Hybrid Symbolic-Waveform Modeling of Music – Opportunities and Challenges

Jens Johannsmeier[1,*], Sebastian Stober[1]

[1] *Otto-von-Guericke University Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany*

Abstract

Generative modeling of music is a challenging task due to the hierarchical structure and long-term dependencies inherent to musical data. Many existing methods, especially powerful deep learning models, operate exclusively on one of two levels: The *symbolic* level, e.g. notes, or the *waveform* level, i.e. outputting raw audio without reference to any musical symbols. In this position paper, we argue that both approaches have fundamental issues which limit their potential for applications in computational creativity, particularly open-ended creative processes. Namely, symbolic models lack grounding in the reality of sound, whereas waveform models lack a level of abstraction beyond sound, such as notes. We argue that *hybrid* models, encompassing components at both levels, combine the best of both, while circumventing their respective disadvantages. While such models already exist, they generally consist of separate components that are combined only after training has finished. In contrast, we advocate for fully integrating both levels from the start. We discuss the opportunities afforded by this approach, as well as ensuing challenges, along with possible solutions. Our belief is that end-to-end hybrid modeling of musical data can substantially advance the quality of generative models as well as our understanding of musical creativity.

Keywords

computational creativity, generative models, music, hybrid models, position paper

1. Introduction

Music, besides images and text, is currently one of the most researched domains both in the field of (deep) generative modeling as well as the computational creativity community. It is also exceptionally challenging due to factors such as the enormous volume of data, hierarchical and repetitive structure, and extremely long-term dependencies. Successfully generating pieces of music with sensible structure spanning minutes has only recently become possible. A main driving factor for this success have been generative models using deep neural networks trained on vast amounts of data [1]. Existing approaches for modeling musical data often work on one of two levels [2]: At the *symbolic* level, we regard music as a sequence of symbols. Examples are sheet music, ABC notation, piano rolls or MIDI. These symbols can be turned into sound via pre-built digital instruments, or by a human performer. At the *audio*, or *waveform* level, we directly model the realization of music as sound, without reference to any symbols. We may also include spectrogram-based models in this definition. Both levels have respective advantages, disadvantages and limitations, which are to some extent complementary. While symbols are generally easier to handle due to their smaller vocabulary and high degree of abstraction, they are also limited with regard to what they can express. On the other hand, modeling waveforms is conceptually straightforward, but difficult in practice due to the high resolution and extreme repetitiveness inherent to oscillating audio waves.

Aside from these more technical aspects, we argue that neither approach is ideal for many applications in computational creativity. These often go beyond merely modeling a fixed dataset via straightforward reduction of some loss function, as is commonly done in deep generative modeling. One such task that we want to pay special attention to is that of *open-ended* learning and creation [3]. This requires models to continuously adapt within an ongoing, ever-changing process, necessarily moving beyond any fixed dataset eventually. In this paper, we present arguments for why neither purely symbolic nor waveform

International Workshop on Artificial Intelligence and Creativity, co-located with ECAI, October 19–24, 2024, Santiago de Compostela, Spain

[*] Corresponding author.

jens.johannsmeier@ovgu.de (J. Johannsmeier); stober@ovgu.de (S. Stober)
https://ai.ovgu.de/Staff/Johannsmeier.html (J. Johannsmeier); https://ai.ovgu.de/Staff/Stober.html (S. Stober)

models are appropriate for such open-ended tasks. We base our arguments not on properties of specific model architectures or symbolic systems, but rather on *fundamental* properties of either level of modeling.

Instead, we argue for a *hybrid* approach which encompasses aspects of both symbolic and waveform levels. Specifically, we envision a setup where the model generates symbolic data, which is then carried into the waveform level and evaluated there. Other than in waveform models, these two aspects are explicitly separated in the hybrid approach. At the same time, we do not completely lose access to the waveform level, as do purely symbolic models. We believe that this combines the advantages of both levels, while circumventing the fundamental limitations of either. While the idea of hybrid modeling is not new per se (e.g. [4]), we put emphasis on combining it in an *end-to-end* fashion with the usage of deep neural networks and gradient-based learning, as these have been shown to produce impressive results for generative modeling in recent years (e.g. [5]). Previous works using hybrid models generally do not train or evolve both components jointly, rather concatenating them after the fact. Such joint end-to-end training leads to unique challenges, to which we also discuss potential solutions. Our main goal with this work is to stimulate investigations into solving outstanding issues with hybrid modeling.

2. The Symbolic Level and its Shortcomings

As previously discussed, symbolic approaches model music as sequences of symbols with some fixed musical interpretation. To this date, most systems for so-called *metacreation* [6] in music operate at this level. Most symbolic systems are fundamentally discrete, which imposes a certain maximum granularity on the representations. In addition, many aspects of music, such as timbre, are not represented at all. Depending on the context, these limitations can become problematic. In particular, it can be difficult to model aspects of *performance*, such as variations in tempo or dynamics. Music lacking these factors may sound robotic and uninspired. It can also be argued that the dimension of sound is a crucial part of musical creativity itself [7].

