Generative Music Evaluation: Why do We Limit to 'Human'?

Róisín Loughran and Michael O'Neill *

NCRA, University College Dublin, Ireland
roisin.loughran@ucd.ie

Abstract. Objective evaluation of aesthetic subjective judgements is, by very definition, tricky business. This may be the reason many generative music systems merely skip the evaluation part entirely. Typical evaluation systems that do exist are often variations of a Turing-style discrimination test whereby the autonomous system must convince a human interrogator that what it has produced was created by a human. In this paper we propose that this may be selling the computational systems short. With the ever increasing power of computational machines, why should we limit these new intelligent systems to a human level of creativity we barely understand ourselves? We consider that autonomous, statistical evaluations would be superior to the traditional human judgement tests. We describe a number of evolutionary systems that have been applied to compositional tasks and propose that these are the most suitable methods to use in developing autonomous evaluation measures.

1 Introduction

One of the most comforting aspects of music is that it is personal. People have their own taste in music, not just music scholars and academics, but all people. From a very young age we prefer certain songs, melodies or musical styles over others. These develop as we grow, impacting not merely what we choose to listen to but often our clothes, friends or lifestyle. If music is this personal, this influential, this inherently *human* then can we actually expect computer programs to produce music that is as beautiful, or meaningful as a skilled human being? If we're being truly honest — do we actually want them to? Do we want these unfeeling bunch of circuits and wires to be able to appreciate or replicate this art form we alone understand — wouldn't it more be comforting to always be able to spot real human music from 'Computer Music'?

Over the past few decades, major advances have been seen in relation to computer programming in the fields of Artificial Intelligence (AI) and Machine Learning (ML). We have seen a program beat a human champion in chess [1] and just this year a program has beaten a professional human in arguably the oldest human board game in the world, Go [2]. The ultimate mark of intelligence

* This work is part of the App'Ed (Applications of Evolutionary Design) project funded by Science Foundation Ireland under grant 13/IA/1850

has been an 'us against them' competition. Almost since Alan Turing developed his first computer he advocated the famous Turing Test (TT) to determine if a computer could fool a human interrogator into thinking it itself was human. From this initial stage of testing computers, this has been the goal — for a machine to trick us into thinking it is *one of us*. In recent years, as AI techniques have been applied to creative tasks such as music composition, the natural progression has been the development of a type of creative or Musical TT. Such tests are used to evaluate a system, deeming that if a listener cannot determine if a composition was written by a human or machine than the machine has won. Such arguments are, however, deeply flawed [3].

Regardless of how this is argued, we (as in humans) appear to be fixed on this idea that the only way to judge merit in a computational, autonomous method is for it to fool ourselves. Assuming that the average human mind, as it is currently evolved, is not at the upper limit in possible intelligence, this paper proposes that this fascination in creating systems that are *as good as us* may be limited and misguided.

The following section offers some background in the field of Computational Creativity and discusses how music is a creative process. A number of generative compositional methods are discussed in Section 3 with a focus on evolutionary computational methods. The fitness measures used in these experiments is contrasted against the more specific evaluation of the end results of the process as discussed in Section 4, including Turing Tests, the Lovelace test, Crowd Sourcing and alternative methods. We discuss the implications of this research and draw some conclusions in Sections 5 and 6.

2 Music as a Computationally Creative Process

Often known as a founder of computer programming, Ada Lovelace had some remarkable insights into the possibilities that computer programming could offer us. In the 1840s, Lovelace saw a capability in Charles Babbage's recently proposed Analytical Engine far greater than that of mere numerical manipulations. She saw that such machines could in time be used to represent art and music. Furthermore, she considered the possibility of these computers playing a role in the study of creativity. This foresight was remarkable in relation to such a new theoretical invention at the time. Nevertheless, her vision is now coming to fruition as the topic of Computational Creativity (CC) has emerged as an exciting field in the past 20 years describing computer systems that are creating music, art and literature [4].

The current definition of a CC is the philosophy, science and engineering of computational systems which exhibit behaviour deemed to be creative by an unbiased observer [5]. This working definition and many others like it are based on the idea of computers exhibiting human-like behaviour as judged by other humans. As creativity in itself is still such a difficult quality to define or distinguish even within the human mind, it is unsurprising that we have little alternative but to measure artificial creativity in terms of human recognition.

Wiggins defends the circular nature of the CC definition by stating '[it] aims to capture a concept or set of concepts which is difficult to define intensionally by means of explicit rules in terms of its extension into the real world of human behaviour'. Thus the inherent difficulty in defining human creativity is inevitably transferred to the the domain of computational creativity.

Boden has stated three ways in which computers can attempt to create new or creative ideas: by combining novel ideas, exploring the limits of conceptual spaces or by transforming established ideas that enable the emergence of previously unknown or impossible ideas [6]. While she advocates that transformational methods hold the potential for most 'shock value' she also concedes that these are the most difficult methods to evaluate, as the transformations make meaningful interpretation or evaluation criteria very difficult to define.

