# Exploring Variational Auto-encoder Architectures, Configurations, and Datasets for Generative Music Explainable AI

Nick Bryan-Kinns[1]      Bingyuan Zhang[1]      Songyan Zhao[2]      Berker Banar[1]

[1]School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK

[2]Computer Science Department, Carleton College, Northfield MN 55057, USA

**Abstract:**   Generative AI models for music and the arts in general are increasingly complex and hard to understand. The field of explainable AI (XAI) seeks to make complex and opaque AI models such as neural networks more understandable to people. One approach to making generative AI models more understandable is to impose a small number of semantically meaningful attributes on generative AI models. This paper contributes a systematic examination of the impact that different combinations of variational auto-encoder models (measureVAE and adversarialVAE), configurations of latent space in the AI model (from 4 to 256 latent dimensions), and training datasets (Irish folk, Turkish folk, classical, and pop) have on music generation performance when 2 or 4 meaningful musical attributes are imposed on the generative model. To date, there have been no systematic comparisons of such models at this level of combinatorial detail. Our findings show that measureVAE has better reconstruction performance than adversarialVAE which has better musical attribute independence. Results demonstrate that measureVAE was able to generate music across music genres with interpretable musical dimensions of control, and performs best with low complexity music such as pop and rock. We recommend that a 32 or 64 latent dimensional space is optimal for 4 regularised dimensions when using measureVAE to generate music across genres. Our results are the first detailed comparisons of configurations of state-of-the-art generative AI models for music and can be used to help select and configure AI models, musical features, and datasets for more understandable generation of music.

**Keywords:**   Variational auto-encoder, explainable AI (XAI), generative music, musical features, datasets.

## 1 Introduction

Music generation is a key use of AI for arts, and is arguably one of the earliest forms of AI art. However, contemporary generative music models rely increasingly on complex machine learning models[1–3] such as neural networks[4, 5] and deep learning techniques[6–9] which are difficult for people to understand and control. This makes it hard to use such models in real-world music making contexts as they are generally inaccessible to musicians or anyone besides their creator.

Making AI models more understandable to users is the focus of the rapidly expanding research field of explainable AI (XAI)[10]. One approach to making machine learning models more understandable is to expose elements of the models to people in semantically meaningful ways, for example, using latent space regularisation[11] to increase the meaningfulness of dimensions in otherwise opaque latent spaces. To date, there has been very little research on the applicability and use of XAI for the arts. Indeed, there is a lack of research on what configurations of generative AI models and datasets are more or less, amenable to explanation.

This paper takes a first step towards understanding the link between the explanation and performance of AI models for the arts by examining what effect different AI model architectures, configurations and training datasets have on the performance of generative AI models that have some explainable features.

## 2 Related work

The field of explainable AI (XAI)[10] explores how complex and difficult to understand AI models such as neural networks can be made more understandable to people. Approaches to increasing the explainability of AI models include generating understandable explanations of

AI model behaviour, structuring and labelling complex AI models to make them more understandable, and approximating the behaviour of complex models with less complex and more understandable models (ibid.). An important element of XAI is the interpretability of an AI model which we take to mean the "ability to explain or to provide the meaning in understandable terms to a human"[12]. Unfortunately the concept of explanation is ambiguous and variously defined[13]. In machine learning (ML) literature, XAI often refers to making the reasons behind ML decisions more comprehensible to humans. For example, the majority of XAI research has examined how to explain the decisions of ML classification and predictor models – see [12] for a thorough survey. There also exists a broader view of the concept of explainability in which "explainability encompasses everything that makes ML models transparent and understandable, also including information about the data, performance, etc."[14] which we follow. In this paper, we are specifically concerned with how to make AI models more interpretable for people so that they can better control the generative aspects of the AI model. Approaches for explaining AI models are most often tied to specific AI models and data types. There is an emerging set of approaches which are not tied to specific AI models or data types, referred to as agnostic explanators[12]. However, agnostic approaches, such as LIME[15], are concerned with building an explanation model to explain the classification/prediction of an AI model whereas in this paper we focus on making the content of an AI model itself more interpretable so that the model can be better controlled for music generation.

