] generated dance moves using the CPPN-NEAT model.

As shown in Table 10.4, genetic algorithms account for the vast majority of approaches in this section (31 out of 32 papers), with only one using genetic programming to design both the structure and connectivity of the convolutional neural networks [192]. As a matter of fact, the standard NEAT framework representation is used in almost half of the models analysed herein and it is procedural by definition (in total there are 33 procedural, 2 parametric and no descriptive

representations). Lastly, regarding the fitness assignment, we conclude that most approaches are dynamic, especially the ones in an adversarial setting.

10.9 Open Problems and Challenges

Throughout the history of evolutionary generative models, several problems have been tackled progressively. It is the case with the fitness assignment of early evolutionary computation as discussed in Sect. 10.4. The first explorations of the generative capabilities of evolutionary computation models required the user to select the outputs for the evolutionary generative model. For many years, and especially in the fields of evolutionary art and music generation, user fitness assignment was the *de facto* technique. However, researchers noted that having the user in the loop with the system presented an enormous bottleneck that had to be addressed for the field to progress, and thus some papers started tackling this issue [12]. Nowadays, even though interactive systems are still very prolific, the most standard way of dealing with fitness assignment is to have some automatic algorithm, be it analytical, statistical (such as Markov models and classifiers) or indirectly informed in some other way. This section addresses some of these problems along with the strides that have recently been taken in addressing them.

Throughout this survey, we have identified the following challenges when dealing with evolutionary generative models.

10.9.1 *How to Represent the Space of Solutions Generated by the Model?*

GANs and other generative machine learning approaches are characterised by having a latent space, from which one selects a sample from that space, inputs it into the model and generates the output solution. Moreover, GANs allow for the conditional exploration of solutions through algebraic vector operations such as addition, subtraction and interpolation, to mention a few, which will, in turn, ease the exploration of such solutions. However, most evolutionary generative models, especially those that apply evolutionary computation directly (see Sect. 10.5), do not aggregate solutions in such an organised latent space. Therefore, a way to better represent the space of generated solutions is oftentimes needed. In this scope, several viable alternatives are considered throughout the literature. Although not latent spaces, spatial representations of the solutions have been constructed by resorting to an archive that stores solutions according to their novelty [42, 144] or by employing manifold learning algorithms to aggregate solutions on a space with lower dimensionality with techniques such as t-distributed stochastic neighbour embedding (t-SNE) [48]. Another trending solution is to build a latent space by combining a variational auto-encoder into the model by training an additional network that learns to encode

an image into a smaller latent vector. These are called the GAN-variational auto-encoder approaches [206]. At the moment, this challenge differentiates *evolutionary computation without machine learning* and *evolutionary computation aided by machine learning* approaches from *machine learning aided by evolutionary computation* and *machine learning evolved by evolutionary computation* because by using machine learning as a generative approach, typically the machine learning model will have the latent space with a solution space to be drawn. In the case of evolutionary computation-based approaches, we have solutions being evolved but they are independent solutions, unstructured, and not contained in a defined space, which can be seen as a drawback.

10.9.2 How to Navigate the Generated Space of Solutions?

This challenge concerns the exploration of all feasible solutions that can be generated by the model. Within the models that organise their solutions in a latent space, an effective way of exploring these solutions is to perform latent variable evolution directly over this space. Because a latent vector is the genotype representation of latent variable evolution, the types of algorithms used are mostly genetic algorithms [64, 73], and evolutionary strategies in some cases [22] (see Sect. 10.7 for a thorough analysis). For models where a latent space does not exist, the exploration of the solutions space cannot be done directly, and thus other methods need to be employed to aid solution exploration during evolution, e.g. exploration with variation operators or novelty search [110, 209]. Such methods typically involve hand-crafting specific variation operators to fence off the bias of the models. This can be seen, for instance, in the work by Li et al. [107] with the proposal of a new crossover operator for the E-GAN model and in the implementation of a mutation operator with features tailored to ease the process of interactive 3D design by Byrne et al. [28].

10.9.3 How to Evaluate the Generative Capabilities of a Model?

In this challenge, we are concerned with evaluating the generative capabilities of the model by itself without comparing it to similar generative models, whether evolutionary or not. This problem is often difficult to tackle as it depends a lot on the desired output of the model and the process itself. For instance, when it comes to music composition, Ng and Jordan [155] used genetic programming to examine evolving compositional processes instead of a musical piece. As further explained by Loughran and O'Neill [118], it is imperative to look beyond the generated output of a model when investigating the ability of the system to compose. This way, as analysed by Li et al. [106], there are two ways to evaluate a music composer: by evaluating the music it composes or by evaluating the composition process. This

idea can be extended outside the specific domain of music generation. Given the difficulties in evaluating isolated solutions, we might see more works where models evaluate the generative process itself and not only the output solutions.

10.9.4 How to Compare Evolutionary and Non-evolutionary Models?

Following the challenge of isolated evaluation, we turn our attention to the comparison of performance between different generative models, namely, between generative models with and without evolutionary computation. It is demonstrated that in adversarial generative models, the loss function is not stable enough to be taken as a comparative measure of generative performance [46]. In fact, it is shown that even simple metrics such as the inception score, which applies the Inception Model [194] to every generated image in order to get the conditional probability distribution of classes in the original data [173], fails to capture similarities between generated images adequately. For this reason, the Frechét inception distance metric [83] has been proposed. Although the Frechét inception distance was specifically developed to evaluate the performance of GANs, it can be applied to any image generation model. The Frechét inception distance directly compares the statistics of real samples with the statistics of generated ones, allowing for the capture of features used for a collection of both real and generated images gathered from training the network on image classification tasks. One major shortcoming of the inception score and Frechét inception distance scores is that they fail to determine whether the proposed model is overfitting or underfitting [20]. This problem pertains not only to evolutionary generative models but also to standard generative models. To overcome this hurdle, metrics such as the Reconstruction Error [212] and the Kernel Inception Distance were introduced [20]. Even though applying these metrics in evolutionary generative models still seems very rare, we posit that using these more advanced metrics will become more widespread in the study of evolutionary generative models as research in the field keeps taking off.

10.9.5 How to Improve the Computational Efficiency and Scalability of Evolutionary Generative Models?

Generally speaking, when it comes to improving computational efficiency, there are two ways a model can be optimised: through macro-optimisations, i.e. algorithmic improvements in our case, or through micro-optimisations, which in computer science typically means small statement-level optimisations to the source code. However, by micro-optimisations, we refer to the adaption of the model to run in different hardware according to the demands of operations in its algorithm. As an example, the work by Turhan and Bilge [206] tackles the problem of efficiently

generating high-resolution versions of handwritten digits, which is a slow task in conventional GAN-variational auto-encoder models. The proposed model converges more efficiently and thus faster than the baseline models by allowing the feature-wise reconstruction of a GAN-variational auto-encoder hybrid model instead of the standard pixel-wise reconstruction approach. This is a typical example of an algorithmic improvement. Another noteworthy example in the category of computational performance is TensorGP [9]. TensorGP is a genetic programming engine that not only performs algorithmic optimisations by changing the internal representation of symbolic expressions from the traditional tree graph to a directed acyclic graph, thus avoiding code re-execution but also has the option to evaluate expressions in dedicated hardware if available. This way, while most of the evolutionary stages such as variation operators, selection and other control mechanisms run on the central processing unit (CPU), the evaluation can take place in graphics processing units (GPUs) or tensor processing units (TPUs), which are capable of vectorising the pipeline of genetic programming operations, something that is still not implemented in many off-the-shelf genetic programming engines [8]. With the continued slowing down of Moore's law and GPU computing still on the rise, it makes sense to expect an increase in the number of frameworks using parallel capable hardware not only in evolutionary generative models but in generative models.

Coupled with the problem of computational efficiency, there is the problem of complexity associated with the generative process. The early works by Sims [185] in image generation using symbolic expressions are straightforward examples of low computational cost in terms of the generative process. As a counter-example, the work by Correia et al. [45] in enhancing image pipelines is fairly heavier than most generative processes because the evolutionary generative model encompasses more transformation steps. In this work, genetic programming was used to evolve image filters that, aside from requiring the traditional evaluation of the individual tree graphs, are then applied to a predefined image, which is, in turn, evaluated using an external classifier for fitness assignment. Overall, it seems that as time passes by, generative models tend to complexify, demanding more computational resources and thus becoming harder to scale and deal with.

10.9.6 How to Improve the Interaction of Evolutionary Generative Models?

Another aspect that we believe to be extremely relevant is the way we interact with evolutionary generative models. Not all generative models are easy to use for the average user, offering a poor user experience which will probably limit their reach to a broad audience. Therefore, it is essential to explore effective ways to control and visualise the different elements of an evolutionary generative model. This goes beyond facilitating a given action. Interaction can help to understand a certain functionality or aspect of the models that are often very technical.

For instance, although interfaces for large language models such as ChatGPT¹ have been trending as of late, the generative pre-trained transformer (GPT) model from which ChatGPT was built is not widely known despite having been created earlier [25]. The main reason for this lies in the fact that ChatGPT has a simple, natural and inviting user interface. Likewise, Midjourney,² a system which offers a model for the generation of images from natural language descriptions in the form of text prompts, is another example of an increasingly popular artificial intelligence service. Both examples have the benefit of being easily accessible: ChatGPT through the OpenAI website and Midjourney through a server hosted on Discord, a popular chat application.

Historically, interfaces in evolutionary generative models were created to aid the user to interact with them, control the evolutionary process, preview individuals and assign fitness [124, 181]. Other systems allow the designing of fitness functions [130, 144], a process that poses challenges in terms of interpretability and explainability to its design. There are other systems that allow for inspecting and navigation of the outputs [80, 143, 182], which also poses challenges from the implementation and user interaction perspectives.

However, the majority of evolutionary generative models possess neither this level of user-friendliness nor accessibility. In fact, most evolutionary generative model frameworks use their own interfaces, with different layouts and sometimes unintuitive functionality, lacking both generalisation and scalability.

We believe that the interaction with these models can be planned not only with human users in mind but also with other computational systems with which they can be integrated, following the example of nowadays machine learning-based systems. This can be done by developing APIs that allow for the development of new systems while promoting their modularity. The solution of using APIs solves the decoupling of the generative model, however, does not solve the part of adding a standardised experience for the interaction with the generative model which can be seen as an open problem.

10.9.7 How to Increase the Availability of Evolutionary Generative Model Systems?

As a final open challenge, we address the availability of the implemented systems to the general public. Within the surveyed literature, the number of papers with publicly source-code repositories is relatively scarce. Truth be told, this is not a problem specific to the field of generative models or even to artificial intelligence but one that is common in most research fields. Albeit a big challenge, websites such as huggingface.com and paperswithcode.com are a good step forward as they help disseminate and organise machine learning and artificial intelligence papers alike,

¹ <https://chat.openai.com/>.