One may wonder then why computation is applied to issues in music and creativity at all. This motivation was examined and discussed in detail in [7] whereby they determined four distinct reasons for applying computation to compositional tasks, namely algorithmic composition, design of compositional tools, computational modelling of musical styles and computational modelling of music cognition. Clearly there is more to be learned by applying algorithms to compositional tasks than merely creating computer music, although arguably algorithmic composition is still the most creative of these tasks. In discussing the motivations and evaluation of the compositional aim however they determine that 'researchers often fail to adopt suitable methodologies for the development and evaluation of composition programs and this, in turn, has compromised the practical or theoretical value of their research.' Thus a fundamental issue in applying computational methods to composition lies in the evaluation of the systems created.

Algorithmic Composition (AC) can be considered a computationally creative task, but only if the compositions created display true originality and creativity. Systems that merely mimic or adapt previously composed music would not, on the surface, appear to be creative. In saying that, David Cope has stated that there is no new music to be written — merely existing snippets to be rearranged [8]. Cope's algorithmic compositional system EMI (Experiments in Musical Intelligence) was created to generate music in a given style and was trained on a corpus of existing music, initially a set of Bach chorales. He developed this system further into Emily Howell, an algorithmic composer who has released albums in her own style. While Cope insists that there is no new music, and therefore music composed by Howell is not novel, he explains that creativity is not the same as novelty. Cope defines creativity as a new linking between ideas; that creativity is based on the initialisation of a connection between ideas not already considered connected.

3 Generative Music

In recent decades a large number of studies have applied numerous ML methods to the task of music composition. A full review of such methods is beyond the

scope of this paper, but the interested reader can find a detailed survey in [9]. In this section we examine the continuous within-system evaluation that is necessary to drive such systems, focussing on a branch of nature inspired algorithms know as Evolutionary Computation (EC) [10].

3.1 Evolutionary Music

EC methods are fundamentally based on Darwin's evolutionary theory of 'survival of the fittest'. A population of random solutions to a given problem are created and each solution is assigned a *fitness* according to how well it solves that problem. The solutions are then selected for survival and reproduction into the next generation based on this fitness. As this process is repeated, the overall population of solutions is improved and the best in the final population can be chosen as the solution to the given problem.

EC methods were developed using problems that had a specific optimal solution such as symbolic regression and the artificial ant trail. In developing these systems for aesthetic purposes, we should perhaps look at a broader way of using and interpreting them. These are tools for us to use as composers, and as our tools we can utilise them in whatever way we see fit. Miranda examined three distant approaches to using evolutionary methods in music: the engineering approach uses EC techniques in the field of sound synthesis, the creative approach uses EC in compositions and the musicological approach which searches for the origins of music by means of computer simulations [11]. An overview of earlier studies in EC for musical composition is offered in [12], determining that Genetic Programming (GP) methods perform better than those that use Genetic Algorithms (GA). This may be unsurprising as GP methods use a tree-based structure whereas GAs are limited to a linear string in their representation. Hence, GP can represent more complex representations and operations — something that would be very useful in representing music. Dahlstedt has discussed how we may use EC as the basis of a wide range of tools but that in doing so we may have to relinquish some level of control [13]. Evolutionary processes work well in aesthetic tasks such as music composition as they are generally non-deterministic. The evolution of a population offers so much scope and possibility that it is reminiscent of the music creation process — a solution is not linearly determined but instead emerges from a fluid, incremental process. The biggest issue in using EC for aesthetic purposes is in the design of the fitness measure. Individual solutions (compositions in the case of AC) can only survive on to the next generation if they are judged worthy according to a predetermined fitness measure designed by the programmer. Thus the problem becomes how do we measure the musical fitness of the individual?

3.2 Measuring Fitness

The most obvious approach to developing an aesthetic judgment based fitness measure is to use a human as the fitness function. Such systems are referred to as

Interactive EC (IEC). In these experiments a human user must rate each individual in every given generation. The survival of that individual is then dependent on the value given by the user. These systems are very well suited to design and creative tasks as they remove the need to automate a subjective judgment. A number of systems have used IEC to successfully create melodies [14–16]. The biggest drawback with interactive methods is that they create a bottleneck, particularly in musical tasks. For the analysis of art, whereby the user can observe a number of creations concurrently, the fitness can be measured very quickly. For musical tasks however, users need to listen to musical excerpts successively, rendering these methods very expensive. For IEC experiments in algorithmic composition, the experiments must be designed so that the user only has to listen to and adjudicate a small number of compositions before fatigue or boredom sets in. Every time an experiment is run a new set of listening tests (possibly with a new set of listeners) must be set up. This makes it very cumbersome to re-run experiments and so IEC experiments must be very carefully prepared. For this reason it is simpler and less costly to develop an automatic fitness function.