To date, most XAI research has been concerned with goal-directed domains where task efficiency and transparency are key factors, for example, generating explanations for why an AI model made a medical diagnosis[16] or how the AI models in self-driving cars work[17, 18]. However, there has been little research on how XAI could be used in more creative domains such as the arts[19]. This lack of explainability typically limits the use of AI models for the arts to the creator of the AI model and severely limits their use by artists and practitioners. Of the limited research on XAI for the arts, Vigliensoni et al.[20] explore the presentation of visual cues between mappings in the latent space of an AI model, and McCormack et al.[21] research the visualisation of levels of mutual trust between an AI system and musicians in music making. This leaves many open research questions on the use of XAI for the arts ranging from questions about the explainability of different AI models and datasets to how to design user interfaces to navigate and manipulate explanations of generative AI models.

Taking music as a key form of artistic endeavour, this paper explores explainable AI for music. Musical problems addressed by AI models include composition, interpretation, improvisation and accompaniment[22]. In this paper we focus on a core use of AI for music – music composition or generation, otherwise known as generative music. Music itself has a multi-level structure that "ranges from timbre and sound through notes, chords, rhythmic patterns, harmonic patterns (e.g., cadences), melodic motifs, themes, sections, etc."[23]. As such, generative AI models range in purpose from generating monophonic sequences of notes (referred to as a melody), to polyphonic melodies, multivoice polyphony, accompaniment to a melody (counterpoint or chord progression), and association of a melody with a chord progression[24]. However, sequencing longer term structures such as themes and sections by integrating short-term and long-term machine learning for music generation remains an open research challenge[23] which is problematic, given that the "long-term and/or hierarchical structure of the music plays an important role"[2]. Applications of generative AI range from polyphonic classical music generation in the style of Bach[7] to monophonic Irish folk music generation[25, 26], and include composition applications such as musical inpainting to generate a melody to fill in the musical gap between two melodies[27] and musical interpolation to generate a set of melodies which incrementally move from one melody to another[28]. However, the complex nature of these generative models means that people often need some technical expertise and knowledge of these algorithms in order to use and adapt them effectively. This makes such approaches difficult for people, especially non-experts, to understand and manipulate.

In this paper, we focus very much on the explainability of the AI model itself and its output. In particular, we examine how semantically meaningful labels can be applied to properties of AI models to provide the opportunity for users to interpret and understand some aspects of the model and its generated output. To date, there have been few comparisons of the performance of explainable generative models for music. For example, research has compared the performance of a novel convolutional-variational neural network for music generation to other neural networks[29] in terms of the information rate of generated music – a measure of musical structure. However, such comparisons compare across models, do not compare the configurations of models themselves or examine a range of semantically meaningful features. We aim to compare the effect of meaningful labels on AI models in different configurations and with different datasets. To reduce the complexity of these combinatorial analyses, we select the core music generation task of generating monophonic melodies. In this way, we contribute the first in-depth analysis of how different AI model architectures and datasets affect music generation when explainable attributes are used. Future work can build on our findings to compare the effects of explainable attributes on more complex polyphonic melody generation and later accompaniment and association. By taking this approach, we improve the field′s understanding of state-of-

the-art deep learning generative models to help inform future generative model development and refinement – understanding where we are today informs where we might go in the future.

## 2.1 Latent spaces for music generation

AI models for music generation range from probability based models such as Markov chains[30, 31] to deep learning techniques explored in this paper[27, 32]. Probabilistic approaches typically offer more controllable music generation with lower computational and dataset requirements, but their outputs are often less novel than those achieved by deep learning approaches. A wide variety of deep learning generative models of music have been developed in recent years[24] and have been demonstrated to generate convincing musical outputs[1–3]. Briot and Pachet[24] provide a thorough survey of deep learning architectures and models used for music generation including variational auto encoder (VAE), restricted Boltzmann machine (RBM), recurrent neural network (RNN), convolutional neural network (CNN), generative adversarial networks (GAN), reinforcement learning (RL) and compound architectures of these approaches. As noted by [32], two of the most popular deep learning approaches to generative AI are generative adversarial networks (GANs)[33] and variational auto-encoders (VAEs)[34]. In this paper, we examine VAEs as they have been demonstrated to be capable of creative tasks including music generation[35, 36], music inpainting[27] and music interpolation[28]. Moreover, whilst comparisons of VAE approaches have to date examined image generation in terms of computation time and re-generation accuracy[37], there has been no systematic comparison of VAEs for music generation, nor in terms of interpretable features. Some recent VAE systems have exposed the latent space of generative music models to users[20, 27, 38–40] as a way for users to navigate the latent space to generate music. These approaches offer increased control of the AI models[40] and increased structure and labelling of the models[27], both of which contribute to increasing explainability. Given the research interest in making latent spaces more explainable, we explore what effect different AI model configurations and training datasets have on one of these approaches when explainable attributes are applied. In this paper, we explore these questions for the popular VAE architecture which shows promise as a deep learning approach to music generation[28].