² <https://docs.midjourney.com/>.

along with their respective code. Moreover, we believe that having open access to new methods and their respective implementations is especially important to advance the field of evolutionary generative models.

Having identified the core challenges and future trends in evolutionary generative models, it is clear that some are inherently more difficult to tackle than others. Challenges such as the availability of implemented evolutionary generative model systems are more of an ongoing problem that, arguably, is not expected to be solved so soon as it is pervasive in many other fields. Similarly, despite recent breakthroughs, technical challenges such as the construction of organised solution spaces or the evaluation of the generative capabilities of a model are non-obvious challenges that, due to being very dependent on the type of evolutionary generative model, will likely keep posing challenges as the topic of evolutionary generative models evolves, and new approaches are proposed. In contrast, challenges such as bottleneck mitigation and overhead reduction can arguably be easier to implement in the sense that they typically consist of general techniques that can be applied to a wide range of models. For instance, with the continuous rise in parallel computing, techniques as well as research and development investments in hardware with vectorisation technologies are increasing. Nonetheless, instead of dwelling on the many hurdles surrounding this topic, it is also worth praising the number of positive efforts being put in place to help advance and develop research on evolutionary generative models.

10.10 Conclusions

In this chapter, we presented a comprehensive survey on evolutionary generative models, analysing their main properties, along with a taxonomy to categorise these models. The lack of a well-established categorisation of these models in the literature led us to identify four main categories of evolutionary generative models: *evolutionary computation without machine learning*, *evolutionary computation aided by machine learning*, *machine learning aided by evolutionary computation* and *machine learning evolved by evolutionary computation*.

The importance of this contribution is twofold. First, it aims to standardise or at least provide a basis for further classifications of evolutionary generative models while easing the process of analysing the existing body of research. Second, we collected and categorised some of the most prominent literature concerning evolutionary generative models. We are aware that the work in evolutionary generative models is extensive, and therefore it is not feasible to carry out an exhaustive survey of all existing approaches.

Early instances of evolutionary generative models started as plain evolutionary computation approaches. With the growth in the research and adoption of machine learning seen within the past decades, the evolutionary computation approaches evolved into models incorporating machine learning techniques. For this reason, we have performed an in-depth listing of applications, surveys, position papers and

other research pertaining to and spanning from the beginnings of the use of evolutionary computation as a generative model to the state-of-the-art approaches. Overall, more than 200 publications ranging from plain evolutionary computation systems to evolutionary generative machine learning for the generation of images, music, text, as well as other domains were explored and classified. Each publication was classified according to the main properties that we have identified, namely, application domain, evolutionary algorithm, population, representation and fitness.

Furthermore, the extent of the present study made it possible to identify open problems and challenges for evolutionary generative models. Namely, challenges such as the need for adequate metrics for evaluating and validating generative performance, the aggregation of the generated outputs in a single, organised solution space as well as the search for well-suited operators capable of efficiently navigating through the associated solutions space is at the forefront of current open problems. It is worth noting that the growth of evolutionary generative models is tied to the maturity of the field of evolutionary machine learning, which, as an emerging field, is primarily propelled by the transparency and availability of the implemented systems to the general public. This challenge is in no way less relevant than the above-mentioned ones.

With the emergence of more promising methods such as GANs and other deep learning techniques, the use of evolutionary computation in the context of generative models might seem lost at first sight. However, based on recent trends demonstrated throughout this chapter, the future of the topic seems to point towards the symbiosis between both fields, where the power of machine learning interplays with the representation flexibility of evolutionary computation to improve the generative performance and computational efficiency of current and future evolutionary generative models.

Acknowledgements This work is funded by the FCT—Foundation for Science and Technology, I.P./MCTES through national funds (PIDDAC), within the scope of CISUC R&D Unit-UIDB/00326/2020 or project code UIDP/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020, under the FCT grant SFRH/BD/08254/2021.

References

1. Al-Dujaili, A., Schmiedlechner, T., O'Reilly, U.M.: Towards distributed coevolutionary GANs. In: Association for the Advancement of Artificial Intelligence (AAAI) 2018 Fall Symposium (2018)
2. Al-Najjar, K., Hämäläinen, M.: A master-apprentice approach to automatic creation of culturally satirical movie titles. In: Krahmer, E., Gatt, A., Goudbeek, M. (eds.) Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, pp. 274–283. Association for Computational Linguistics (2018). Accessed from 5–8 Nov 2018
3. Albarracn-Molina, D.D., Raglio, A., Rivas-Ruiz, F., Vico, F.J.: Using formal grammars as musical genome. *Appl. Sci.* **11**(9) (2021)
4. Alvarado, F.H.C., Lee, W.H., Huang, Y.H., Chen, Y.S.: Melody similarity and tempo diversity as evolutionary factors for music variations by genetic algorithms. In: Cardoso, F.A., Machado, P., Veale, T., Cunha, J.M. (eds.) Proceedings of the Eleventh International Confer-

- ence on Computational Creativity, ICCC 2020, Coimbra, Portugal, pp. 251–254. Association for Computational Creativity (ACC) (2020). 7–11 Sept 2020
5. Angelino, P.J.: Evolving fractal movies. In: Proceedings of the 1st Annual Conference on Genetic Programming, pp. 503–511. MIT Press (1996)
 6. Ashlock, D.A.: Evolutionary exploration of the mandelbrot set. In: IEEE International Conference on Evolutionary Computation, CEC 2006, part of WCCI 2006, Vancouver, BC, Canada, pp. 2079–2086. IEEE (2006). Accessed from 16–21 July 2006
 7. Aupetit, S., Bordeau, V., Monmarché, N., Slimane, M., Venturini, G.S.: Interactive evolution of ant paintings. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2003, Canberra, Australia, pp. 1376–1383. IEEE (2003). Accessed from 8–12 Dec 2003
 8. Baeta, F., Correia, J., Martins, T., Machado, P.: Speed benchmarking of genetic programming frameworks. In: Chicano, F., Krawiec, K. (eds.), GECCO '21: Genetic and Evolutionary Computation Conference, Lille, France, pp. 768–775. ACM (2021). Accessed from 10–14 July 2021
 9. Baeta, F., Correia, J., Martins, T., Machado, P.: Tensor GP - genetic programming engine in tensor flow. In: Castillo, P.A., Laredo, J.L.J. (eds.), Applications of Evolutionary Computation - 24th International Conference, Evo Applications 2021, Held as Part of Evo Star 2021, Virtual Event, Proceedings. Lecture Notes in Computer Science, vol. 12694, pp. 763–778. Springer (2021). Accessed from 7–9 April 2021
 10. Baeta, F., Correia, J., Martins, T., Machado, P.: Exploring expression-based generative adversarial networks. In: Fieldsend, J.E., Wagner, M. (eds.) GECCO '22: Genetic and Evolutionary Computation Conference, Companion Volume, Boston, Massachusetts, USA, pp. 1862–1869. ACM (2022). 9–13 July 2022
 11. Baker, E., Seltzer, M.: Evolving line drawings. In: Forrest, S. (ed.) Proceedings of the 5th International Conference on Genetic Algorithms, Urbana-Champaign, IL, USA, p. 627. Morgan Kaufmann (1993)
 12. Baluja, S., Pomerleau, D., Jochem, T.: Towards automated artificial evolution for computer-generated images. *Connect. Sci.* **6**(2–3), 325–354 (1994)
 13. Bell, C.: Algorithmic music composition using dynamic markov chains and genetic algorithms. *J. Comput. Sci. Coll.* **27**(2), 99–107 (2011)
 14. Bentley, P.J., Kumar, S.: Three ways to grow designs: a comparison of embryogenies for an evolutionary design problem. In: Banzhaf, W., Daida, J.M., Eiben, A.E., Garzon, M.H., Honavar, V.G., Kakiela, M.J., Smith, R.E. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999), Orlando, Florida, USA, pp. 35–43. Morgan Kaufmann (1999). Accessed from 13–17 July 1999
 15. Bergen, S., Ross, B.J.: Automatic and interactive evolution of vector graphics images with genetic algorithms. *Visual Comput.* **28**(1), 35–45 (2012)
 16. Bergen, S., Ross, B.J.: Aesthetic 3D model evolution. *Genetic Program. Evolvable Mach.* **14**(3), 339–367 (2013)
 17. Biles, J., Anderson, P., Loggi, L.: Neural Network Fitness Functions for a Musical IGA. In RIT Scholar Works, Rochester Institute of Technology (1996)
 18. Biles, J.A.: GenJam: a genetic algorithm for generating Jazz Solos. In: Proceedings of the 1994 International Computer Music Conference, ICMC 1994, Aarhus, Denmark. Michigan Publishing (1994). Accessed from 12–17 Sept 1994
 19. Bilotta, E., Pantano, P., Cupellini, E., Rizzuti, C.: Evolutionary methods for melodic sequences generation from non-linear dynamic systems. In: Giacobini, M., Brabazon, A., Cagnoni, S., Caro, G.D., Drechsler, R., Farooq, M., Fink, A., Lutton, E., Machado, P., Minner, S., O’Neill, M., Romero, J., Rothlauf, F., Squillero, G., Takagi, H., Uyar, S., Yang, S. (eds.) Applications of Evolutionary Computing, EvoWorkshops 2007: EvoCoMnet, EvoFIN, EvoIASP, EvoINTERACTION, EvoMUSART, EvoSTOC and EvoTransLog, Valencia, Spain, Proceedings. Lecture Notes in Computer Science, vol. 4448, pp. 585–592. Springer (2007). Accessed from 11–13 Apr 2007
 20. Binkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying MMD GANs. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, Conference Track Proceedings (2018). Accessed from April 30–May 3 2018