Some studies work with the idea that if the initial population only contains individuals that are already of high quality then you can either randomly select any individual for reproduction, or use the entire population [17–19]. The idea of a random fitness function is alien to EC programmers as it is non-sensical to evolve a population without any fitness measure. If the system uses a priori musical knowledge to ensure the entire population is of high fitness, then the search space is confined so that the evolutionary process can be used to traverse the space safely. This may not be considered a proper use of EC — but it can make good music.

The use of a traditional, autonomous measure of fitness may be more economical than IEC and make more sense than random selection but such a measure is not easy to define. An overview of the most prevalent measures and ideas used to examine and evaluate melodies is given in [20]. They discuss ten attributes used in the evaluation of melodies based on pitch and rhythm measurements, concluding that previous approaches to formalise a fitness function for melodies have not comprehensively incorporated all measures. Nevertheless, many studies have used various type of autonomous fitness functions to drive EC systems to create music [21–24].

3.3 What's the Objective?

The above argument only considers EC applications but other ML music creation systems suffer from the same dilemma. Any supervised machine learning algorithm needs an error function — a target which it must aim towards. Backpropagation, used in Artificial Neural Networks such as the Multi-layered Perceptron requires a mean-squared error, which requires a target. Similarly any other supervised ML algorithm needs an error function — a target which it must try to approach or optimise towards.

Such targets are completely mis-aligned with the human method of composing however. Human composers do not start with a target composition and

iterate towards that. Students of academic music may be given assignments in which they must conform to a set of theoretical rules or emanate a given composers style — but this is not where great compositions come from. Is the purpose of applying AI to music to produce a bunch of mediocre students or to create new genuinely good and novel music?

One problem with traditional fitness functions is that they result in good or bad results, leading to a scale of 'goodness' depending on how close an individual is to a specified objective. Some AI researchers would propose that using a pre-specified objective is not necessarily a good idea when searching a space to solve a problem. This theory suggests that searching for novelty is a better method in looking for a great solution, that the optimal solution can often be found when looking for a different solution or when searching for no particular solution at all [25, 26]. Such a theory fits very well in searching any creative space. A musician does not know what music they are trying to create when they start, they work through ideas, changing their process and hence their output as they observe what they are creating. We propose that for any automated machine learning system to be truly creative there cannot be a pre-defined objective, the fitness function should be a measure of the progress of the system.

In recent years, the field of CC has embraced this idea that creating an artefact means more than outputting a number. The context within which a creative product is judged, including background information and the feeling it evokes in the creator is defined as Framing [27]. Such a concept reveals that there is more to CC than the output, and that intent, motivation and aspect of the creative or computational process all contribute to the overall result. Similarly, a Computational Creativity Theory (CCT) has been proposed to provide a computationally detailed description of how creation could be generated and the impact it can have [28]. These studies demonstrate that there is more to measuring the progress of a creative system than merely taking a numerical measure of error, target or fitness.

In the case of using EC techniques for compositional tasks we must be very clear on the distinction between *fitness measure* and *evaluation*. The fitness is the continuous measure taken from individuals within the population that drives the evolution of the composition. Evaluation in this sense refers to the measure of the performance of the system as a whole — how successful is the given system at composing a piece of music. In creative tasks such as music creation, this results in a distinct disjoint between fitness measurement and the perceived quality of the output — one that is not present in more traditional, empirical uses of EC. We highlighted EC applications to music creation above as this fitness measure plays a crucial role although many other types of machine learning methods have been applied to the task of music composition [9]. Regardless of the type of algorithm used, with any optimisation or error-based functionality, some metric of the aesthetic progress of the melody must be given throughout the composition process. This is not the same as evaluation however. Evaluation involves measuring the overall success of the system either from the process involved or the final result produced.

4 Evaluation

When looking at CC systems we want to ensure that there is enough creativity present for truly original output but we need some tangible way of measuring this creativity. The lack of evaluation in CC systems has been noted in recent years [29]. Such studies highlight the need for a clear definition of what can be considered creative. Colton designed a framework entitled the Creative Tripod to determine if a system is creative, or if it merely has the perception of being creative [30]. The Tripod framework describes a system as creative if it exhibits three elements — skill, appreciation and imagination. Furthermore, the framework states that there are three involved parties that may be perceived as contributing to this creativity namely the programmer, the computer and the consumer. For creativity to be experienced all three elements must be exhibited by at least one of these three parties. This is an extremely important step in the description and definition of creativity as it can separate the idea of creativity from the human user. If a programmer shows no creativity but the program she creates does, then creativity is present.