A VAE architecture consists of 1) an encoder which encodes training data into 2) a multi-dimensional latent space which is used by 3) a decoder which decodes data from the latent space to generate data in the style of the training data as illustrated in Fig. 1. Modifying values of the latent space dimensions will have an effect on the generated data. The challenge for explainable VAE data generation is how to offer users meaningful control of the

generated data given that the latent space is the result of unsupervised learning with no meaningful structure. There are two main approaches to attribute-based control of generative models: unsupervised disentanglement learning, and supervised regularisation methods[11]. However, unsupervised disentanglement necessarily requires some post-training analysis to identify the possible meaning of the disentangled dimensions (ibid.).
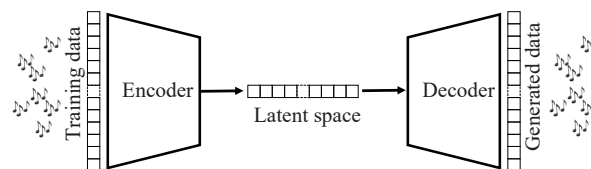


Fig. 1    Variational auto-encoder architecture

## 3 Research questions

As outlined in previous sections, there are many approaches to music generation using deep learning models, and each year new models are added to the repertoire of music generation systems. However, to date there has been no systematic analysis of how different training datasets and AI model architectures might impact the performance of XAI models for music. For example, to date the only experiments on the effect of latent space dimensionality on model performance have been conducted on images[41]. Our core research question is: What effects do different AI model architectures, configurations, and training data have on the performance of generative AI models for music with explainable features? No research has been undertaken to establish these effects to date. In answering this question, we help researchers to understand the properties of state-of-the-art generative music architectures and so help to build a baseline from which to explore many more model features and generative architectures.

This paper begins to address the core research question by systematically asking the following questions about the performance of VAE generative models with explainable features:

**RQ1.** What is the effect of VAE model architectures on performance?

**RQ2.** What effect do the musical features imposed on the latent space have on performance?

**RQ3.** What effect does the size of latent space have on performance?

**RQ4.** What effect do training datasets have on performance?

## 4 Methods

Following [27] which demonstrates that VAEs are successful in generating short pieces of monophonic music, we restrict our music generation to monophonic meas-

ures of music represented by 24 characters. Each character can represent a musical note, a note continuation, or a rest.

## 4.1 Candidate AI models

As a first step in understanding what effect explainable features have on the performance of generative AI model architectures, we compare two representative example VAE generative music models – measureVAE[27] and adversarialVAE[32]. Both approaches build on a VAE architecture (Section 2.1) to generate music but differ in terms of how musically semantic information is applied to the music generation with measureVAE imposing regularised dimensions on the latent space and adversarialVAE adding control attributes to the decoder.