On the other hand, a limited vocabulary generally simplifies modeling – if there are not that many choices, it is easier to make the correct one. Also, symbols generally represent sensible musical concepts. As such, the model only needs to choose between these concepts at any given time, making it less likely (or even impossible) to only produce unstructured noise, for example. Furthermore, the fact that we can simply use instruments (real or digital) to turn them into sound removes the need to model fine-grained harmonic oscillations, as is the case for waveform models. Another major advantage of common symbolic representations is that they are relatively compact in time: A handful of notes can represent several seconds of audio. This presents a reduction by orders of magnitude over modeling at the waveform level, where a few seconds of audio may already require tens of thousands of values. This, in turn, makes it simpler for models to represent the long-term dependencies we see in music.

And yet, we want to highlight a fundamental issue that all symbolic representations share: They are, themselves, meaningless, with any musical meaning imposed by human interpretation of the symbols. Before we expand on this, to understand why it is an issue we need to discuss the relation between *data-driven* modeling and computational creativity.

2.1. On Data-Driven Modeling

Arguably, the most (superficially) impressive computer-generated artworks are produced by contemporary generative models based on deep learning, such as Stable Diffusion [8] for images, ChatGPT [9] for text, or MusicGen [10] for music. These models are trained to approximate the underlying probability distributions of vast amounts of already existing data. While it is impossible to deny the massive improvements in this field in recent years, the computational creativity community tends to concern itself with different issues. For example, in the taxonomy of Ventura [11], even these impressive models lack important capacities such as judging and filtering their own results, referring to explicit knowledge bases and/or previous generations, as well as any sort of intentionality. Berns and Colton [12] put forward similar arguments specifically for deep generative models, noting they are “currently only good at producing more of the same”.[1]

[1] It should be noted that both cited papers were written before the recent wave of large generative models.

A fundamental limit of all these approaches is that they have a fixed target distribution that they attempt to reach, and once they have fulfilled this goal, they are essentially static. We believe that a more fruitful basis for computational creativity is the idea of open-endedness [13]. Here, we model the creative process as such, a process that is never finished, but rather continually evolving with reference to itself.

The challenge is to design this process in such a fashion that it does not evolve into meaningless chaos. A necessary condition to achieve this, we argue, is that the data have some degree of *inherent* meaning, a *grounding* in something real and tangible. To see this, consider asking someone to compose pieces of music in a symbolic system they are entirely unfamiliar with. They are informed about the vocabulary, but not what any of it means. Also, they are not allowed to listen to any renditions of their pieces. As such, they are unable to judge what any of their compositions sound like. Clearly, this will not work.[2] There are two solutions: First, show them many examples of compositions in that system, so they may learn from and copy (aspects of) them. This is what straightforward generative modeling does. Second, inform them about the meaning of the symbols, so they may use them skillfully and with intention. In a similar vein, they could be allowed to listen to their compositions, and thus improve by trial-and-error. As we want to avoid (or at least go beyond) option 1, we necessarily end up with option 2 – grounding the symbols with meaning, by connecting them with what they *actually sound like*.

A full treatment of open-endedness is beyond the scope of this work. We discuss some approaches in Section 5. For now, we simply observe that symbols require grounding, and this grounding to be made available to the models in some fashion, to achieve meaningful (or even sensible) open-ended creative output.

2.2. Back to the Symbolic Level

In light of these circumstances, we argue that purely symbolic data does not meet this grounding condition. Let us take western sheet music and twelve-tone equal temperament as an example. Within this context, certain tonal intervals are viewed as *consonant* (low tension) or *dissonant* (high tension). For example, the interval of a fifth (seven semitones) is often judged as a consonant interval. However, this is obviously *not* due to the fact that the notes are seven semitones apart, but rather that the *base frequencies* of the two tones are in a certain relation (approximately 3:2 in this case) which people generally find pleasant [14, 15]. Accordingly, if one where to play in a *microtonal* system with more than twelve tones per octave, this correspondence would no longer hold, and seven semitones may or may not represent a consonant interval. This shows that rules on the symbolic level cannot stand on their own; they are rather shaped by the actual tones that the symbols represent.

So how come that contemporary symbolic models can create pleasant music? This is because they essentially learn to mimic existing data, i.e. compositions by humans, which have been created with sound in mind. A human may use the interval of a fifth because they want to achieve a certain sound. A symbolic model will use the interval of a fifth because it has a high probability given what it has learned from the training data. Thus, a symbolic model *cannot* sensibly create new symbolic rules, since it does not have access to the meaning of the symbols. It follows that such models are inappropriate for data-independent learning tasks. That this is actually an issue can be seen, for instance, in existing approaches using genetic algorithms to evolve music. These tend to hard-code human assumptions into their fitness function, for example rewarding consonant intervals and punishing undesirable melodic sequences [16, 17]. Alternatively, human-in-the-loop methods employ a human "fitness function" to evaluate outputs [18, 19, 20]. While an important field of study in its own right, this is slow and cumbersome to scale with regards to automatic music generation.