In contrast to this, one of the Open Problems in Evolutionary Music and Art [31] states that it is important to create evolutionary *music* and not just concentrate on the interesting method that created these sounds. While the final output of a musical process is arguably the most important aspect of any computational system, we should not disregard the methods used to achieve them as insignificant or an irrelevant means to an end. Defining the merit of music by the process that created it is not a modern day phenomenon. The serialist works of Schoenberg are remembered as much for the manner in which they were composed as the output that they produced. Likewise, much Musique Concréte works are as focussed on the way in which the sounds within a piece are made as the final output. Simply put, if the method is of interest then that in itself gives merit to the system and hence the music it produces.

Despite the problems in evaluating generative music, a number of studies have tried to propose systems that include a formal evaluation. A framework for evaluating genre-specific compositions was proposed in [32]. In this work they describe a framework that examines each phase of a generative music system culminating in a discrimination test. This evaluation was performed by human subjects by asking them how well the generated music conformed to a pre-specifed genre. Pearce et al use a subsequent study to evaluate melodies with a learning-based perceptual model of music listening [33]. This study involved using a number of experienced human observers to judge the output of three computational methods of creating chorales, and then statistically analysing their judgements in order to help develop towards an autonomous creative system. This work proposes an excellent study in modelling a computational system on measured cognitive behaviour, however they acknowledge that their results suggest that these compositional tasks still present significant challenges in modelling cognitive processes.

A further discussion of various methods of evaluation applied to musically creative systems is given in [3]. They discuss the Musical Directive Toy Test

(MDtT) whereby an interrogator, using a computer interface, gives musical directive to two composers — one human and one machine. The given directive may be a style or abstract instruction and the interrogator must decide which output is from the human. A similar Musical Output Toy Test (MOtT) is described whereby two composers (again one human one machine) produce a piece of music that may be related in terms of style or instrumentation but are created without specific directive. Again the goal is to convince the interrogator that they are the human composer. They compare the application of these tests in numerous studies but note that these tests, unlike the traditional TT, do not rely on or require natural language, and that the decision made by the interrogator may rely as much on preference or subjective judgments as on logic. They propose that the continued use of such tests does more to 'investigate the limits of musical judgement than the innovation of generative music systems'.

4.1 The Lovelace Test

As discussed in Section 2 Ada Lovelace saw a creative potential in the development of computational machines. She posed a number of questions in this regard which have been distinguished by Boden into the four *Lovelace Questions* [4]. These questions ask:

- Can computational ideas help us understand human creativity?
- Can computers ever do things that appear creative?
- Can computers ever recognise creativity?
- Can computers ever *really* be creative?

Most people would agree that the first two questions have been answered (with a resounding 'Yes'). The third may offer more argument, but it is the fourth question that causes the most bother to people. Bringsjord et al consider that Lovelace posed these questions as an objection to the idea that computers could actually be creative. They note her objection that creation requires the *origination* of something whereas computers are not capable of originating anything [34]. They subsequently developed the aptly named Lovelace Test (LT) for creativity. This test involves an artificial agent A, its output o and its human architect H. Simply put, the test is passed if H cannot explain how A produced o. While this may seem like a simple criteria, it is actually extremely difficult to pass. This test requires that the algorithm written by the programmer must produce an output that the programmer, or another agent with the programmer's expertise, cannot explain. On the surface it may seem like many AC systems would quickly pass this. EC compositional systems for example (see Section 3.1), having a non-deterministic nature, can produce output not-predictable by the programmer. Not predictable is not the same as not explainable however. The programmer can explain the representation, fitness measures or grammars used in such systems, thus explaining the process of how the music is produced. For the LT to be passed, the output must be truly surprising and unexplainable to the programmer.

The LT is much more difficult than other TT style tests as it is the programmer, the one person who understands the workings of the algorithm more than anyone else, that acts as the interrogator of the system. In a sense the program must fool or trick its own creator for it to be deemed successful. If the programmer made a mistake, and suddenly could not remotely explain the output of their own system, would this be allowed to pass the LT? We would assume not, since a mistake implies randomness (on the programmers part) and randomness is not equivalent to creativity. However, if the seemingly random human mistake led to a genuine creative streak — shouldn't this satisfy the specified criteria to pass the LT? Often our own most creative successes are attributed to a moment of inspiration. Could this not be seen as a 'mistake' in the mind that we cannot explain? If we can accept the results of our own random mistakes as creative, why does it need so much more explanation in the programs we create and, paradoxically, why is it that once we can explain it it no longer can be claimed as creative?