### 4.1.1 MeasureVAE

The popular measureVAE implementation[1, 2][27, 42] has been demonstrated to be "successful in modeling individual measures of music"[27]. MeasureVAE uses a bidirectional recurrent neural network (RNN) for the encoder, and a combination of two unidirectional RNNs and linear stacks for the decoder[27]. The generated music can be varied by modifying the values of the dimensions in the latent space but the relationships between the dimensions and the music produced is not meaningful to people. To improve the explainability of the measureVAE, we can apply latent space regularisation (LSR)[43] when training the VAE. LSR has been widely used to allow more user controlled generation of images[44] and music[42, 45]. Following **(Pati and Lerch)**[11, 42], we use LSR to force a small number of dimensions of the latent space to represent specific musical attributes (see Section 4.3). These regularised dimensions are the explainable features of the measureVAE model which increase the explainability of the generative model. Fig. 2 illustrates the VAE architecture with 4 regularised dimensions in the latent space. See [27] for details of the measureVAE model architecture.
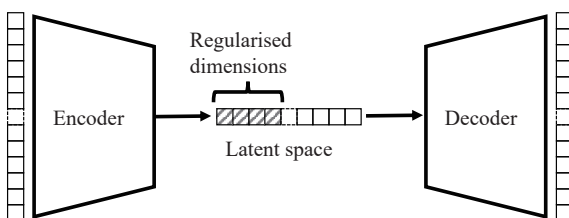


Fig. 2    Variational auto-encoder with latent space regularisation of 4 dimensions

In measureVAE, which is a typical VAE encoder-decoder architecture[11], data points $\boldsymbol{x}$ in a high-dimensional space $\boldsymbol{X}$ are mapped to a low-dimensional latent space $\boldsymbol{Z}$ using the encoder, where latent vectors are represen-

[2] https://github.com/ashispati/AttributeModelling

ted with $\boldsymbol{z}$. The latent vectors, $\boldsymbol{z}$ are mapped back to the data space $\boldsymbol{X}$ via the decoder. Latent vector, $\boldsymbol{z}$, is considered as a random variable and generation process is defined with the sampling processes of $\boldsymbol{z} - p(\boldsymbol{z})$ and $\boldsymbol{x} - p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$. $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ is the $\theta$ parameterised decoder architecture and $p(\boldsymbol{z})$ is the prior distribution over the latent space $\boldsymbol{Z}$, as per the variational inference. The encoder is represented with $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$, which is the posterior parameterised by $\phi$. In this context, the loss function is defined with the equation below as also defined in [11]:

$$L_{VAE}(\theta, \phi) = L_R(\theta, \phi) + L_{KLD}(\theta, \phi). \qquad (1)$$

In (1), the first term, $L_R$, represents the reconstruction loss, which is the L2 norm between $\boldsymbol{x}$, original data vector, and $\hat{\boldsymbol{x}}$, its reconstruction version. The second, $L_{KLD}$, represents the KL-divergence regularisation, typical to VAEs.

To apply the latent space regularisation in the context of measureVAE, firstly an attribute distance matrix is defined, which is $D_a \in \mathbf{R}^{m \times m}$, where $m$ is training examples in a mini-batch, as in [11]:

$$D_a(i, j) = a(\boldsymbol{x}_i) - a(\boldsymbol{x}_j) \qquad (2)$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ represent arbitrary data vectors and $a(\cdot)$ is the calculation of any attribute for the data vector, $\boldsymbol{x}$. Then, another distance matrix, $D_r \in \mathbf{R}^{m \times m}$ is calculated for the regularised dimension, $r$, of the latent vectors $\boldsymbol{z}$:

$$D_r(i, j) = z_i^r - z_j^r \qquad (3)$$

where $z_i^r$ and $z_j^r$ are the $r$-th dimension values of the arbitrary latent vectors $z_i$ and $z_j$. Lastly, the additional loss term for the latent space regularisation is defined with the following equation, as in [11]:

$$L_{r,a} = MAE(\tanh(\delta D_r) - \text{sgn}(D_a) \qquad (4)$$

which is added to the VAE loss in (1). In (1), $MAE$ is the mean absolute error, tanh is the hyperbolic tangent, sgn is the sign function and $\delta$ is a parameter that controls the spread of the posterior. Due to this additional term, increasing or decreasing relationships between the calculated attributes for $x_i$ and $x_j$ are similarly reflected to the relationship between $z_i^r$ and $z_j^r$, respectively. The code for this measureVAE implementation based on [11] can be found here[3].