A more flexible approach is used by Mitrano et al. [21]: They use pre-trained Recurrent Neural Networks (RNNs) to replace human critics in fitness functions. The networks are trained on symbolic datasets of human-composed music. In a similar vein, Ostermann et al. [22] use the discriminator of a Generative Adversarial Network (GAN) to differentiate between real and artificial compositions. Since neural networks can express highly complex functions, the fitness functions they represent can be similarly complex.

[2] We disregard the possibility that a person may be able to *guess* the meaning of symbols, as this clearly cannot be done by a straightforward generative computer system without any world knowledge.

Despite these advances, by formulating purely symbolic rules, none of these approaches tackle the fundamental issue: In all these cases, the musical meaning is imposed from the outside, as there is no way by which it could arise from itself. We want to emphasize that we do not mean to imply that using human preconceptions, or training on a fixed dataset, is somehow less desirable than open-ended processes. Also, the problem of grounding goes far beyond the use of symbolic or waveform data; see e.g. [23, 24] for more detailed discussions.

3. The Waveform Level and its Shortcomings

Many state-of-the-art music generation models approach the task directly at the waveform level. Due to the high temporal resolution of high-quality audio, this requires very large models capable of modeling extremely long-term dependencies. The main advantage here is that such models can produce any arbitrary audio waves in principle, without relying on limited symbolic systems and digital instruments. In particular, even complex polyphonic pieces of music can be modeled as a single stream of samples.

And yet, humans would likely be completely overwhelmed if asked to create a musical piece sample by sample. Rather, we *do* operate at a more abstract, quasi-symbolic level, where we manipulate pre-built instruments via a set of gestures. While this obviously imposes limitations, it also makes "making music" affordable in the first place. For example, when playing the piano, we do not need to worry about being off-tune, or to even be tonal at all; we simply hit a key to produce a fixed pitch. When playing the violin, there is no need to model the correct timbre along with a possibility to accidentally produce, say, a flute sound, or an undesirable constant buzz in the background. Thus, we argue that waveform models have two main disadvantages:

1. They spend enormous capacity on learning what audio is in the first place (oscillations) and how to produce the correct kind of audio (specific instruments). Further capacity is spent on having to model each sample individually.
2. The lack of well-defined symbolic controls makes them hardly useful for non-data-driven computational creativity tasks. This is because emergent behavior and new rules should be much simpler to produce in the restricted lower-dimensional symbolic space.

4. Hybrid Approach

As we have seen, neither purely symbolic nor waveform modeling seems satisfactory for open-ended creativity. We argue that a *hybrid* modeling approach can provide all the components necessary for creative music systems. We envision a setup that includes aspects of both symbolic and waveform levels: In a first step, musical symbols are created; these symbols have a specific meaning and are thus inherently interpretable. Then, these symbols are sonified through dedicated instruments, which may be fixed or themselves part of the model. Crucially, the *evaluation* of the produced musical sequences happens at the waveform level. This guarantees that the symbolic sequences are grounded through the sound they represent. With the hybrid setup, we have removed the main disadvantages of both levels: Symbols are no longer meaningless, as they are connected to the audio level within the model itself. On the other hand, the model no longer needs to produce musical audio sample by sample, instead being able to rely on symbolic abstractions and instruments.

4.1. Revisiting the Grounding Problem

We have argued that a symbolic approach is not suitable for data-independent learning, as the symbols are not grounded. We claimed that this grounding can be achieved by linking symbols to sound within the model itself. But without any data, how would *the audio* be grounded? If there is no reference, there is no reason for a model to prefer musical audio to, say, random noise, or some more structured but non-musical patterns. Thus, we still require some kind of grounding even at the waveform level. We argue that, if one wants to investigate the process of creativity from the ground up, this should *not* take the form of musical knowledge, e.g. hardcoding a preference for certain pitches, or even harmonic oscillations at all. Rather, a more fundamental grounding may be sufficient. Heath and Ventura [25] argue that "before a computer can draw, it must first learn to see". We believe a similar statement can be made for music and hearing.

As such, our hypothesis is that training a deep neural network as a *general* feature extractor for (not necessarily musical) audio can provide enough grounding for open-ended generation to be feasible. Such a model should learn to efficiently encode perceptually relevant features, such as harmonic oscillations, if these are found in the data [26]. This could provide sufficient grounding without forcing any "musical ideas" onto the model, which in turn may allow for investigation of computational creativity "from the ground up". We do not see any comparable opportunity for models operating purely at the symbolic level without any grounding.