The LT can be seen as an attempt to satisfy the fourth Lovelace Question posed above, and therein lies the difficulty. To *really be creative* is something that many humans feel they can only aspire to. This is partially in response to a misunderstanding or lack of clarity as to what is meant by creativity. Colloquially when we use the term 'creative' it is often in response to an artistic quality, some people may even use it in place of 'talented'. Thus there is a magical element to the idea of creativity that many feel should not be possible to 'mimic' by computers. But creativity is not magic; its not an elite quality only to be found in a lucky few. We all have creative capabilities. Just because we are not professional artists, poets or musicians does not mean we do not have creative capabilities. The LT may be doomed to be unpassable — by definition if the programmer understands their own code, they can always offer some explanation as to the output that is produced. As algorithms become more complex however, involving domain transformations, stochastic, statistical and non-deterministic measures then surely this explanation will become a more abstract way of explaining how the output came about — rather than an exact explanation of how A produced o. Human artists are not held up to such scrutiny as to how they create a work of art. Critics may examine an artist through their teachers, mentors and influences determining their reasoning for a given style according to what they have learned along their career path. This explanation of influences or learning does not negate the resultant creativity of a human artist. Why should such an explanation automatically negate the creativity of an algorithm?

4.2 Crowd Sourcing: The Opinion of the Masses

With online resources it is now quite simple to create music or listening tests and merely ask your online audience to evaluate and give you feedback. The problem with such activities is that they are often shared among our peers, those that have an interest in computer generated music or who study or work directly with it. This means that the your audience is not a typical representation of the public at large. Music Technology students and practitioners know what they

are looking for when asked to listen to music and bring assumptions to the task that others may not. In a simple instrument recognition listening test, given to general candidates and Music Technology students, a number of the students reported that they could 'hear' the sounds that were synthesised even though all notes were real recordings of instruments and they were not asked to identify any synthesised instruments [35, 36]. This strongly indicates that peoples perceptions of what they hear are largely influenced by their pre-conceptions of what they are expecting to hear.

One way around this would be to play computer generated music to an unsuspecting live audience such as was undertaken by Cope [8]. To get a measurable response however, you must ask the audience for an opinion, and they may resent you for it. It's not a long term feasible plan to ask audiences to participate in questionnaires when they have had planned — and paid money — to simply listen to music. In addition, audiences that have been questioned once at a given concert may be expecting to be asked again, and then not listen passively for enjoyment of the music but instead actively listen for tell-tale signs that the music is computer generated.

4.3 Alternative Methods of Evaluation

In this paper we criticise the use of human-standard evaluations for computational creative processes. If we were to take this on board, the next question of course is 'What's the alternative?' Unfortunately this may not be an easy question to answer. If we must throw away any notion of human-comparisons then we must find a way of adjudicating the output of a proposed system. Deterministic rule-based methods may work but are unlikely to genuinely portray creativity, or if they did — how would we measure it?

Boden concluded that domain expertise and evaluation of systems are the major bottlenecks in CC systems [6]. These two aspects are clearly linked if we are to evaluate single artefacts according to a deterministic quality. If we are comparing two musical compositions to determine one better than the other, then we need to know much about musical theory. Depending on the system, we may need specific criteria for a given genre or style of music that is to be preferred. If this is too restrictive, the problem reverts back to the objective fitness problem discussed in Section 3.3 whereby a target composition is approached. If the criteria are too relaxed, then it becomes difficult to hold any control over the system and it may reduce to chaotic or random behaviour. Ideally, there should be a defensible, statistical method to determine one artefact better than another that does not rely on the specifications of the given problem.

A common metric for evaluating creativity across disciplines based on the three criteria of novelty, value and unexpectedness was proposed in [37]. Although they acknowledge that individually these measures are not novel, they propose that a combination of such metrics is what is necessary to measure true creativity.

A notable recent study demonstrated that in CC systems, it is only important that the decision of fitness need be defensible; what makes one creative

item better than another may not be what a human would choose but it must be a sensible, defensible and reproducible choice by the computer program. This was investigated using the the idea of a preference function by measuring qualities such as specificity, transivity and reflexivity to determine the choice of a system in a number of subjective tasks [38]. This preference function chooses one item over another due to a logical system of comparing between items and determining a decisive preference. Such a measure may not agree with what a human may choose as the best but, most importantly, it agrees with itself. A very interesting method of evolving music using the idea of 'sociability' was proposed by Miranda in [39]. In this study agents were created who had repertoires of melodies. The sociability was measured in terms of similarity of the agents repertoires; individual melodies could survive or be altered depending on reinforcement feedback between co-evolving agents. In a similar concept to this, we are currently working on an experimental system that judges musical conformity or agreement among a population of evolved critics. The fitness measure is based on correlation between each agent's opinion and the overall opinion of the population, rather than on a similarity measure. The overall system is evaluated, not directly on an aesthetic value of the output but on how the diversity of the population of melodies changes as more and more are produced recursively by the system.

These proposed methods are methods of continuous evaluation or fitness as described in Section 3.2 rather than pure evaluation of the output of the system. Systems such as these, however, do offer a platform for developing and investigating such measures. Evolutionary computational systems can be analysed and assessed as they create artistic works such as music, using such non-deterministic, domain-independent objective measures. Furthermore, EC systems such as Grammatical Evolution use grammars that allow transformation of the data being analysed, resulting in further scope in the complexity and search space allowable to the algorithm. This is the main reason we feel that evolutionary systems offer a huge potential in investigating aesthetic, objective evaluations within the field of CC.