### 4.1.2 AdversarialVAE

The adversarialVAE[32] uses a one-layer bidirectional gated recurrent unit (GRU) for the encoder followed by linear layers (MLP) for the mean and variance of the variational sampling at the latent space bottleneck, and a two-layer GRU for the decoder followed by a linear layer (MLP) which in contrast for measureVAE uses both the

[3] https://github.com/bbanar2/Exploring_XAI_in_GenMus_via_L-SR

latent space and additional control attributes to generate the music. The latent space itself has an adversarial classifier-discriminator added which "induces the encoder to remove the attribute information from the latent vector" (ibid.) as illustrated in Fig. 3. In contrast to measure-VAE where specific dimensions of the latent space are tied to semantic musical features, music generation with the adversarialVAE is controlled by musical attributes fed to the decoder – these are the explainable features of the adversarialVAE model.
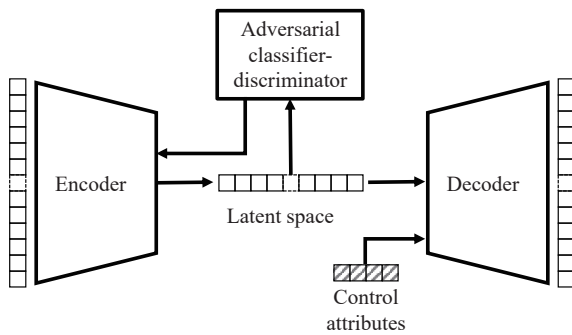


Fig. 3    Variational auto-encoder with adversarial classifier and decoder control attributes

The adversarialVAE model as defined in [32], similar to measureVAE, has an additional loss term on top of the original VAE loss, which is defined in (1). The additional loss term here, denoted as $L_D$, is adversarial and it belongs to a separate architecture that is a classifier-discriminator which consists of linear layers with tanh activations, except for the last layer where sigmoid activation is utilised as per the classification task. The objective of this additional classifier-discriminator is to determine the value of a musical attribute using discretely defined levels given the latent vector, $z$, of a musical sequence by learning a probability distribution $s_\psi$, where $\psi$ is the parameterised classifier-discriminator.

To construct $L_D$, firstly $N$ musical attributes are defined. Then, based on the training data, each attribute is quantised into $K$ bins (specifically, $K = 8$ in this study), where $\mu$-law compression is used as in [32] to obtain equal number of samples in each bin given the characteristics of the training data. Labels of each sample in the training data are one-hot encoded according to the quantised bin that the sample belongs to. Considering $N$ musical metrics and $K$ discrete levels, each sample yields in a matrix $B \in \mathbf{R}^{N \times K}$ for the target musical attributes, which is the output of the classifier-discriminator network. As per the adversarial objective defined in [32], the encoder tries to prevent the classifier-discriminator to predict to correct targets for the musical attributes, therefore the targets from the perspective of the encoder are defined as $1 - B$, where $1$ is the matrix of ones as per the one-hot encoding.

After having the $B$ and $1 - B$ matrices, the $L_D$ is defined as follows:

$$L_D(\psi \mid \phi) = -E_{q_\phi(z|x)}[\log(s_\psi(B \mid z))] \qquad (5)$$

$$L_D(\phi \mid \psi) = -E_{q_\phi(z|x)}[\log(s_\psi(1 - B \mid z))] \qquad (6)$$

where $\phi$ is the parameterised encoder architecture, $\psi$ is the parameterised classifier-discriminator, $q_\phi(z \mid x)$ is the posterior distribution denoting the encoder in accordance with the notation in (1).

Then, the overall loss becomes

$$L(\phi, \theta \mid \psi) = L_R(\theta, \phi) + L_{KLD}(\theta, \phi) + L_D(\phi \mid \psi) \qquad (7)$$

$$L(\psi \mid \phi) = L_D(\phi \mid \psi) \qquad (8)$$

following a similar notation as above.

See [32] for details of the adversarialVAE model architecture and the repository here[4] for the implementation of adversarialVAE based on [32].

## 4.2  Datasets

Generative AI music models are typically trained and evaluated on one musical dataset such as monophonic Irish folk melodies[26] which have been used to train and test measureVAE[27]. However, as noted in [46], different musical genres have different musical features which may have an impact on the performance of a generative AI model and its explainability potentially.

In this paper, we use the frequently used Irish folk dataset[26], compare and contrast this with datasets of Turkish folk music, pop music and classical music as described in this section. Table 1 presents key features of the datasets used including their musical features from Section 4.3.