21. Bontrager, P., Lin, W., Togelius, J., Risi, S.: Deep interactive evolution. In: Liapis, A., Cardalda, J.J.R., Ekárt, A. (eds.) *Computational Intelligence in Music, Sound, Art and Design - 7th International Conference, EvoMUSART 2018, Parma, Italy, Proceedings. Lecture Notes in Computer Science*, vol. 10783, pp. 267–282. Springer, (2018). Accessed from 4–6 April 2018
22. Bontrager, P., Roy, A., Togelius, J., Memon, N., Ross, A.: Deep master prints: generating master prints for dictionary attacks via latent variable evolution. In: 9th IEEE International Conference on Biometrics Theory, Applications and Systems, BTAS 2018, Redondo Beach, CA, USA, pp. 1–9. IEEE (2018). Accessed from 22–25 Oct 2018
23. Bontrager, P., Roy, A., Togelius, J., Memon, N., Ross, A.: Deep master prints: generating master prints for dictionary attacks via latent variable evolution. In: 9th IEEE International Conference on Biometrics Theory, Applications and Systems, BTAS 2018, Redondo Beach, CA, USA, pp. 1–9. IEEE (2018). Accessed from 22–25 Oct 2018
24. Brown, J.A., Ashlock, D., Orth, J., Houghten, S.: Autogeneration of fractal photographic mosaic images. In: *Proceedings of the IEEE congress on evolutionary computation, CEC 2011, New Orleans, LA, USA*, pp. 1116–1123. IEEE (2011). Accessed from 5–8 June 2011
25. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS 2020, Virtual Event*, pp. 1877–1901 (2020). Accessed from 6–12 Dec 2020
26. Browne, C., Maire, F.: Evolutionary game design. *IEEE Trans. Comput. Intell. AI in Games* **2**(1), 1–16 (2010)
27. Burton, A.R., Vladimirova, T.: Genetic algorithm utilising neural network fitness evaluation for musical composition. In: Smith, G.D., Steele, N.C., Albrecht, R.F. (eds.) *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms, ICANNGA 1997, Norwich, UK, 1997*, pp. 219–223. Springer (1997)
28. Byrne, J., McDermott, J., López, E.G., O’Neill, M.: Implementing an intuitive mutation operator for interactive evolutionary 3D design. In: *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010, Barcelona, Spain*, pp. 1–7. IEEE (2010). Accessed from 18–23 July 2010
29. Cardona, A.B., Hansen, A.W., Togelius, J., Friberger, M.G.: Open trumps, a data game. In: Mateas, M., Barnes, T., Bogost, I. (eds.) *Proceedings of the 9th International Conference on the Foundations of Digital Games, FDG 2014, Liberty of the Seas, Caribbean. Society for the Advancement of the Science of Digital Games (2014)*. Accessed from 3–7 Apr 2014
30. Cardona, A.B., Togelius, J., Nelson, M.J.: Competitive coevolution in Ms. Pac-Man. In: *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2013, Cancun, Mexico*, pp. 1403–1410. IEEE (2013). Accessed from 20–23 June 2013
31. Chapuis, J., Lutton, E.: Artie-fract: interactive evolution of fractals. In: *International Conference on Generative Art (2001)*
32. Chen, J., Chen, X., Yang, J., Shan, S., Wang, R., Gao, W.: Optimization of a training set for more robust face detection. *Pattern Recogn.* **42**(11), 2828–2840 (2009)
33. Chen, S., Wang, W., Xia, B., You, X., Peng, Q., Cao, Z., Ding, W.: CDE-GAN: cooperative dual evolution-based generative adversarial network. *IEEE Trans. Evol. Comput.* **25**(5), 986–1000 (2021)
34. Cho, H., Kim, Y.: Stabilized training of generative adversarial networks by a genetic algorithm. In: López-Ibáñez, M., Auger, A., Stützle, T. (eds.) *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2019, Prague, Czech Republic*, pp. 51–52. ACM (2019). Accessed from 13–17 July 2019
35. Collomosse, J.P.: Evolutionary search for the artistic rendering of photographs. In: Romero, J., Machado, P. (eds.) *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music, Natural Computing Series*, pp. 39–62. Springer (2008)

36. Colton, S.: Automatic invention of fitness functions with application to scene generation. In: Giacobini, M., Brabazon, A., Cagnoni, S., Caro, G.D., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A., Farooq, M., Fink, A., McCormack, J., O'Neill, M., Romero, J., Rothlauf, F., Squillero, G., Uyar, S., Yang, S. (eds.) *Applications of Evolutionary Computing, EvoWorkshops 2008: EvoCOMNET, EvoFIN, EvoHOT, EvoIASP, EvoMUSART, EvoNUM, EvoSTOC, and EvoTransLog*, Naples, Italy, *Proceedings, Lecture Notes in Computer Science*, vol. 4974, pp. 381–391. Springer (2008). Accessed from 26–28 March 2008
37. Colton, S.: Evolving a library of artistic scene descriptors. In: Machado, P., Romero, J., Carballal, A. (eds.) *Evolutionary and Biologically Inspired Music, Sound, Art and Design - First International Conference, EvoMUSART 2012*, Málaga, Spain, *Proceedings. Lecture Notes in Computer Science*, vol. 7247, pp. 35–47. Springer (2012). Accessed from 11–13 Apr 2012
38. Colton, S.: Evolving neural style transfer blends. In: Romero, J., Martins, T., Rodríguez-Fernández, N. (eds.) *Artificial Intelligence in Music, Sound, Art and Design - 10th International Conference, EvoMUSART 2021*, Held as Part of EvoStar 2021, Virtual Event, *Proceedings. Lecture Notes in Computer Science*, vol. 12693, pages 65–81. Springer (2021). Accessed from 7–9 Apr 2021
39. Cook, M., Colton, S.: redesigning computationally creative systems for continuous creation. In: Pachet, F., Jordanous, A., León, C. (eds.) *Proceedings of the Ninth International Conference on Computational Creativity, ICCO 2018*, Salamanca, Spain, pp. 32–39. Association for Computational Creativity (ACC) (2018). Accessed from 25–29 June 2018
40. Cook, M., Colton, S., Gow, J.: The ANGELINA videogame design system - Part I. *IEEE Trans. Comput. Intell. AI in Games* **9**(2), 192–203 (2017)
41. Correia, J., Machado, P., Romero, J., Carballal, A.: Evolving figurative images using expression-based evolutionary art. In: Maher, M.L., Veale, T., Saunders, R., Bown, O. (eds.) *Proceedings of the Fourth International Conference on Computational Creativity, ICCO 2013*, Sidney, Australia, pp. 24–31 (2013). Accessed from 12–14 June 2013
42. Correia, J., Machado, P., Romero, J., Martins, P., Cardoso, F.A.: Breaking the mould an evolutionary quest for innovation through style change. In: Veale, F.A., Cardoso, F.A. (eds.) *Computational Creativity - The Philosophy and Engineering of Autonomously Creative Systems*, pp. 353–398. Springer (2019)
43. Correia, J., Martins, T., Machado, P.: Evolutionary data augmentation in deep face detection. In: López-Ibáñez, M., Auger, A., Stützle, T. (eds.) *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2019*, Prague, Czech Republic, pp. 163–164. ACM (2019). Accessed 13–17 July 2019
44. Correia, J., Martins, T., Martins, P., Machado, P.: X-faces: the eXploit is out there. In: Pachet, F., Cardoso, A., Corruble, V., Ghedini, F. (eds.) *Proceedings of the Seventh International Conference on Computational Creativity, ICCO 2016*, UPMC, Paris, France, pp. 164–171. Sony CSL (2016). Accessed from June 27–July 1 2016
45. Correia, J., Vieira, L., Rodríguez-Fernández, N., Romero, J., Machado, P.: Evolving image enhancement pipelines. In: Romero, J., Martins, T., Rodríguez-Fernández, N. (eds.) *Artificial Intelligence in Music, Sound, Art and Design - 10th International Conference, EvoMUSART 2021*, Held as Part of EvoStar 2021, Virtual Event, *Proceedings. Lecture Notes in Computer Science*, vol. 12693, pp. 82–97. Springer (2021). Accessed from 7–9 Apr 2021
46. Costa, V., Lourenço, N., Correia, J., Machado, P.: COEGAN: evaluating the coevolution effect in generative adversarial networks. In: Auger, A., Stützle, T. (eds.) *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019*, Prague, Czech Republic, pp. 374–382. ACM (2019). Accessed from 13–17 July 2019
47. Costa, V., Lourenço, N., Correia, J., Machado, P.: Exploring the evolution of gans through quality diversity. In: Coello, C.A.C. (ed.) *GECCO '20: Genetic and Evolutionary Computation Conference*, Cancún Mexico, pp. 297–305. ACM (2020). Accessed from 8–12 July 2020
48. Costa, V., Lourenço, N., Correia, J., Machado, P.: Demonstrating the evolution of GANs through t-SNE. In: Castillo, P.A., Laredo, J.L.J. (eds.) *Applications of Evolutionary Computation - 24th International Conference, EvoApplications 2021*, Held as Part of EvoStar 2021,

- Virtual Event, Proceedings. Lecture Notes in Computer Science, vol. 12694, pp. 618–633. Springer (2021). Accessed from 7–9 Apr 2021
49. Costa, V., Lourenço, N., Machado, P.: Coevolution of generative adversarial networks. In: Kaufmann, P., Castillo, P.A. (eds.) Applications of Evolutionary Computation - 22nd International Conference, EvoApplications 2019, Held as Part of EvoStar 2019, Leipzig, Germany, Proceedings. Lecture Notes in Computer Science, vol. 11454, pp. 473–487. Springer (2019). Accessed from 24–26 Apr 2019
 50. Cunha, J.M., Martins, P., Lourenço, N., Machado, P.: Emojecting co-creativity: integrating self-evaluation and context-adaptation. In: Cardoso, F.A., Machado, P., Veale, T., Cunha, J.M. (eds.) Proceedings of the Eleventh International Conference on Computational Creativity, ICCO 2020, Coimbra, Portugal, pp. 85–88. Association for Computational Creativity (ACC) (2020). Accessed from 7–11 Sept 2020
 51. Dahlstedt, P.: Sounds Unheard of: Evolutionary Algorithms as Creative Tools for the Contemporary Composer. Ph.D. thesis, Chalmers, School of Architecture, University of Gothenburg (2004)
 52. Dawkins, R.: *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe Without Design*. W. W. Norton & Company (1986)
 53. de la Puente, A.O., Alfonso, R.S., Alfonseca, M.: Automatic composition of music by means of grammatical evolution. In: Proceedings of the 2002 International Conference on APL: Array Processing Languages: Lore, Problems, and Applications, APL 2002, Madrid, Spain, pp. 148–155. ACM (2002). Accessed from 22–25 July 2002
 54. den Heijer, E., Eiben, A.E.: Evolving pop art using scalable vector graphics. In: Machado, P., Romero, J., Carballal, A. (eds.) Evolutionary and Biologically Inspired Music, Sound, Art and Design - First International Conference, EvoMUSART 2012, Málaga, Spain, Proceedings. Lecture Notes in Computer Science, vol. 7247, pp. 48–59. Springer (2012). Accessed from 11–13 Apr 2012
 55. DiPaola, S.: Evolving creative portrait painter programs using darwinian techniques with an automatic fitness function. In: EVA 2005 London Conference (2005)
 56. DiPaola, S.R., Gabor, L.: Incorporating characteristics of human creativity into an evolutionary art algorithm. In: Thierens, D. (ed.) Genetic and Evolutionary Computation Conference, GECCO 2007, Proceedings, London, England, UK, Companion Material, pp. 2450–2456. ACM (2007). Accessed from 7–11 July 2007
 57. Donnelly, P.J., Sheppard, J.W.: Evolving four-part harmony using genetic algorithms. In: Chio, C.D., Brabazon, A., Caro, G.A.D., Drechsler, R., Farooq, M., Grahl, J., Greenfield, G., Prins, C., Romero, J., Squillero, G., Tarantino, E., Tettamanzi, A., Urquhart, N., Etnan-Uyar, A.S. (eds.) Applications of Evolutionary Computation - EvoApplications 2011: EvoCOMNET, EvoFIN, EvoHOT, EvoMUSART, EvoSTIM, and EvoTRANSLOG, Torino, Italy, Proceedings, Part II. Lecture Notes in Computer Science, vol. 6625, pp. 273–282. Springer (2011). Accessed from 27–29 April 2011
 58. Draves, S.: The electric sheep screen-saver: a case study in aesthetic evolution. In: Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G. (eds.) Applications of Evolutionary Computing, EvoWorkshops 2005: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, and EvoS-TOC, Lausanne, Switzerland, Proceedings. Lecture Notes in Computer Science, vol. 3449, pp. 458–467. Springer (2005). Accessed from March 30–April 1, 2005
 59. Dubbin, G.A., Stanley, K.O.: Learning to dance through interactive evolution. In: Chio, C.D., Brabazon, A., Caro, G.A.D., Ebner, M., Farooq, M., Fink, A., Grahl, J., Greenfield, G., Machado, P., O’Neill, M., Tarantino, E., Urquhart, N. (eds.) Applications of Evolutionary Computation, EvoApplications 2010: EvoCOMNET, EvoENVIRONMENT, EvoFIN, EvoMUSART, and EvoTRANSLOG, Istanbul, Turkey, Proceedings, Part II. Lecture Notes in Computer Science, vol. 6025, pp. 331–340. Springer (2010). Accessed from 7–9 Apr 2010
 60. Eigenfeldt, A., Bown, O., Brown, A.R., Gifford, T.: Flexible generation of musical form: beyond mere generation. In: Pachet, F., Cardoso, A., Corruble, V., Ghedini, F. (eds.) Proceedings of the Seventh International Conference on Computational Creativity, ICCO 2016, UPMC, Paris, France, pp. 264–271. Sony CSL (2016). Accessed from June 27–July 1, 2016