Perhaps a similar direction is afforded by the idea of using *hallucinations* as creative outputs [27]. Methods such as Deep Dream [28] certainly produce aesthetically pleasing outputs, and the same framework can be leveraged for audio, as well. Wyse [29] similarly puts emphasis on creative behaviors emerging from networks that are purely intended for perception (i.e. classifiers). Now, one might counter that such a model is unlikely to hallucinate music specifically, rather than arbitrary sounds. However, if we limit the expressiveness of the model to symbolic control of pre-made instruments, we believe we have created the necessary preconditions for expression that is musical, and yet less constrained by human biases.

Finally, we need to emphasize that this is conjecture. We intend to put these hypotheses to the test in future work, and we hope to convince other researchers that this is a worthwhile direction of inquiry. In particular, general computational theories for the rise of creativity in artificial systems have been proposed in the past [30, 31, 32]. We believe it would be exciting to apply such models of creative evolution within the context of hybrid symbolic-waveform models for music.

4.2. Challenges

Jointly modeling symbolic and waveform levels comes with its own issues. Chief among them may be gradient propagation. Most state-of-the-art generative audio models are trained using gradient descent to minimize a specific loss function [33]. Given how successful this approach has been, it makes sense to want to adopt it for training hybrid models.

Gradient descent requires the entire model, as well as the loss, to be differentiable all the way through. However, many operations at the symbolic level are fundamentally discrete. For example, a specific piano key is either pressed, or not. Usually, only a few keys, at most, are pressed down at any given time. Symbolic generative models, however, most often return a soft probability distribution over keys (or symbols, more generally), which can then be sampled from to choose a key. This sampling operation is not differentiable. This is fine for a purely symbolic model, as during training, we can directly compare the soft outputs to the targets, so no sampling is necessary. Conversely, when using the trained model to generate pieces of music, sampling is fine as no gradients are required.

However, when training a hybrid model, we most likely do *not* want to keep the soft distribution, as this corresponds to pressing *all* keys at the same time, at varying strengths (which may be likened to MIDI velocity). Thus, we arrive at the main dilemma of the hybrid approach: To sensibly transform symbols into audio, we require discrete operations, but these do not allow for backpropagation of gradients, seemingly preventing use of the most successful learning method of our time, gradient descent.

There are several ways to tackle this issue. First off, the Gumbel-Softmax, or concrete distribution, [34, 35] uses the *reparameterization trick* to draw samples from a soft distribution in a differentiable manner. Furthermore, these samples can be smoothly interpolated towards being approximately discrete using a hyperparameter. Usually, this is changed during the learning procedure, such that samples are soft initially, and become progressively closer to the discrete ideal over time. After training is finished, one can simply draw discrete samples instead. While powerful and simple, this approach has limits; for some actions it simply does not make sense to have a soft choice. Also, we are strictly speaking still training on soft distributions, incurring a gap between training and deployment of the model.

Aside from that, there are methods that do not require gradients. Genetic or evolutionary algorithms are such approaches. These have been used extensively in computational creativity research. However, such methods are most often used for relatively constrained models with small search spaces. In contrast, random mutations are unlikely to succeed in the context of deep neural networks. This is due to their very large number of parameters; the bigger the search space, the less likely it is that unprincipled random changes to the weights will lead to improvements. As such, a purely genetic approach seems feasible only for very small networks. One example would be the NEAT algorithm [36] applied to Compositional Pattern Producing Networks (CPPNs) [37]. These networks have been shown to be able to produce complex structures with only a handful of neurons. It is also possible to evolve network structures and then further train the weights [38].

Alternatively, methods from reinforcement learning could be employed. In particular, policy gradients [39] are used in deep reinforcement learning to compute neural network gradients despite making discrete choices among a set of possible actions. This is done by rewriting the gradients such that they can be approximated by sampling actions. The downside of this method is that the sampling can give imprecise approximations of the true expected gradient. To offset this, we would need to sample many actions independently for each gradient step. Combined with the high sampling rate common with high-quality audio, this can quickly become unmanageable in terms of computational resource requirements.

Finally, there is already a kind of hybrid symbolic model in use in state-of-the-art generative models, namely the Vector-Quantized Variational Autoencoder (VQ-VAE) [40]. When applied to audio, this will superficially act like a regular autoencoder, outputting audio directly, with no reference to the symbolic level. However, the latent space is *discretized* through the method of vector quantization. Strictly speaking, this limits the autoencoder to producing realizations of these discrete vectors, which could be classified as symbols. However, these symbols are *not* pre-determined, like an instrument with specific controls, for example. Rather, the quantization is learned along with the autoencoder. Furthermore, the vectors need not map to any interpretable or discernible concepts. Rather, they tend to simply perform a kind of clustering of the latent space. Next, the codes are usually processed through many layers of convolutions, which intermingles the effects of the different codes and makes it difficult do ascribe a specific meaning to any one of them. Still, given the issues outlined with other approaches, this could be a promising starting point if the discrete codes could somehow be forced towards a more meaningful representation. In general, VAE latent spaces have been judged as appropriate to learn conceptual spaces for musical data [41].