We advocate that systems such as these that rely on non-human evaluations may point the way forward to more genuinely creative music generation systems. We are sure that critics may say that such systems may not be ready to actually create music that is any good (at least not according to fellow humans) but is that really relevant? If we could pull out the criteria to satisfy humans couldn't we open the field to higher possibilities that have not been achieved by human minds. We cannot assume that AI understanding or thought processes work in the same manner as a human mind; the human mind is an extremely complex system that spans several fields of study and our understanding of AI is still really in its infancy. Maybe in part we could consider putting our own judgements aside and letting the machines *run with it* for a while.

We already have humans that can create beautiful music we love; we have systems that can 'create' music that we love; this year saw the first commercial musical created using AI techniques [40]. Beautiful music can be created when

we get to say what's beautiful. A more interesting system is one that can discover itself what it believes to be beautiful — not what it is told is beautiful. Boden came to a conclusion that 'The ultimate vindication of AI-creativity would be a program that generated novel ideas which initially perplexed or even repelled us, but which was able to persuade us that they were indeed valuable' [41]. To even begin to achieve this we must take away or at the very least diminish the importance of human evaluation on computational output.

5 Discussion

Boden warns against 'The super human fallacy' whereby we shouldn't say that an AI has failed just because it cannot match the maximum heights of human intelligence [41]. We do not consider our own intelligence to have failed if it does not match up to the greatest human minds in every aspect. In contrast to this we also do not want to fall victim to associating too much intelligence to emergent behaviour that we may witness but not truly understand. Such tendencies amount to anthromorphizing the machine, sometimes known as the Eliza effect [42]. We must be careful not to project our own understanding of the outcome of computational tests to infer a greater understanding than that which is actually present within the machine. Such assumptions are easy to fall into, as when we see a behaviour that we wish to see — we naturally assume that the machine *meant to do it*. Such meaning or intent requires self-awareness however. This is reminiscent of early arguments against machine intelligence by Jefferson whereby he stated that intelligence is only present if the machine not only created something but was aware that it had created it [43]. This idea of self-awareness is still paramount to our understanding of CC and AI in general. The concept of Zombanimals illustrate the view that current AI practices do not in fact approach levels of conscious thought or intent as capable by a person [44]. Can an AI agent produce thought or can it merely trick us into thinking it is thinking? If it can trick us well enough, would we even know or should we even care?

The LT was proposed to combat TT attempts that are merely progressing due to 'the strength of clever but shallow trickery' [34]. They advocate that passing such a test is merely an attempt by the programmers to trick a naive human observer and is not a true measure of intelligence. While this notion of 'trickery' seems fiendish or almost non-scientific, other studies admit upfront that it is an acceptable outcome to expect. Ellis et al for example stated that computing machines can't be genuinely creative in the musical field — but that tricking humans into thinking they are is sufficient and possible in the coming years [45]. While this is a pragmatic outlook it also seems to admit failure before even setting out. If true CC is impossible — then why is there still so much discussion on it? It we consider it possible, then why are we so negative about the prospects of it occurring? What if instead of taking the argument that an artificially created artefact must be proven creative, we declared it to be creative and demanded a proof to the contrary?

5.1 Why do we fear the machines?

What if as an experiment we consider the AI mind, not to be a computer, but to be another 'type' of human mind, one with the same capabilities as our own. Suddenly our views as to how they must prove their intelligence or creativity could be considered quite prejudiced or discriminatory. The manner in which we decide how 'good' an AI is, is by its ability to fool us into thinking its one of us. In this we are inherently assuming our own superiority. There is a comfort in thinking that while they can compute faster than us, they will never be as smart as us for they do not know what being part of 'us' is. But ultimately isn't this our fault? Nearly all fictional works that involve the development of AI end in them rising up against us, either by outsmarting us or outgrowing us. Is this in response to some inherent fear we have that because we have cut them off from the more beautiful things in life that we inevitably create psychopathic entities that are wholly out for self-preservation? If so, is this a prediction or a, possibly subconscious, view that even if they over-power us it is only if they become 'bad' and therefore they are still at least morally inferior to us, their empirical creators.

In a sense the LT is searching for a 'Ghost in the Machine' as if looking for some spark that is unexplainable to our human minds. The problem is that we as humans do not always like what we do not understand. Technology that cannot be explained is seen as magical. So are we looking for divine inspirations within the algorithm — do we want to be awed? But this is unfair as human counterparts are not subjected to such rigorous examination. We rarely (ever?) demand a professional artist or musician to explain exactly what their process is and how they came up with it. We do however often subject students to such rigour; is this another example of our feeling of superiority to AI Agents? We treat them as students, never masters unless they can fool us into thinking they are one of us. This in turn leads to a 'Fear of the Imposter' idea — we gauge AI agents not as to how smart they are but as to how like us they are, or more accurately how like us they appear to be. This lends credence to the idea propagated by the world of sci-fi that we fear these new AI beings. And what we fear, we need to control but first and foremost — we need to identify.