61. Eigenfeldt, A., Pasquier, P.: Populations of populations: composing with multiple evolutionary algorithms. In: Machado, P., Romero, J., Carballal, A. (eds.) *Evolutionary and Biologically Inspired Music, Sound, Art and Design - First International Conference, EvoMUSART 2012*, Málaga, Spain. *Proceedings. Lecture Notes in Computer Science*, vol. 7247, pp. 72–83. Springer (2012). Accessed from 11–13 Apr 2012
62. Ekárt, A.: Genetic programming for the design of lace knitting stitch patterns. In: Ellis, R., Allen, T., Petridis, M. (eds.) *Applications and Innovations in Intelligent Systems XV*, pp. 261–274. Springer, London (2008)
63. Ekern, E.G., Gambäck, B.: Interactive, efficient and creative image generation using compositional pattern-producing networks. In: Romero, J., Martins, T., Rodríguez-Fernández, N. (eds.) *Artificial Intelligence in Music, Sound, Art and Design - 10th International Conference, EvoMUSART 2021*, Held as Part of *EvoStar 2021*, Virtual Event, *Proceedings. Lecture Notes in Computer Science*, vol. 12693, pp. 131–146. Springer (2021). Accessed from 7–9 Apr 2021
64. Fernandes, P., Correia, J., Machado, P.: Evolutionary latent space exploration of generative adversarial networks. In: Castillo, P.A., Laredo, J.L.J., de Vega, F.F. (eds.) *Applications of Evolutionary Computation - 23rd European Conference, EvoApplications 2020*, Held as Part of *EvoStar 2020*, Seville, Spain, *Proceedings. Lecture Notes in Computer Science*, vol. 12104, pp. 595–609. Springer (2020). Accessed from 15–17 Apr 2020
65. Flores, D., Hemberg, E., Toutouh, J., O'Reilly, U.: Coevolutionary generative adversarial networks for medical image augmentation at scale. In: Fieldsend, J.E., Wagner, M. (eds.) *GECCO '22: Genetic and Evolutionary Computation Conference*, Boston, Massachusetts, USA, pp. 367–376. ACM (2022). Accessed from 9–13 July 2022
66. Frade, M., de Vega, F.F., Cotta, C.: Modelling video games' landscapes by means of genetic terrain programming - a new approach for improving users' experience. In: Giacobini, M., Brabazon, A., Cagnoni, S., Caro, G.D., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A., Farooq, M., Fink, A., McCormack, J., O'Neill, M., Romero, J., Rothlauf, F., Squillero, G., Uyar, S., Yang, S. (eds.) *Applications of Evolutionary Computing, EvoWorkshops 2008: EvoCOMNET, EvoFIN, EvoHOT, EvoIASP, EvoMUSART, EvoNUM, EvoSTOC, and EvoTransLog*, Naples, Italy. *Proceedings. Lecture Notes in Computer Science*, vol. 4974, pp. 485–490. Springer (2008). Accessed from 26–28 March 2008
67. Galanter, P.: What is generative art? Complexity theory as a context for art theory. In: *GA2003 – 6th Generative Art Conference* (2003)
68. Garciarena, U., Santana, R., Mendiburu, A.: Evolved GANs for generating pareto set approximations. In: Aguirre, H.E., Takadama, H.E. (eds.) *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018*, Kyoto, Japan, pp. 434–441. ACM (2018). Accessed from 15–19 July 2018
69. Gartland-Jones, A.: MusicBlox: a real-time algorithmic composition system incorporating a distributed interactive genetic algorithm. In: Raidl, G.R., Meyer, J., Middendorf, M., Cagnoni, S., Cardalda, J.J.R., Corne, D., Gottlieb, J., Guillot, A., Hart, E., Johnson, C.G., Marchiori, E. (eds.) *Applications of Evolutionary Computing, EvoWorkshop 2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, and EvoSTIM*, Essex, UK, *Proceedings. Lecture Notes in Computer Science* vol. 2611, pp. 490–501. Springer (2003). Accessed from 14–16 Apr 2003
70. Gervás, P.: Evolutionary elaboration of daily news as a poetic stanza. In: *IX Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados - MAEB 2013*, Universidad Complutense de Madrid, Spain (2013)
71. Gibson, P., Byrne, J.: NEUROGEN, musical composition using genetic algorithms and cooperating neural networks. In: *1991 Second International Conference on Artificial Neural Networks*, pp. 309–313 (1991)
72. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, Montreal, Quebec, Canada, pp. 2672–2680 (2014). Accessed from 8–13 Dec 2014

73. Grabe, I., Zhu, J., Agirrezabal, M.: Fashion style generation: evolutionary search with gaussian mixture models in the latent space. In: Martins, T., Rodríguez-Fernández, N., Rebelo, S.M. (eds.) *Artificial Intelligence in Music, Sound, Art and Design - 11th International Conference, EvoMUSART 2022, Held as Part of EvoStar 2022, Madrid, Spain, Proceedings*. Lecture Notes in Computer Science, vol. 13221, pp.84–100. Springer (2022). Accessed from 20–22 Apr 2022
74. Graf, J., Banzhaf, W.: Interactive evolution of images. In: McDonnell, J.R., Reynolds, R.G., Fogel, D.B. (eds.) *Proceedings of the Fourth Annual Conference on Evolutionary Programming, EP 1995, San Diego, CA, USA, pp. 53–65*. MIT Press (1995). Accessed from 1–3 March 1995
75. Greenfield, G.: A platform for evolving controllers for simulated drawing robots. In: Machado, P., Romero, J., Carballal, A. (eds.) *Evolutionary and Biologically Inspired Music, Sound, Art and Design - First International Conference, EvoMUSART 2012, Málaga, Spain, Proceedings*. Lecture Notes in Computer Science, vol. 7247, pp. 108–116. Springer (2012). Accessed from 11–13 Apr 2012
76. Greenfield, G.R.: New directions for evolving expressions. In: *Bridges: Mathematical Connections in Art, Music, and Science*, pp. 29–36 (1998)
77. Greenfield, G.R.: Evolving expressions and art by choice. *Leonardo* **33**(2), 93–99 (2000)
78. Greenfield, G.R.: Tilings of sequences of co-evolved images. In: Raidl, G.R., Cagnoni, S., Branke, J., Corne, D., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G. (eds.) *Applications of Evolutionary Computing, EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, and EvoSTOC, Coimbra, Portugal, Proceedings*. Lecture Notes in Computer Science, vol. 3005, pp. 427–436. Springer (2004). Accessed from 5–7 Apr 2004
79. Greenfield, G.R.: Evolutionary methods for ant colony paintings. In: Raidl, G.R., Cagnoni, S., Branke, J., Corne, D., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G. (eds.) *Applications of Evolutionary Computing, EvoWorkshops 2005: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, and EvoSTOC, Lausanne, Switzerland, Proceedings*. Lecture Notes in Computer Science, vol. 3449, pp. 478–487. Springer (2005). Accessed from March 30–April 1 2005
80. Hart, D.A.: Toward greater artistic control for interactive evolution of images and animation. In: MGiacobini, M., Brabazon, A., Cagnoni, S., Caro, G.D., Drechsler, R., Farooq, M., Fink, A., Lutton, E., Machado, P., Minner, S., O’Neill, M., Romero, J., Rothlauf, F., Squillero, G., Takagi, H., Uyar, S., Yang, S. (eds.) *Applications of Evolutionary Computing, EvoWorkshops 2007: EvoCoMnet, EvoFIN, EvoIASP, EvoINTERACTION, EvoMUSART, EvoSTOC and EvoTransLog, Valencia, Spain, Proceedings*. Lecture Notes in Computer Science, vol. 4448, pp. 527–536. Springer (2007). Accessed from 11–13 Apr 2007
81. Hemberg, E., Toutouh, J., Al-Dujaili, A., Schmiedlechner, T., O’Reilly, U.: Spatial coevolution for generative adversarial network training. *ACM Trans. Evol. Learn. Optim.* **1**(2), 6:1–6:28 (2021)
82. Hervás, R., Robinson, J., Gervás, P.: Evolutionary assistance in alliteration and allelic drift. In: Giacobini, M., Brabazon, A., Cagnoni, S., Caro, G.D., Drechsler, R., Farooq, M., Fink, A., Lutton, E., Machado, P., Minner, S., O’Neill, M., Romero, J., Rothlauf, F., Squillero, G., Takagi, H., Uyar, S., Yang, S. (eds.) *Applications of Evolutionary Computing, EvoWorkshops 2007: EvoCoMnet, EvoFIN, EvoIASP, EvoINTERACTION, EvoMUSART, EvoSTOC and EvoTransLog, Valencia, Spain, Proceedings*. Lecture Notes in Computer Science, vol. 4448 pp. 537–546. Springer (2007). Accessed from 11–13 Apr 2007
83. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, pp. 6626–6637* (2017). Accessed from 4–9 Dec 2017
84. Hickenbotham, S.J., Stepney, S.: Augmenting live coding with evolved patterns. In: Johnson, C.G., Ciesielski, V., Correia, J., Machado, P. (eds.,) *Evolutionary and Biologically Inspired*