As we can see, there are a number of established methods that may be leveraged for hybrid models. Future work needs to compare how the different methods work in practice. After a well-performing method has been found, hybrid models can then be evaluated in open-ended creative learning tasks.

5. Related Work

In this section, we will first review exemplary state-of-the-art models at the symbolic and waveform levels, before relating our arguments to more integrative works from the computational creativity community.

There is a long history of modeling at the level of musical symbols. Some early examples are Mozer [42] or Eck and Schmidhuber [43]. Modern approaches tend to use powerful Transformer [44] models, trained on large datasets, to generate symbolic sequences autoregressively. Examples include Music Transformer [45] and MuseNet [46]. This approach is conceptually very simple, but is the same idea that has been shown to be extremely effective in the recent surge of Large Language Models [47].

Other works have incorporated *hierarchical structures* into the models [48, 49, 50]. These are clearly present in music [51]. For example, modern popular music generally consists of a sequence of structures such as verse, chorus, bridge, etc., which in turn consist of repeating motives, and so on. Still, it seems more common to ignore any such inductive biases and simply model musical data as a flat sequence.

For raw audio, early successful work in modeling at scale relied on autoregressive models to generate samples one by one. A key breakthrough here was WaveNet [52]. While originally developed for text-to-speech applications, it was also shown to work for generating music at the waveform level. The main disadvantage of WaveNet is that it is extremely slow due to the sample-by-sample generation, taking minutes to generate just one second of audio, although this was improved by future work [53, 54].

A different approach is taken by the Differentiable Digital Signal Processing (DDSP) model [55]: Here, audio is modeled via a fundamental frequency and a distribution of overtones of that frequency, which is turned into waveforms via sinusoidal oscillators. Alternatively, more flexible learnable wavetable synthesizers can be used. Non-tonal audio can be created via filtered noise. Other effects, such as reverb, can be added in a differentiable fashion. These choices mean that the model does not rely on autoregressive generation of samples, and does not need to learn to generate oscillations in the first place. There is, however, a loss of flexibility: In particular, polyphonic audio is difficult to model with only a single F0 contour.

Recent models use yet another method: JukeBox [56] was the first model to tackle generation of diverse multi-genre, polyphonic music directly at the waveform level, including various conditioning options. The model uses a hierarchical VQ-VAE to downsample the high-resolution audio data; an autoregressive Transformer model is trained to generate sequences of codes which are then decoded back to audio space. This is somewhat close to our proposed approach, but as we mentioned, the learned code "symbols" are rarely interpretable. Furthermore, the codes still have a relatively high sampling rate (on the order of 100s of Hz). Beyond JukeBox, the recent generation of text-to-music models, such as MusicLM [5] or MusicGen [10], use similar approaches, although the hierarchical VQ-VAE is replaced by a single-level residual one.

As for hybrid models, we are not aware of previous works that employ these for open-ended music generation. As such, we will review works that have used hybrid models in some capacity. Manzelli et al. [4] condition a Wavenet model on symbolic (MIDI) sequences generated by a *biaxial LSTM* [57]. They raise many of the same concerns we have with either purely symbolic or waveform models. However, in their setup the symbolic model is trained first, and its outputs are used as-is for the waveform model. There is no real interaction between the levels, with the symbolic model receiving no further learning signal from the waveform level. This once again removes the possibility of "learning to play the instrument" from audio feedback.

Wu et al. [58] propose MIDI-DDSP, an extension of the original DDSP architecture. This is a three-level model that maps notes to expression parameters, and expression parameters to synthesis parameters, which are in turn used by a DDSP model to produce audio. This allows for intuitive control of the DDSP model by manipulating notes and/or how those notes are expressed. However, the three components of this model are trained separately, so MIDI-DDSP is not an end-to-end hybrid model in the sense we envision.

Prang and Esling [59] propose mapping symbolic data to a signal-like space to improve embeddings learned with a Variational Autoencoder [60]. They show that their representation results in better reconstructions than symbolic data, as well as more musically sensible structure in the latent space. This presents yet more evidence that disregarding the waveform level may be harmful to any model working on musical data, including creative output and generation.

Finally, we can also look outside the musical domain. Colton et al. [61] propose *Stable Evolusion*, where text prompts to a Stable Diffusion model are evolved, instead of evolving images directly. This is somewhat analogous to the proposed symbolic-waveform hybrids: Instead of searching the high-dimensional, semantically unstructured image space, the search is relegated to the more meaningful textual level.