6 Conclusion

Recent work in CC has called for better evaluation of proposed systems. In subjective, aesthetic domains it is very tempting to continue to use human judgement to validate a system, but by continuing these human evaluations we may never allow an autonomous system to reach its full creative capacity. What if the human observer cannot recognise the greatness within the machine? If we dismiss artefacts as soon as we don't like them or find them uncomfortable, how can we expect a CC system to actually surprise us?

This paper discussed evaluation techniques applied to generative music systems. We looked at Music as a CC process and considered the evaluation that must be carried out as part of this generative process. We considered a number

of EC applications, as such methods offer a wide search space and the capability to examine and analyse the system as it develops. We propose that this kind of analysis of the system as it progresses is better and more meaningful in the field of CC than merely taking a human evaluation of the system. Human evaluation is prone to bias, personal opinion, mis-understanding, assumptions, prejudice and fatigue. It is also fundamentally limited in the assumption that 'human is best'. What if humans can't even appreciate the best at the moment?

On the other hand, if this music is not to be judged by humans, only computers — would this suggest that it is actually written *for* computers? Arguably, one day AI agents may be the target audience, but we are not near this yet. We are not arguing as to who music is for, but rather for an alternative way of measuring the success of artificial agents that create music — either by evaluating their output or their systems.

In reality, human judgement tests are likely to still prevail as methods of evaluation for some time yet. Comparing a newly developed system against our own judgements is natural and will remain the evaluation of choice for many CC practitioners until better alternatives have been developed and become commonplace. What we would like to emphasise is that it is vital that these are not the only evaluation tests being carried out. We are at an exciting phase in CC and AI research whereby we can shape the future development of these fields. The manner in which we evaluate and test our progress should always be itself debated and re-evaluated. What we have proposed throughout this study is that we should not necessarily limit the perceived success of autonomous creative agents to what we humans perceive as good. While we do not want to fall victim to the super human fallacy, we also don't want to limit the potential of these systems to sub-human either. Only success measured independent of human opinion or capabilities will allow autonomous systems to progress past our own human limitations.

Acknowledgments

This work is part of the App'Ed (Applications of Evolutionary Design) project funded by Science Foundation Ireland under grant 13/IA/1850. We would also like to thank the reviewers for their thoughtful comments and suggestions.

References

1. Campbell, M., Hoane, A.J., Hsu, F.h.: Deep blue. Artificial intelligence **134** (2002) 57–83
2. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. Nature **529** (2016) 484–489
3. Ariza, C.: The interrogator as critic: The turing test and the evaluation of generative music systems. Computer Music Journal **33** (2009) 48–70

4. Boden, M.A.: The creative mind: Myths and mechanisms. Psychology Press (2004)
5. Colton, S., Wiggins, G.A., et al.: Computational creativity: the final frontier? In: ECAI. Volume 12. (2012) 21–26
6. Boden, M.A.: Creativity and artificial intelligence. Artificial Intelligence **103** (1998) 347–356
7. Pearce, M., Meredith, D., Wiggins, G.: Motivations and methodologies for automation of the compositional process. Musicae Scientiae **6** (2002) 119–147
8. Cope, D.: Virtual music: computer synthesis of musical style. MIT press (2004)
9. Fernández, J.D., Vico, F.: AI methods in algorithmic composition: A comprehensive survey. Journal of Artificial Intelligence Research **48** (2013) 513–582
10. Brabazon, A., O'Neill, M., McGarraghy, S.: Grammatical evolution. In: Natural Computing Algorithms. Springer (2015) 357–373
11. Miranda, E.: At the crossroads of evolutionary computation and music: Self-programming synthesizers, swarm orchestras and the origins of melody. Evolutionary Computation **12** (2004) 137–158
12. Burton, A.R., Vladimirova, T.: Generation of musical sequences with genetic techniques. Computer Music Journal **23** (1999) 59–73
13. Dahlstedt, P.: Thoughts on creative evolution: a meta-generative approach to composition. Contemporary Music Review **28** (2009) 43–55
14. Biles, J.: GenJam: A genetic algorithm for generating jazz solos. In: Proceedings of the International Computer Music Conference, International Computer Music Association (1994) 131–131
15. Moroni, A., Manzolli, J., Von Zuben, F., Gudwin, R.: Vox populi: An interactive evolutionary system for algorithmic music composition. Leonardo Music Journal **10** (2000) 49–54
16. Reddin, J., McDermott, J., O'Neill, M.: Elevated Pitch: Automated grammatical evolution of short compositions. In: Applications of Evolutionary Computing. Springer (2009) 579–584
17. Waschka II, R.: Composing with genetic algorithms: GenDash. In: Evolutionary Computer Music. Springer (2007) 117–136
18. Eigenfeldt, A., Pasquier, P.: Populations of populations: composing with multiple evolutionary algorithms. In: Evolutionary and Biologically Inspired Music, Sound, Art and Design. Springer (2012) 72–83
19. Loughran, R., McDermott, J., O'Neill, M.: Grammatical music composition with dissimilarity driven hill climbing. In: Evolutionary and Biologically Inspired Music, Sound, Art and Design. Springer (2016)
20. de Freitas, A.R., Guimaraes, F.G., Barbosa, R.V.: Ideas in automatic evaluation methods for melodies in algorithmic composition. In: Sound and Music Computing Conference. (2012)
21. Todd, P.M., Werner, G.M.: Frankensteinian methods for evolutionary music. Musical networks: parallel distributed perception and performace (1999) 313
22. Dahlstedt, P.: Autonomous evolution of complete piano pieces and performances. In: Proceedings of Music AL Workshop, Citeseer (2007)
23. Loughran, R., McDermott, J., O'Neill, M.: Tonality driven piano compositions with grammatical evolution. In: Evolutionary Computation (CEC), 2015 IEEE Congress on, IEEE (2015) 2168–2175
24. Munoz, E., Cadenas, J., Ong, Y.S., Acampora, G.: Memetic music composition. IEEE Transactions on Evolutionary Computation **20** (2016)
25. Lehman, J., Stanley, K.O.: Efficiently evolving programs through the search for novelty. In: Proceedings of the 12th annual conference on Genetic and evolutionary computation, ACM (2010) 837–844