- Music, Sound, Art and Design - 5th International Conference, EvoMUSART 2016, Porto, Portugal, Proceedings. Lecture Notes in Computer Science, vol. 9596, pp. 31–46. Springer (2016). March 30–April 1, 2016
85. Hofmann, D.M.: A genetic programming approach to generating musical compositions. In: Johnson, C.G., Carballal, A., Correia, J. (eds.) *Evolutionary and Biologically Inspired Music, Sound, Art and Design - 4th International Conference, EvoMUSART 2015*, Copenhagen, Denmark, Proceedings. Lecture Notes in Computer Science, vol. 9027, pp. 89–100. Springer (2015). Accessed from 8–10 Apr 2015
 86. Hoover, A., Szerlip, P., Stanley, K.: Generating musical accompaniment through functional scaffolding. In: *Proceedings of the Eighth Sound and Music Computing Conference (SMC 2011)*, pp. 161–168 (2011)
 87. Hoover, A.K., Cachia, W., Liapis, A., Yannakakis, G.N.: Audio in space: exploring the creative fusion of generative audio, visuals and gameplay. In: Johnson, C.G., Carballal, A., Correia, J. (eds.), *Evolutionary and Biologically Inspired Music, Sound, Art and Design - 4th International Conference, EvoMUSART 2015*, Copenhagen, Denmark, Proceedings. Lecture Notes in Computer Science, vol. 9027, pp. 101–112. Springer (2015). Accessed from 8–10 Apr 2015
 88. Hoover, A.K., Rosario, M.P., Stanley, K.O.: Scaffolding for interactively evolving novel drum tracks for existing songs. In: Giacobini, M., Brabazon, A., Cagnoni, S., Caro, G.D., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A., Farooq, M., Fink, A., McCormack, J., O’Neill, M., Romero, J., Rothlauf, F., Squillero, G., Uyar, S., Yang, S. (eds.) *Applications of Evolutionary Computing, EvoWorkshops 2008: EvoCOMNET, EvoFIN, EvoHOT, EvoIASP, EvoMUSART, EvoNUM, EvoSTOC, and EvoTransLog*, Naples, Italy, Proceedings. Lecture Notes in Computer Science, vol. 4974, pp. 412–422. Springer (2008). Accessed from 26–28 March 2008
 89. Hoover, A.K., Stanley, K.O.: Exploiting functional relationships in musical composition. *Connect. Sci.* **21**(2&3), 227–251 (2009)
 90. Hoover, A.K., Szerlip, P.A., Stanley, K.O.: Functional scaffolding for composing additional musical voices. *Comput. Music J.* **38**(4), 80–99 (2014)
 91. Horner, A., Goldberg, D.E.: Genetic algorithms and computer-assisted music composition. In: Belew, R.K., Booker, L.B. (eds.) *Proceedings of the 4th International Conference on Genetic Algorithms*, San Diego, CA, USA, July 1991, pp. 437–441. Morgan Kaufmann (1991)
 92. Horowitz, D.: Generating rhythms with genetic algorithms. In: Hayes-Roth, B., Korf, R.E. (eds.) *Proceedings of the 12th National Conference on Artificial Intelligence*, Seattle, WA, USA, Vol. 2, p. 1459. AAAI Press/The MIT Press (1994). Accessed from July 31–August 4, 1994
 93. Hui Yap, A.Y., Soong, H.-C., Hong Tse, S.S.: Real-time evolutionary music composition using JFUGUE and genetic algorithm. In: *2021 IEEE 19th Student Conference on Research and Development (SCOREd)*, pp. 377–382 (2021)
 94. Ianigro, S., Bown, O.: Plecto: a low-level interactive genetic algorithm for the evolution of audio. In: Johnson, C.G., Ciesielski, V., Correia, J., Machado, P. (eds.) *Evolutionary and Biologically Inspired Music, Sound, Art and Design - 5th International Conference, EvoMUSART 2016*, Porto, Portugal, Proceedings. Lecture Notes in Computer Science, vol. 9596, pp. 63–78. Springer (2016). Accessed from March 30–April 1, 2016
 95. Jacob, B.L.: *Composing with genetic algorithms*. In: *Proceedings of the 1995 International Computer Music Conference, ICMC 1995*, Banff, AB, Canada. Michigan Publishing (1995). Accessed from 3–7 Sept 1995
 96. Johanson, B., Poli, R.: GP-music: an interactive genetic programming system for music generation with automated fitness raters. In: Koza, J.R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D.B., Garzon, M.H., Goldberg, D.E., Iba, H., Riolo, R. (eds.) *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pp. 181–186. Morgan Kaufmann (1998)
 97. Kaliakatsos-Papakostas, M.A., Epitropakis, M.G., Floros, A., Vrahatis, M.N.: Interactive evolution of 8-bit melodies with genetic programming towards finding aesthetic measures

- for sound. In: Machado, P., Romero, J., Carballal, A. (eds.) *Evolutionary and Biologically Inspired Music. Sound, Art and Design*, pp. 141–152. Springer, Berlin Heidelberg (2012)
98. Khalifa, Y.M.A., Khan, B.K., Begovic, J., Wisdom, A., Wheeler, A.M.: Evolutionary music composer integrating formal grammar. In: Thierens, D. (ed.) *Genetic and Evolutionary Computation Conference, GECCO 2007, Proceedings, London, England, UK, Companion Material*, pp. 2519–2526. ACM (2007). Accessed from 7–11 July 2007
 99. Korde, C.G., K., M.R., V, M.H., N. K. Y. B.: Training of generative adversarial networks with hybrid evolutionary optimization technique. In: 2019 IEEE 16th India Council International Conference (INDICON), pp. 1–4 (2019)
 100. Koza, J.R.: Genetic programming as a means for programming computers by natural selection. *Stat Comput* 4(2) (1994)
 101. Kunitatsu, K., Ishikawa, Y., Takata, M., Joe, K.: A music composition model with genetic programming – a case study of chord progression and bassline. In: *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, pp. 256–262 (2015)
 102. Laine, P., Kuuskankare, M.: Genetic algorithms in musical style oriented generation. In: *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Orlando, Florida, USA*, pp. 858–862. IEEE (1994). Accessed from 27–29 June 1994
 103. Levy, R.P.: A computational model of poetic creativity with neural network as measure of adaptive fitness. In: *Proceedings of the First Workshop on Creative Systems, International Conference of Case-Based Reasoning* (2001)
 104. Lewis, M.: Aesthetic evolutionary design with data flow networks. In: *International Conference on Generative Art* (2000)
 105. Lewis, M.: *Evolutionary Visual Art and Design*, pp. 3–37. Springer, Berlin Heidelberg (2008)
 106. Li, J., Jing, M., Lu, K., Ding, Z., Zhu, L., Huang, Z.: Leveraging the invariant side of generative zero-shot learning. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA*, pp. 7402–7411. Computer Vision Foundation/IEEE (2019). Accessed from 16–20 June 2019
 107. Li, J., Zhang, J., Gong, X., Lü, S.: Evolutionary generative adversarial networks with crossover based knowledge distillation. In: *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China*, pp. 1–8. IEEE (2021). Accessed from 18–22 July 2021
 108. Liapis, A.: Recomposing the Pokémon color palette. In: Sim, K., Kaufmann, P. (eds.) *Applications of Evolutionary Computation*, pp. 308–324. Springer International Publishing (2018)
 109. Liapis, A.: Artificial intelligence for designing games. In: Machado, P., Romero, J., Greenfield, G. (eds.) *The Handbook of Artificial Intelligence and the Arts*. Springer (2021)
 110. A. Liapis, H. Martínez, J. Togelius, and G. Yannakakis. Transforming Exploratory Creativity with DeLeNoX. In M. Maher, T. Veale, R. Saunders, and O. Bown, editors, *Proceedings of the 4th International Conference on Computational Creativity, ICCCC 2013*, pages 56–63. Faculty of Architecture, Design and Planning, The University of Sydney, 2013
 111. Liapis, A., Yannakakis, G.N., Togelius, J.: Adapting models of visual aesthetics for personalized content creation. *IEEE Trans. Comput. Intell. AI Games* 4(3), 213–228 (2012)
 112. Liu, Z., Wang, J., Liang, Z.: CatGAN: category-aware generative adversarial networks with hierarchical evolutionary learning for category text generation. In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA*, pp. 8425–8432. AAAI Press (2020). Accessed from 7–12 Feb 2020
 113. Lopes, D., Correia, J., Machado, P.: Adea - evolving glyphs for aiding creativity in typeface design. In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, GECCO '20*, pp. 97–98. Association for Computing Machinery (2020)
 114. Lopes, P., Liapis, A., Yannakakis, G.N.: Framing tension for game generation. In: Pachet, F., Cardoso, A., Corruble, V., Ghedini, F. (eds.) *Proceedings of the Seventh International Conference on Computational Creativity, ICCCC 2016, UPMC, Paris, France*, pp. 205–212. Sony CSL (2016). Accessed from June 27–July 1, 2016