Regarding open-endedness, a detailed discussion of *why* this is itself a desirable goal is beyond the scope of this paper. As such, we will only provide pointers to other works which have previously made the case for this direction of research. Lehman and Stanley [3] propose *novelty search* as a method to solve complex machine learning problems (see also [62]). Here, the "fitness function" rewards new emerging behaviors, rather than progress on an objective function. Complex patterns emerge automatically, as there are simply not many options to do something novel with simple behaviors. We believe such a paradigm to be particularly well-suited for creative systems. Later work proposes *minimal criterion coevolution* [63]; here, two populations evolve together with minimal evolutionary pressure. Any organism that is capable of solving some minimal task is allowed to evolve. Once again, we believe this to fit particularly well with a two-sided musician-listener model, and in fact similar setups have been applied to music in the past [64, 65]. For a more philosophical discussion focused on applications in music, see [66]. They argue against seeing human evaluation, or objective evaluation against human standard (e.g. by computing loss metrics comparing to human-composed music), as the ideal way of evaluating artificial creative systems. Guckelsberger et al. [67] offer a different perspective, suggesting a framework where agents strive to increase their (and others) *empowerment*. The authors put emphasis on multi-agent settings, where agents may also strive for minimizing or maximizing the empowerment of other individuals. As such, their *coupled empowerment maximization* also includes the social dimension, which is evidently important to many creative activities.

Few works have tackled open-ended generation with large neural networks. Elgammal et al. [68] propose Creative Adversarial Networks (CANs), a framework similar to Generative Adversarial Networks. The generator is trained to fool a discriminator into classifying its outputs as real art, as with standard GANs. However, a second loss term encourages the generator to produce outputs that cannot be classified into any style known by the discriminator, essentially creating new styles that still conform to the overall art distribution. Chemla-Romeu-Santos and Esling [69] propose a framework of *divergence maximization*, where a pre-trained generative model is purposefully made to extrapolate beyond its learned distribution in order to generate novel outputs. The authors also implemented prototypes for simple image data [70]. Here, a VAE is first trained as normal; afterwards it is trained to produce outputs *diverging* from the given class distributions, while regularized to still remain in the overall data distribution. Similar to CANs, this objective is best fulfilled by new types of outputs that do not conform to any known class, yet are still similar to the given data overall.

As we can see, there are powerful models available both for symbolic as well as waveform generation. Furthermore, the computational creativity community has recently provided several examples of hybrid models, incorporating both levels of modeling, producing favorable results. Finally, open-ended generation and the search for novelty has been a long-running topic both in computational creativity and in areas such as artificial life. The ingredients are there; it is time to start cooking.

6. Conclusion

In this paper, we have presented arguments for the limitations of modeling music purely at either the symbolic or waveform level. A purely symbolic representation lacks grounding, limiting its usefulness especially with regards to open-ended creativity beyond human-imposed rules and preferences. Modeling waveforms directly, on the other hand, makes the task significantly more complex due to the high sampling rate and the need to learn concepts such as harmonic oscillations from scratch. It also does not provide a good fit to human musical activity, which generally evolves around manipulating instruments in a (quasi-)symbolic fashion. Instead, we make a case for *hybrid* models that work at the symbolic level, but additionally perform the transformation to the waveform level, including a learning signal at that level, as well. This combines the best aspects of both levels into one, and is particularly well-suited for open-ended learning tasks with much creative potential. Systems built in such a way could learn to play instruments based on the sound they actually create, rather than purely symbolic notions. They may even modify existing instruments, or create entirely new ones to suit their "needs", given the capacity.

We also presented challenges with this approach, however. Chief among these is the transition from symbols to waveforms, as this needs to be carried out in a fashion that allows learning signals to pass through. This is particularly troublesome for gradient-based learning approaches. We believe that such approaches, used in powerful deep neural networks, are the method of choice for difficult modeling tasks due to their unparalleled potential to model very complex data distributions with long-term dependencies. As we have presented, though, several methods to tackle this issue are already available. A proper investigation of these options is needed next.

For a long time, symbolic modeling of music has been the dominant paradigm in the computational creativity community. However, we have seen that a variety of authors have recently made the case for including information from the raw waveforms, as well. Such models require significantly more computation, and this is true for hybrid models, too: While we may not need to learn *how to produce* proper musical audio in this framework, we still have to *represent* and work with this high-dimensional data. However, with the recent advances in computing hardware, and the software necessary to use it [71, 72], this has become far more achievable.

As such, we hope to inspire further research into technical solutions to the unique challenges hybrid modeling of music brings with it. We believe that such an approach could power new insights into open-ended creative processes for generating music, bring such systems closer to creative autonomy, and teach us about the roots of our own creativity, as well.

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (2015) 436–444.

[2] G. Wiggins, E. Miranda, A. Smaill, M. Harris, A framework for the evaluation of music representation systems, Computer Music Journal 17 (1993) 31–42.

[3] J. Lehman, K. O. Stanley, Exploiting open-endedness to solve problems through the search for novelty., in: ALIFE, 2008, pp. 329–336.

[4] R. Manzelli, V. Thakkar, A. Siahkamari, B. Kulis, An end to end model for automatic music generation: Combining deep raw and symbolic audio networks, in: Musical metacreation workshop, 2018.

[5] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, et al., Musiclm: Generating music from text, arXiv preprint arXiv:2301.11325 (2023).