26. Stanley, K.O., Lehman, J.: Why Greatness Cannot Be Planned: The Myth of the Objective. Springer (2015)
27. Charnley, J., Pease, A., Colton, S.: On the notion of framing in computational creativity. In: Proceedings of the Third International Conference on Computational Creativity. (2012) 77–81
28. Colton, S., Pease, A., Charnley, J.: Computational creativity theory: The face and idea descriptive models. In: Proceedings of the Second International Conference on Computational Creativity. (2011) 90–95
29. Jordanous, A.: Evaluating evaluation: Assessing progress in computational creativity research. (2011)
30. Colton, S.: Creativity versus the perception of creativity in computational systems. In: AAAI Spring Symposium: Creative Intelligent Systems. (2008) 14–20
31. McCormack, J.: Open problems in evolutionary music and art. In: Applications of Evolutionary Computing. Springer (2005) 428–436
32. Pearce, M., Wiggins, G.: 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, Citeseer (2001) 22–32
33. Pearce, M.T., Wiggins, G.A.: Evaluating cognitive models of musical composition. In: Proceedings of the 4th international joint workshop on computational creativity, Goldsmiths, University of London (2007) 73–80
34. Bringsjord, S., Bello, P., Ferrucci, D.: Creativity, the turing test, and the (better) lovelace test. In: The Turing Test. Springer (2003) 215–239
35. Loughran, R.B.: Musical Instrument Identification with Feature Selection Using Evolutionary Methods. PhD thesis, University of Limerick (2009)
36. Loughran, R., Walker, J., O'Neill, M., McDermott, J.: Genetic programming for musical sound analysis. In: Evolutionary and Biologically Inspired Music, Sound, Art and Design. Springer (2012) 176–186
37. Maher, M.L.: Evaluating creativity in humans, computers, and collectively intelligent systems. In: Proceedings of the 1st DESIRE Network Conference on Creativity and Innovation in Design, Desire Network (2010) 22–28
38. Cook, M., Colton, S.: Generating code for expressing simple preferences: Moving on from hardcoding and randomness. In: Proceedings of the Sixth International Conference on Computational Creativity June. (2015) 8
39. Miranda, E.R.: On the evolution of music in a society of self-taught digital creatures. Digital Creativity **14** (2003) 29–42
40. http://beyondthefencemusical.com: Beyond the fence (2016)
41. Boden, M.A.: Computer models of creativity. AI Magazine **30** (2009) 23
42. Hofstadter, D.R.: Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought. Basic books (2008)
43. Jefferson, G.: The mind of mechanical man. British Medical Journal **1** (1949) 1105
44. Bringsjord, S., Caporale, C., Noel, R.: Animals, zombanimals, and the total turing test. Journal of Logic, Language and Information **9** (2000) 397–418
45. Ellis, S., Haig, A., Bringsjord, S., Valerio, J., Braasch, J., Oliveros, P., et al.: Handle: Engineering artificial musical creativity at the "trickery" level. In: Computational Creativity Research: Towards Creative Machines. Springer (2015) 285–308