115. Loughran, R., McDermott, J., O'Neill, M.: Tonality driven piano compositions with grammatical evolution. In: *IEEE Congress on Evolutionary Computation, CEC 2015, Sendai, Japan*, pp. 2168–2175. IEEE (2015). Accessed from 25–28 May 2015
116. Loughran, R., McDermott, J., O'Neill (2016) Grammatical music composition with dissimilarity driven hill climbing. In: Johnson, C.G., Ciesielski, V., Correia, J., Machado, P. (eds.) *Evolutionary and Biologically Inspired Music, Sound, Art and Design - 5th International Conference, EvoMUSART 2016, Porto, Portugal, Proceedings. Lecture Notes in Computer Science*, vol. 9596, pp. 110–125. Springer (2016). Accessed from March 30–April 1, 2016
117. Loughran, R., O'Neill, M.: My little chunky: towards live-coding with grammatical evolution. In: *Musical Metacreation (MuMe)* (2017)
118. Loughran, R., O'Neill, M.: Evolutionary music: applying evolutionary computation to the art of creating music. *Genetic Program. Evolvable Mach.* **21**(1–2), 55–85 (2020)
119. Loughran, R., O'Neill, M.: the popular critic: evolving melodies with popularity driven fitness. In: *MUME 2016 - The Fourth International Workshop on Musical Metacreation* (2016)
120. Lucas, P., Martinho, C.: Stay awhile and listen to 3buddy, a co-creative level design support tool. In: Goel, A.K., Jordanous, A., Pease, A. (eds.) *Proceedings of the Eighth International Conference on Computational Creativity, ICC 2017, Atlanta, Georgia, USA*, pp. 205–212. Association for Computational Creativity (ACC) (2017). Accessed from 19–23 June 2017
121. Maças, C., Lourenço, N., Machado, P.: Interactive evolution of swarms for the visualisation of consumptions. In: Brooks, A.L., Brooks, E., Sylla, C. (eds.) *Interactivity, Game Creation, Design, Learning, and Innovation - 7th EAI International Conference, ArtsIT 2018, and 3rd EAI International Conference, DLI 2018, ICTCC 2018, Braga, Portugal, Proceedings. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 265, pp. 101–110. Springer (2018). Accessed from 24–26 Oct 2018
122. MacCallum, R.M., Mauch, M., Burt, A., Leroi, A.M.: Evolution of music by public choice. *Proc. Natl. Acad. Sci.* **109**(30), 12081–12086 (2012)
123. Machado, P.: Evolutionary art and design: representation, fitness and interaction. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '21*, pp. 1002–1031, New York, NY, USA (2021). Association for Computing Machinery
124. Machado, P., Cardoso, A.: NEvAr - the assessment of an evolutionary art tool. In: Wiggins, G. (ed.) *AISB'00 Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science* (2000)
125. Machado, P., Cardoso, A.: All the truth about NEvAr. *Appl. Intell.* **16**(2), 101–118 (2002)
126. Machado, P., Correia, J.: Semantic aware methods for evolutionary art. In: Arnold, D.V. (ed.) *Genetic and Evolutionary Computation Conference, GECCO '14, Vancouver, BC, Canada*, pp. 301–308. ACM (2014). Accessed from 12–16 July 2014
127. Machado, P., Correia, J., Romero, J.: Expression-based evolution of faces. In: Machado, P., Romero, J., Carballal, A. (eds.) *Evolutionary and Biologically Inspired Music, Sound, Art and Design - First International Conference, EvoMUSART 2012, Málaga, Spain, Proceedings. Lecture Notes in Computer Science*, vol. 7247, pp. 187–198. Springer (2012). Accessed from 11–13 Apr 2012
128. Machado, P., Correia, J., Romero, J.: Improving face detection. In: Moraglio, A., Silva, S., Krawiec, K., Machado, P., Cotta, C. (eds.) *Genetic Programming - 15th European Conference, EuroGP 2012, Málaga, Spain. Proceedings. Lecture Notes in Computer Science*, vol. 7244, pp. 73–84. Springer (2012). Accessed from 11–13 Apr 2012
129. Machado, P., Graça, F.: Evolutionary pointillist modules: evolving assemblages of 3D objects. In: Giacobini, M., Brabazon, A., Cagnoni, S., Caro, G.D., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A., Farooq, M., Fink, A., McCormack, J., O'Neill, M., Romero, J., Rothlauf, F., Squillero, G., Uyar, S., Yang, S. (eds.) *Applications of Evolutionary Computing, EvoWorkshops 2008: EvoCOMNET, EvoFIN, EvoHOT, EvoIASP, EvoMUSART, EvoNUM, EvoS-TOC, and EvoTransLog, Naples, Italy. Proceedings. Lecture Notes in Computer Science*, vol. 4974, pp. 453–462. Springer (2008). Accessed from 26–28 March 2008
130. Machado, P., Martins, T., Amaro, H., Abreu, P.H.: An interface for fitness function design. In: Romero, J., McDermott, J., Correia, J. (eds.) *Evolutionary and Biologically Inspired Music,*

- Sound, Art and Design - Third European Conference, EvoMUSART 2014, Granada, Spain, Revised Selected Papers. Lecture Notes in Computer Science, vol. 8601, pp. 13–25. Springer (2014). Accessed from 23–25 Apr 2014
131. Machado, P., Nunes, H., Romero, J.: Graph-based evolution of visual languages. In: Chio, C.D., Brabazon, A., Caro, G.A.D., Ebner, M., Farooq, M., Fink, A., Grahl, J., Greenfield, G., Machado, P., O’Neill, M., Tarantino, E., Urquhart, N. (eds.) Applications of Evolutionary Computation, EvoApplications 2010: EvoCOMNET, EvoENVIRONMENT, EvoFIN, EvoMUSART, and EvoTRANSLOG, Istanbul, Turkey, Proceedings, Part II. Lecture Notes in Computer Science, vol. 6025, pp. 271–280. Springer (2010). Accessed from 7–9 Apr 2010
 132. Machado, P., Pereira, L.: Photogrowth: non-photorealistic renderings through ant paintings. In: Soule, T., Moore, J.H. (eds.) Genetic and Evolutionary Computation Conference, GECCO ’12, Philadelphia, PA, USA, pp. 233–240. ACM (2012). Accessed from 7–11 July 2012
 133. Machado, P., Romero, J., Manaris, B.Z.: Experiments in computational aesthetics. In: Romero, J., Machado, P. (eds.) The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music, Natural Computing Series, pp. 381–415. Springer (2008)
 134. Machado, P., Vinhas, A., Correia, J., Ekárt, A.: Evolving ambiguous images. In: Yang, Q., Wooldridge, M.J. (eds.) Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, pp. 2473–2479. AAAI Press (2015). Accessed from 25–31 July 2015
 135. Machín, B., Nesmachnow, S., Toutouh, J.: Multi-target evolutionary latent space search of a generative adversarial network for human face generation. In: Fieldsend, J.E., Wagner, M. (eds.) GECCO ’22: Genetic and Evolutionary Computation Conference, Companion Volume, Boston, Massachusetts, USA, pp. 1878–1886. ACM (2022). Accessed from 9–13 July 2022
 136. Manaris, B., Hughes, D., Vassilandonakis, Y.: Monterey mirror: combining Markov models, genetic algorithms , and power laws an experiment in interactive evolutionary music performance. In: Proceedings of 1st Workshop in Evolutionary Music, 2011 IEEE Congress on Evolutionary Computation (CEC 2011), pp. 33–40 (2011)
 137. Manaris, B., Roos, P., Machado, P., Krehbiel, D., Pellicoro, L., Romero, J.: A Corpus-based hybrid approach to music analysis and composition. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, Vancouver, British Columbia, Canada, pp. 839–845. AAAI Press (2007). Accessed from 22–26 July 2007
 138. Manaris, B., Vaughan, D., Wagner, C., Romero, J., Davis, R.B.: Evolutionary music and the zipf-mandelbrot law: developing fitness functions for pleasant music. In: Raidl, G.R., Meyer, J., Middendorf, M., Cagnoni, S., Cardalda, J.J.R., Corne, D., Gottlieb, J., Guillot, A., Hart, E., Johnson, C.G., Marchiori, E. (eds.) Applications of Evolutionary Computing, EvoWorkshop 2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, and EvoSTIM, Essex, UK, Proceedings. Lecture Notes in Computer Science, vol. 2611 , pp. 522–534. Springer (2003). Accessed from 14–16 Apr 2003
 139. Manurung, H.M.: An Evolutionary Algorithm Approach to Poetry Generation . Ph.D. thesis, Institute for Communicating and Collaborative Systems, School of Informatics, University of Edinburgh (2003)
 140. Manurung, H.M., Ritchie, G.D., Thompson, H.S.: Towards A Computational Model Of Poetry Generation. Technical report, Institute for Communicating and Collaborative Systems, Division of Informatics, University of Edinburgh (2000)
 141. Manurung, R., Ritchie, G., Thompson, H.: Using genetic algorithms to create meaningful poetic text. *J. Exp. Theor. Artif. Intell.* **24**(1), 43–64 (2012)
 142. Martínez-Rodríguez, B.: A new fitness function for evolutionary music composition. In: Montiel, M., Agustín-Aquino, O.A., Gómez, F., Kastine, J., Lluís-Puebla, E., Milam, B. (eds.) Mathematics and Computation in Music - 8th International Conference, MCM 2022, Atlanta, GA, USA, Proceedings. Lecture Notes in Computer Science, vol. 13267, pp. 205–217. Springer (2022). Accessed from 21–24 June 2022
 143. Martins, T., Correia, J., Costa, E., Machado, P.: Evotype: evolutionary type design. In: Johnson, C.G., Carballal, A., Correia, J. (eds.) Evolutionary and Biologically Inspired Music, Sound, Art and Design - 4th International Conference, EvoMUSART 2015, Copenhagen, Denmark,

- Proceedings. Lecture Notes in Computer Science, vol. 9027, pp. 136–147. Springer (2015). Accessed from 8–10 Apr 2015
144. Martins, T., Correia, J., Costa, E., Machado, P.: Evolving Stencils for Typefaces: Combining Machine Learning, User’s Preferences and Novelty. *Complexity*, 2019 (2019)
 145. McCormack, J., Bown, O.: Life’s what you make: niche construction and evolutionary art. In: Giacobini, M., Brabazon, A., Cagnoni, S., Caro, G.A.D., Ekárt, A., Esparcia-Alcázar, A., Farooq, M., Fink, A., Machado, P., McCormack, J., O’Neill, M., Neri, F., Preuss, M., Rothlauf, F., Tarantino, E., Yang, S. (eds.) *Applications of Evolutionary Computing, EvoWorkshops 2009: EvoCOMNET, EvoENVIRONMENT, EvoFIN, EvoGAMES, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, EvoNUM, EvoSTOC, EvoTRANSLOG*, Tübingen, Germany. Proceedings. Lecture Notes in Computer Science, vol. 5484, pp. 528–537. Springer (2009). Accessed from 15–17 Apr 2009
 146. McCormack, J., Eldridge, A., Dorin, A., McIlwain, P.: Generative algorithms for making music: emergence, evolution, and ecosystems. In: Dean, R.T. (ed.) *The Oxford Handbook of Computer Music*, Oxford Handbooks. Oxford University Press (2011)
 147. McDermott, J., O’Reilly, U.: An executable graph representation for evolutionary generative music. In: Krasnogor, N., Lanzi, P.L. (eds.) *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings*, Dublin, Ireland, pp. 403–410. ACM (2011). 12–16 July 2011
 148. McIntyre, N., Lapata, M.: Plot Induction and evolutionary search for story generation. In: Hajic, J., Carberry, S., Clark, S. (eds.) *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, pp. 1562–1572. The Association for Computer Linguistics (2010). Accessed from 11–16 July 2010
 149. Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., Raju, B., Shahrzad, H., Navruzyan, A., Duffy, N., Hodjat, B.: Evolving deep neural networks. In: Kozma, R., Alippi, C., Choe, Y., Morabito, F.C. (eds.) *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, pp. 293–312. Academic (2019)
 150. Montero, C.A.S., Araki, K.: Is it correct? - Towards web-based evaluation of automatic natural language phrase generation. In: Calzolari, N., Cardie, C., Isabelle, P. (eds.) *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, Sydney, Australia. The Association for Computer Linguistics (2006). Accessed from 17–21 July 2006
 151. Moroni, A., Manzolli, J., von Zuben, F., Gudwin, R.: Vox Populi: an interactive evolutionary system for algorithmic music composition. *Leonardo Music J.* **10**, 49–54 (2000)
 152. Morris, R.G., Burton, S.H., Bodily, P.M., Ventura, D.: Soup over bean of pure joy: culinary ruminations of an artificial chef. In: Maher, M.L., Hammond, K.J., Pease, A., y Pérez, R.P., Ventura, D., Wiggins, G.A. (eds.) *Proceedings of the Third International Conference on Computational Creativity, ICCO 2012, Dublin, Ireland*, pp. 119–125 (2012). Accessed from May 30 - June 1, 2012
 153. Muehlbauer, M., Burry, J., Song, A.: automated shape design by grammatical evolution. In: Correia, J., Ciesielski, V., Liapis, A. (eds.) *Computational Intelligence in Music, Sound, Art and Design - 6th International Conference, EvoMUSART 2017, Amsterdam, The Netherlands, Proceedings. Lecture Notes in Computer Science*, vol. 10198, pp. 217–229 (2017). Accessed from 19–21 Apr 2017
 154. Neufeld, C., Ross, B.J., Ralph, W.: The evolution of artistic filters. In: Romero, J., Machado, P. (eds.) *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, Natural Computing Series, pp. 335–356. Springer (2008)
 155. Ng, A., Jordan, M.: On discriminative vs. generative classifiers: a comparison of logistic regression and Naive Bayes. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) *Advances in Neural Information Processing Systems 14, NIPS 2001, Vancouver, British Columbia, Canada, December 3-8, 2001*, pp. 841–848. MIT Press (2001)
 156. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. In: *IEEE Conference on Computer Vision and Pattern*

- Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, pp. 427–436. IEEE Computer Society (2015)
157. Nishio, K., Murakami, M., Mizutani, E., Honda, N.: Fuzzy Fitness Assignment in an Interactive Genetic Algorithm for a Cartoon Face Search, pp. 175–192. World Scientific (1997)
 158. Oliveira, H.G.: Automatic generation of poetry: an overview. In: 1st Seminar of Art, Music, Creativity and Artificial Intelligence (2009)
 159. Olseng, O., Gambäck, B.: Co-evolving melodies and harmonization in evolutionary music composition. In: Liapis, A., Cardalda, J.J.R., Ekárt, A. (eds.) *Computational Intelligence in Music, Sound, Art and Design - 7th International Conference, EvoMUSART 2018*, Parma, Italy, Proceedings. Lecture Notes in Computer Science, vol. 10783, pp. 239–255. Springer (2018). Accessed from 4–6 Apr 2018
 160. O’Reilly, U., Toutouh, J., Pertierra, M.A., Sanchez, D.P., Garcia, D., Lugo, A.E., Kelly, J., Hemberg, E.: Adversarial genetic programming for cyber security: a rising application domain where GP matters. *Genetic Program. Evolvable Mach.* **21**(1–2), 219–250 (2020)
 161. Parente, J., Martins, T., Bicker, J., Machado, P.: Which type is your type? In: Cardoso, F.A., Machado, P., Veale, T., Cunha, J.M. (eds.) *Proceedings of the Eleventh International Conference on Computational Creativity, ICCO 2020*, Coimbra, Portugal, pp. 476–483. Association for Computational Creativity (ACC) (2020). Accessed from 7–11 Sept 2020
 162. Pearce, M., Wiggins, G.A.: Towards a framework for the evaluation of machine compositions. In: *Proceedings of the AISB’01 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pp. 22–32. Society for the Study of Artificial Intelligence and the Simulation of Behaviour (AISB) (2001)
 163. Phon-Amnuaisuk, S., Law, E.H.H. and Kuan, H.C.: Evolving music generation with som-fitness genetic programming. In: Giacobini, M., Brabazon, A., Cagnoni, S., Caro, G.D., Drechsler, R., Farooq, M., Fink, A., Lutton, E., Machado, P., Minner, S., O’Neill, M., Romero, J., Rothlauf, F., Squillero, G., Takagi, H., Uyar, S., Yang, S. (eds.) *Applications of Evolutionary Computing, EvoWorkshops 2007: EvoCoMnet, EvoFIN, EvoIASP, EvoINTERACTION, EvoMUSART, EvoSTOC and EvoTransLog*, Valencia, Spain, Proceedings. Lecture Notes in Computer Science, vol. 4448, pp. 557–566. Springer (2007). Accessed from 11–13 Apr 2007
 164. Pirnia, A., McCormack, J.: Compressed multidimensional trees for evolutionary music representation. In: *Non-Cochlear Sound: Proceedings of the 38th International Computer Music Conference, ICMC 2012*, Ljubljana, Slovenia. Michigan Publishing (2012). Accessed from 9–14 Sept 2012
 165. Prisco, R.D., Zaccagnino, G., Zaccagnino, R.: Evobasscomposer: a multi-objective genetic algorithm for 4-voice compositions. In: Pelikan, M., Branke, J. (eds.) *Genetic and Evolutionary Computation Conference, GECCO 2010*, Proceedings, Portland, Oregon, USA, pp. 817–818. ACM (2010). Accessed from 7–11 July 2010
 166. Prisco, R.D., Zaccagnino, G., Zaccagnino, R.: EvoComposer: an evolutionary algorithm for 4-voice music compositions. *Evol. Comput.* **28**(3), 489–530 (2020)
 167. Rebelo, S., Bicker, J., Machado, P.: Evolutionary experiments in typesetting of letterpress-inspired posters. In: Cardoso, F.A., Machado, P., Veale, T., Cunha, J.M. (eds.) *Proceedings of the Eleventh International Conference on Computational Creativity, ICCO 2020*, Coimbra, Portugal, pp. 110–113. Association for Computational Creativity (ACC) (2020). Accessed from 7–11 Sept 2020
 168. Reddin, J., McDermott, J., O’Neill, M.: Elevated pitch: automated grammatical evolution of short compositions. In: Giacobini, M., Brabazon, A., Cagnoni, S., Caro, G.A.D., Ekárt, A., Esparcia-Alcázar, A., Farooq, M., Fink, A., Machado, P., McCormack, J., O’Neill, M., Neri, F., Preuss, M., Rothlauf, F., Tarantino, E., Yang, S. (eds.) *Applications of Evolutionary Computing, EvoWorkshops 2009: EvoCOMNET, EvoENVIRONMENT, EvoFIN, EvoGAMES, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, EvoNUM, EvoSTOC, EvoTRANSLOG*, Tübingen, Germany. Proceedings. Lecture Notes in Computer Science, vol. 5484, pp. 579–584. Springer (2009). Accessed from 15–17 Apr 2009

169. Reynolds, C.W.: Evolving textures from high level descriptions: gray with an accent color. In: Chio, C.D., Brabazon, A., Caro, G.A.D., Drechsler, R., Farooq, M., Grahl, J., Greenfield, G., Prins, C., Romero, J., Squillero, G., Tarantino, E., Tettamanzi, A., Urquhart, N., Etaner-Uyar, A.S. (eds.) Applications of Evolutionary Computation - EvoApplications 2011: EvoCOMNET, EvoFIN, EvoHOT, EvoMUSART, EvoSTIM, and EvoTRANSLOG, Torino, Italy, Proceedings, Part II. Lecture Notes in Computer Science, vol. 6625, pp. 384–393. Springer (2011). Accessed from 27–29 April 2011
170. Rodrigues, A., Costa, E., Cardoso, A., Machado, P., Cruz, T.: Evolving L-systems with musical notes. In: Johnson, C.G., Ciesielski, V., Correia, J., Machado, P. (eds.) Evolutionary and Biologically Inspired Music, Sound, Art and Design - 5th International Conference, EvoMUSART 2016, Porto, Portugal, Proceedings. Lecture Notes in Computer Science, vol. 9596, pp. 186–201. Springer (2016). March 30–April 1 2016
171. Romero, J., Machado, P. (eds.): The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music. Springer, Natural Computing Series (2008)
172. Rooke, S.: Chapter 13 - eons of genetically evolved algorithmic images. In: Bentley, P.J., Corne, D.W. (eds.) Creative Evolutionary Systems, The Morgan Kaufmann Series in Artificial Intelligence, pp. 339–365. Morgan Kaufmann (2002)
173. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, Barcelona, Spain, pp. 2226–2234 (2016). Accessed from 5–10 Dec 2016
174. Santos, A., Pinto, H., Pereira Jorge, R., Correia, N.: MuSyFI: music synthesis from images. In: Silva Garza, A., Veale, T., Aguilar, W., Pérez y Pérez, R. (eds.) Proceedings of the 12th International Conference on Computational Creativity, pp. 103–112. Association for Computational Creativity (ACC) (2021)
175. Saunders, R.: Artificial creative systems and the evolution of language. In: Ventura, D., Gervás, P., Harrell, D.F., Maher, M.L., Pease, A., Wiggins, G.A. (eds.) Proceedings of the Second International Conference on Computational Creativity, ICCO 2011, Mexico City, Mexico, pp. 36–41 (2011). Accessed from 27–29 Apr 2011
176. Saunders, R., Gero, J.S.: The digital clockwork muse: a computational model of aesthetic evolution. In: Proceedings of the Convention for Artificial Intelligence and Simulation for Behaviour (AISB) 2001 - Symposium on AI and Creativity in Arts and Science (2001)
177. Schnier, T., Gero, J.S.: Learning genetic representations as alternative to hand-coded shape grammars. In: Gero, J.S., Sudweeks, F. (eds.) Artificial Intelligence in Design '96, pp. 39–57. Springer, Netherlands (1996)
178. Schrum, J., Volz, V., Risi, S.: CPPN2GAN: combining compositional pattern producing networks and gans for large-scale pattern generation. In: Coello, C.A.C. (ed.) GECCO '20: Genetic and Evolutionary Computation Conference, Cancún Mexico, pp. 139–147. ACM (2020). Accessed from 8–12 July 2020
179. Scirea, M., Barros, G.A.B., Shaker, N., Togelius, J.: SMUG: scientific music generator. In: Toivonen, H., Colton, S., Cook, M., Ventura, D. (eds.) Proceedings of the Sixth International Conference on Computational Creativity, ICCO 2015, Park City, Utah, USA, pp. 204–211 (2015). Accessed from June 29–July 2 2015
180. Scirea, M., Togelius, J., Eklund, P., Risi, S.: MetaCompose: a compositional evolutionary music composer. In: Johnson, C.G., Ciesielski, V., Correia, J., Machado, P. (eds.) Evolutionary and Biologically Inspired Music, Sound, Art and Design - 5th International Conference, EvoMUSART 2016, Porto, Portugal, Proceedings. Lecture Notes in Computer Science, vol. 9596, pp. 202–217. Springer (2016). March 30–April 1 2016
181. Secretan, J., Beato, N., D Ambrosio, D.B., Rodriguez, A., Campbell, A., Stanley, K.O.: Picbreeder: evolving pictures collaboratively online. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '08, pp. 1759–1768. Association for Computing Machinery (2008)