[6] P. M. Bodily, D. Ventura, Musical metacreation: past, present, and future, in: Proceedings of the sixth international workshop on musical metacreation, 2018.

[7] M. Windsor, Using raw audio neural network systems to define musical creativity, in: AIMC, 2022. URL: https://doi.org/10.5281/zenodo.7088438. doi:10.5281/zenodo.7088438.

[8] StabilityAI, Stable diffusion, https://stability.ai/stable-image (2022). URL: https://stability.ai/stable-image.

[9] OpenAI, Chatgpt, https://chat.openai.com/ (2022). URL: https://chat.openai.com/.

[10] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, A. Défossez, Simple and controllable music generation, arXiv preprint arXiv:2306.05284 (2023).

[11] D. Ventura, Mere generation: Essential barometer or dated concept, in: ICCC, 2016, pp. 17–24.

[12] S. Berns, S. Colton, Bridging generative deep learning and computational creativity., in: ICCC, 2020, pp. 406–409.

[13] K. O. Stanley, Why open-endedness matters, Artificial life 25 (2019) 232–235.

[14] E. G. Schellenberg, S. E. Trehub, Natural musical intervals: Evidence from infant listeners, Psychological science 7 (1996) 272–277.

[15] C. L. Krumhansl, The cognition of tonality–as we know it today, Journal of New Music Research 33 (2004) 253–268.

[16] R. A. McIntyre, Bach in a box: The evolution of four part baroque harmony using the genetic algorithm, in: Proceedings of the first ieee conference on evolutionary computation. ieee world congress on computational intelligence, IEEE, 1994, pp. 852–857.

[17] Z. Ren, Style composition with an evolutionary algorithm, in: AIMC, 2020.

[18] A. Jordanous, A fitness function for creativity in jazz improvisation and beyond, in: ICCC, 2010.

[19] Y. Zhou, Y. Koyama, M. Goto, T. Igarashi, Generative melody composition with human-in-the-loop bayesian optimization, arXiv preprint arXiv:2010.03190 (2020).

[20] S. Dasari, J. Freeman, Directed evolution in live coding music performance, 2020.

[21] P. Mitrano, A. Lockman, J. Honicker, S. Barton, Using recurrent neural networks to judge fitness in musical genetic algorithms, in: Musical metacreation workshop, 2017.

[22] F. Ostermann, I. Vatolkin, G. Rudolph, Artificial Music Producer: Filtering Music Compositions by Artificial Taste, in: AIMC, 2022. URL: https://doi.org/10.5281/zenodo.7088395. doi:10.5281/zenodo.7088395.

[23] M. M. Al-Rifaie, M. Bishop, Weak and strong computational creativity, in: Computational creativity research: Towards creative machines, Springer, 2014, pp. 37–49.

[24] C. Guckelsberger, C. Salge, S. Colton, Addressing the "why?" in computational creativity: A non-anthropocentric, minimal model of intentional creative agency, in: ICCC, 2017.

[25] D. Heath, D. Ventura, Before a computer can draw, it must first learn to see, in: ICCC, 2016, pp. 172–179.

[26] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE transactions on pattern analysis and machine intelligence 35 (2013) 1798–1828.

[27] L. Berov, K.-U. Kühnberger, Visual hallucination for computational creation, in: ICCC, 2016, pp. 107–114.

[28] A. Mordvintsev, C. Olah, M. Tyka, Inceptionism: Going deeper into neural networks (2015). URL:

TODOBLOG.

[29] L. Wyse, Mechanisms of artistic creativity in deep learning neural networks, in: ICCC, 2019.

[30] R. Saunders, Artificial creative systems and the evolution of language, in: ICCC, 2011, pp. 36–41.

[31] L. Gabora, S. DiPaola, How did humans become so creative? a computational approach, in: ICCC, 2012.

[32] O. Bown, A model of runaway evolution of creative domains., in: ICCC, 2014, pp. 247–253.

[33] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, nature 323 (1986) 533–536.

[34] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, arXiv preprint arXiv:1611.01144 (2016).

[35] C. J. Maddison, A. Mnih, Y. W. Teh, The concrete distribution: A continuous relaxation of discrete random variables, arXiv preprint arXiv:1611.00712 (2016).

[36] K. O. Stanley, R. Miikkulainen, Evolving neural networks through augmenting topologies, Evolutionary computation 10 (2002) 99–127.

[37] K. O. Stanley, Exploiting regularity without development., in: AAAI Fall Symposium: Developmental Systems, 2006, p. 49.

[38] C. Fernando, D. Banarse, M. Reynolds, F. Besse, D. Pfau, M. Jaderberg, M. Lanctot, D. Wierstra, Convolution by evolution: Differentiable pattern producing networks, in: Proceedings of the Genetic and Evolutionary Computation Conference 2016, 2016, pp. 109–116.

[39] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine learning 8 (1992) 229–256.

[40] A. v. d. Oord, O. Vinyals, K. Kavukcuoglu, Neural discrete representation learning, in: NeurIPS, 2017, pp. 6309–6318.

[41] M. Peeperkorn, R. Saunders, O. Bown, A. Jordanous, Mechanising conceptual spaces using variational autoencoders, in: ICCC, 2022.

[42] M. C. Mozer, Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing, Connection Science 6 (1994) 247–280.

[43] D. Eck, J. Schmidhuber, Learning the long-term structure of the blues, in: International Conference on Artificial Neural Networks, Springer, 2002, pp. 284–289.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[45] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, D. Eck, Music transformer: Generating music with long-term structure, in: ICLR, 2018.

[46] OpenAI, Musenet, https://openai.com/research/musenet (2019). URL: https://openai.com/research/musenet.

[47] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, D. Amodei, Scaling laws for neural language models, arXiv preprint arXiv:2001.08361 (2020).

[48] B. Smith, G. Garnett, Improvising musical structure with hierarchical neural nets, in: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, volume 8, 2012, pp. 63–67.

[49] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, D. Eck, A hierarchical latent vector model for learning long-term structure in music, in: ICML, 2018, pp. 4364–4373.

[50] G. Zixun, D. Makris, D. Herremans, Hierarchical recurrent neural networks for conditional melody generation with long-term structure, in: 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8.

[51] F. Lerdahl, R. Jackendoff, An overview of hierarchical structure in music, Music Perception (1983) 229–252.

[52] A. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, Wavenet: A generative model for raw audio, arXiv preprint arXiv:1609.03499 (2016).

[53] A. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Driessche, E. Lockhart, L. Cobo, F. Stimberg, et al., Parallel wavenet: Fast high-fidelity speech synthesis, in: ICML, 2018,

pp. 3918–3926.

[54] L. Hantrakul, J. H. Engel, A. Roberts, C. Gu, Fast and flexible neural audio synthesis., in: ISMIR, 2019, pp. 524–530.

[55] J. Engel, C. Gu, A. Roberts, et al., Ddsp: Differentiable digital signal processing, in: ICLR, 2019.

[56] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, I. Sutskever, Jukebox: A generative model for music, arXiv preprint arXiv:2005.00341 (2020).

[57] D. D. Johnson, Generating polyphonic music using tied parallel networks, in: International conference on evolutionary and biologically inspired music and art, Springer, 2017, pp. 128–143.

[58] Y. Wu, E. Manilow, Y. Deng, R. Swavely, K. Kastner, T. Cooijmans, A. Courville, C.-Z. A. Huang, J. Engel, Midi-ddsp: Detailed control of musical performance via hierarchical modeling, in: ICLR, 2021.

[59] M. Prang, P. Esling, Signal-domain representation of symbolic music for learning embedding spaces, in: AIMC, 2020. URL: https://doi.org/10.5281/zenodo.4285386. doi:10.5281/zenodo.4285386.

[60] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[61] S. Colton, A. Smith, B. Pérez Ferrer, S. Berns, Artist discovery with stable evolusion, in: ICCC, 2023.

[62] J. Schmidhuber, Art & science as by-products of the search for novel patterns, or data compressible in unknown yet learnable ways, Multiple ways to design research. Research cases that reshape the design discipline, Swiss Design Network-Et al. Edizioni (2009) 98–112.

[63] J. C. Brant, K. O. Stanley, Minimal criterion coevolution: a new approach to open-ended search, in: GECCO, 2017, pp. 67–74.

[64] P. M. Todd, G. M. Werner, Frankensteinian methods for evolutionary music composition, Musical networks: Parallel distributed perception and performance 3 (1999) 7.

[65] B. Andrus, N. Fulda, A data-driven architecture for social behavior in creator networks, in: ICCC, 2022, pp. 339–348.

[66] R. Loughran, M. O'Neill, Generative music evaluation: why do we limit to 'human', in: Proceedings of the first Conference on Computer Simulation of Musical Creativity (CSMC), 2016.

[67] C. Guckelsberger, C. Salge, R. Saunders, S. Colton, Supportive and antagonistic behaviour in distributed computational creativity via coupled empowerment maximisation, in: ICCC, 2016, pp. 9–16.

[68] A. Elgammal, B. Liu, M. Elhoseiny, M. Mazzone, Can: Creative adversarial networks, generating" art" by learning about styles and deviating from style norms, arXiv preprint arXiv:1706.07068 (2017).

[69] A. Chemla–Romeu-Santos, P. Esling, Challenges in creative generative models for music: a divergence maximization perspective, in: AIMC, 2022. URL: https://doi.org/10.5281/zenodo.7088272. doi:10.5281/zenodo.7088272.

[70] A. Chemla–Romeu-Santos, P. Esling, Creative divergent synthesis with generative models, arXiv preprint arXiv:2211.08861 (2022).

[71] M. Abadi, Tensorflow: learning functions at scale, in: Proceedings of the 21st ACM SIGPLAN international conference on functional programming, 2016, pp. 1–1.

[72] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).