182. Secretan, J., Beato, N., D'Ambrosio, D.B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J.T., Stanley, K.O.: Picbreeder: a case study in collaborative evolutionary exploration of design space. *Evol. Comput.* **19**(3), 373–403 (2011)
183. Shaker, N., Nicolau, M., Yannakakis, G.N., Togelius, J., O'Neill, M.: Evolving levels for super mario bros using grammatical evolution. In: 2012 IEEE Conference on Computational Intelligence and Games, CIG 2012, Granada, Spain, pp. 304–311. IEEE (2012). Accessed from 11–14 Sept 2012
184. Shao, J., McDermott, J., O'Neill, M., Brabazon, A.: Jive: a generative, interactive, virtual, evolutionary music system. In: Chio, C.D., Brabazon, A., Caro, G.A.D., Ebner, M., Farooq, M., Fink, A., Grahl, J., Greenfield, G., Machado, P., O'Neill, M., Tarantino, E., Urquhart, N. (eds.) *Applications of Evolutionary Computation, EvoApplications 2010: EvoCOMNET, EvoENVIRONMENT, EvoFIN, EvoMUSART, and EvoTRANSLOG*, Istanbul, Turkey, Proceedings, Part II. *Lecture Notes in Computer Science*, vol. 6025, pp. 341–350. Springer, 2010. Accessed from 7–9 Apr 2010
185. Sims, K.: Artificial evolution for computer graphics. In: Thomas, J.J. (ed.) *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1991*, Providence, RI, USA, pp. 319–328. ACM (1991). Accessed from 27–30 Apr 1991
186. Sims, K.: Interactive evolution of equations for procedural models. *Vis. Comput.* **9**(8), 466–476 (1993)
187. Smith, J.R.: Designing biomorphs with an interactive genetic algorithm. In: Belew, R.K., Booker, L.B. (eds.) *Proceedings of the 4th International Conference on Genetic Algorithms*, San Diego, CA, USA, pp. 535–538. Morgan Kaufmann (1991)
188. Spector, L., Alpern, A.: Criticism, culture, and the automatic generation of artworks. In: Hayes-Roth, B., Korf, R.E. (eds.) *Proceedings of the 12th National Conference on Artificial Intelligence*, Seattle, WA, USA, vol. 1, pp. 3–8. AAAI Press/The MIT Press (1994). Accessed from July 31–August 4 1994
189. Spector, L., Alpern, A.: Induction and recapitulation of deep musical structure. In: *Proceedings of International Joint Conference on Artificial Intelligence, IJCAI'95 Workshop on Music and AI* (1995)
190. Stanley, K.O.: Compositional pattern producing networks: a novel abstraction of development. *Genetic Program. Evolvable Mach.* **8**(2), 131–162 (2007)
191. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. *Evol. Comput.* **10**(2), 99–127 (2002)
192. Suganuma, M., Shirakawa, S., Nagao, T.: A genetic programming approach to designing convolutional neural network architectures. In: Bosman, P.A.N. (ed.) *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017*, Berlin, Germany, pp. 497–504. ACM (2017). Accessed from 15–19 July 2017
193. Sulyok, C., McPherson, A., and Harte, C.: Corpus-taught evolutionary music composition. In: Andrews, P.S., Caves, L.S.D., Doursat, R., Hickinbotham, S.J., Polack, F.A.C., Stepney, S., Taylor, T., Timmis, J. (eds.) *Proceedings of the Thirteenth European Conference Artificial Life, ECAL 2015*, York, UK, pp. 587–594. MIT Press (2015). Accessed from 20–24 July 2015
194. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, pp. 2818–2826. IEEE Computer Society (2016). Accessed from 27– 30 June 2016
195. Tanjil, F., Ross, B.J.: Deep learning concepts for evolutionary art. In: Ekárt, A., Liapis, A., Pena, M.L.C. (eds.) *Computational Intelligence in Music, Sound, Art and Design - 8th International Conference, EvoMUSART 2019*, Held as Part of EvoStar 2019, Leipzig, Germany, Proceedings. *Lecture Notes in Computer Science*, vol. 11453, pp. 1–17. Springer (2019). Accessed from 24–26 April 2019
196. Thywissen, K.: GeNotator: an environment for exploring the application of evolutionary techniques in computer-assisted composition. *Organised Sound* **4**(2), 127–133 (1999)

197. Ting, C., Wu, C., Liu, C.: A novel automatic composition system using evolutionary algorithm and phrase imitation. *IEEE Syst. J.* **11**(3), 1284–1295 (2017)
198. Todd, P.M., Werner, G.M.: *Frankensteinian Methods for Evolutionary Music Composition*, pp. 313–339. MIT Press (1999)
199. Todd, S., Latham, W.: *Evolutionary Art and Computers*. Academic (1992)
200. Togelius, J., Burrow, P., Lucas, S.M.: Multi-population competitive co-evolution of car racing controllers. In: *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2007, Singapore*, pp. 4043–4050. IEEE (2007). Accessed from 25–28 Sept 2007
201. Togelius, J., Karakovskiy, S., Koutník, J., Schmidhuber, J.: Super Mario evolution. In: Lanzi, P.L. (ed.) *Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Games, CIG 2009, Milano, Italy*, pp. 156–161. IEEE (2009). Accessed from 7–10 Sept 2009
202. Tomasic, P., Papa, G., Znidarsic, M.: Using a genetic algorithm to produce slogans. *Informatika (Slovenia)* **39**(2) (2015)
203. Toutouh, J., Hemberg, E., O’Reilly, U.M.: Spatial evolutionary generative adversarial networks. In: Auger, A., Stützle, T. (eds.) *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019, Prague, Czech Republic*, pp. 472–480. ACM (2019). Accessed from 13–17 July 2019
204. Toutouh, J., Hemberg, E., O’Neilly, U.M.: Analyzing the components of distributed coevolutionary GAN training. In: Bäck, T., Preuss, M., Deutz, A.H., Wang, H., Doerr, C., Emmerich, M.T.M., Trautmann, H. (eds.) *Parallel Problem Solving from Nature - PPSN XVI - 16th International Conference, PPSN 2020, Leiden, The Netherlands, Proceedings, Part I. Lecture Notes in Computer Science*, vol. 12269, pp. 552–566. Springer (2020). Accessed from 5–9 Sept 2020
205. Toutouh, J., O’Reilly, U.: Signal Propagation in a Gradient-based and Evolutionary Learning System. In: Chicano, F., Krawiec, K. (eds.) *GECCO ’21: Genetic and Evolutionary Computation Conference, Lille, France*, pp. 377–385. ACM (2021). Accessed from 10–14 July 2021
206. Turhan, C.G., Bilge, H.S.: Variational autoencoded compositional pattern generative adversarial network for handwritten super resolution image generation. In: *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pp. 564–568 (2018)
207. Unemi, T.: SBART 2.4: breeding 2D CG images and movies and creating a type of collage. In: Jain, L.C. (ed.) *Third International Conference on Knowledge-Based Intelligent Information Engineering Systems, KES 1999, Adelaide, South Australia, Proceedings*, pages 288–291. IEEE (1999). Accessed from 31 August–1 September 1999
208. Ventrella, J.: Evolving the mandelbrot set to imitate figurative art. In: Hingston, P.F., Barone, L.C., Michalewicz, Z. (eds.) *Design by Evolution: Advances in Evolutionary Design*, pp. 145–167. Springer, Berlin, Heidelberg (2008)
209. Vinhas, A., Assunção, F., Correia, J., Ekárt, A., Machado, P.: Fitness and novelty in evolutionary art. In: Johnson, C.G., Ciesielski, V., Correia, J., Machado, P. (eds.) *Evolutionary and Biologically Inspired Music, Sound, Art and Design - 5th International Conference, EvoMUSART 2016, Porto, Portugal, Proceedings. Lecture Notes in Computer Science*, vol. 9596, pp. 225–240. Springer (2016). Accessed from March 30–April 1 2016
210. Volz, V., Schrum, J., Liu, J., Lucas, S.M., Smith, A., Risi, S.: Evolving mario levels in the latent space of a deep convolutional generative adversarial network. In: Aguirre, H.E., Takadama, K. (eds.) *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018, Kyoto, Japan*, pp. 221–228. ACM (2018). Accessed from 15–19 July 2018
211. Wang, C., Xu, C., Yao, X., Tao, D.: Evolutionary generative adversarial networks. *IEEE Trans. Evol. Comput.* **23**(6), 921–934 (2019)
212. Webster, R., Rabin, J., Simon, L., Jurie, F.: Detecting overfitting of deep generative networks via latent recovery. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA*, pp. 11273–11282. Computer Vision Foundation/IEEE (2019). Accessed from 16–20 June 2019
213. Winters, T., Delobelle, P.: Survival of the wittiest: evolving satire with language models. In: de Silva Garza, A.G., Veale, T., Aguilar, W., y Pérez, R.P. (eds.) *Proceedings of the Twelfth*

- International Conference on Computational Creativity, México City, México (Virtual), pp. 82–86. Association for Computational Creativity (ACC) (2021). Accessed from 14–18 Sept 2021
214. L. World: Aesthetic selection: the evolutionary art of Steven Rooke. *IEEE Comput. Graph. Appl.* **16**(1), 4 (1996)
 215. Zaltron, N., Zurlo, L., Risi, S.: CG-GAN: an interactive evolutionary gan-based approach for facial composite generation. In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020*, New York, NY, USA, pp. 2544–2551. AAAI Press (2020). Accessed from 7–12 Feb 2020
 216. Zhang, J., Taarnby, R., Liapis, A., Risi, S.: Draw compile evolve: sparking interactive evolutionary art with human creations. In: *Johnson, C.G., Carballal, A., Correia, J. (eds.) Evolutionary and Biologically Inspired Music, Sound, Art and Design - 4th International Conference, EvoMUSART 2015*, Copenhagen, Denmark, Proceedings. *Lecture Notes in Computer Science*, pp. 261–273, vol. 9027. Springer (2015). Accessed from 8–10 Apr 2015
 217. Zoric, V., Gamback, B.: the image artist: computer generated art based on musical input. In: *Pachet, F., Jordanous, A., León, C. (eds.) Proceedings of the Ninth International Conference on Computational Creativity, ICC 2018*, Salamanca, Spain, pp. 296–303. Association for Computational Creativity (ACC) (2018). Accessed from 25–29 June 2018