### 4.2.1  Irish folk dataset

The Irish folk dataset contains 20 000 monophonic Irish folk melodies[26][5] from which 5.6 M notes are extracted for these experiments. The dataset has the highest note range and density of the music used in these experiments meaning that it is the most complex musically. It is also by far the largest dataset used in this experiment and is commonly used in generative AI research.

### 4.2.2  Turkish makam dataset

The Turkish makam dataset[47][6] as used in [48] which consists of approximately 2 200 musical scores related to Turkish makam music which is a form of Turkish folk music. This results in approximately 755 K musical notes of monophonic folk songs. The Turkish makam dataset has similarly high mean note density, note range and rhythmic complexity to the Irish folk music dataset, suggesting similarly high musical complexity. The Turkish

---

Table 1     Summary statistics of the datasets

| Dataset | Note density | | Note range | | Rhy. complexity | | Avg. Int. jump | | Genres | No. of notes |
|---------|------|------|------|------|------|------|------|------|--------|--------------|
| | Mean | STD | Mean | STD | Mean | STD | Mean | STD | | |
| Muse Bach | 3.206 | 3.346 | 3.336 | 4.073 | 1.742 | 2.150 | 0.761 | 1.933 | Classical | 965 244 |
| Lakh clean | 2.504 | 3.178 | 1.515 | 3.381 | 1.462 | 2.146 | 2.865 | 2.840 | Mainly pop | 1 697 053 |
| Turkish makam | 6.609 | 3.161 | 4.972 | 2.828 | 3.907 | 2.456 | 1.564 | 0.818 | Folk | 755 785 |
| Irish folk | 6.765 | 2.056 | 7.809 | 3.440 | 3.756 | 1.998 | 2.653 | 2.653 | Folk | 5 662 498 |

makam dataset is the smallest dataset used in these experiments.

### 4.2.3   Muse Bach dataset

MuseData[7] consists of Baroque to early classical music, including both monophonic and polyphonic instrumental pieces. Given the wide range of styles contained in MuseData, we selected all pieces composed by Bach in MuseData to provide a coherent style of music given. 593 Bach pieces were extracted from MuseData, resulting in 4 531 single-lined melodies and almost 1 M musical notes which we refer to as the Muse Bach dataset. The Muse Bach dataset has the lowest mean average interval jump of the datasets used, and middling note density, range and complexity.

### 4.2.4   Lakh clean dataset

The Lakh dataset[49][8] contains 176 581 unique musical instrument digital interface (MIDI) files. For this experiment, we use a subset of the Lakh dataset – the clean MIDI (sub)dataset which contains pieces by 2 199 artists. The distribution of the genres in the clean MIDI dataset is: 33% pop, 32% rock, 13% Jazz and Blues, 10% R&B, and 12% other, providing a dataset of contemporary popular music. Almost 7 K monophonic clips were extracted from these pieces, resulting in approximately 1.7 M notes which we refer to as the Lakh clean dataset. This dataset has the lowest mean note range and rhythmic complexity of the datasets used in these experiments, suggesting that it contains some of the least musically complex music.

### 4.2.5   Data preparation

Each dataset was converted into a measure based ABC format using the MIDI2ABC functions in EasyABC[9]. Each measure is represented by 24 characters including notes names, continuation and rest tokens. As the VAE models in this experiment work with monophonic melodies, single line melodies were extracted from the dataset using EasyABC. All musical instruments were then separated into separate files and any remaining chords were converted into single notes based on the chord′s highest pitch.

## 4.3   Musical features

There are many musical features that could be im-

posed on music generation. For example, the popular jSymbolic[50] offers analysis of 246 unique musical features. In this research, we use a subset of the most frequently used features in music research, and follow [19] to select the following musical attributes:

1) **Note density (ND)** – the number of notes in a measure;

2) **Note range (NR)** – the highest minus lowest pitch in a measure;

3) **Rhythmic complexity (RC)** – how syncopated a musical measure is[51];

4) **Average interval jump (AIJ)** – the average of the absolute difference between adjacent notes in a measure.

These features cover both rhythmical properties (ND and RC) and melodic properties (NR and AIJ). They are used to 1) characterise the musical properties of the datasets (Section 4.2) used to train the AI models; and 2) as attributes of control of music generation – because the features have some musical meaning they serve to increase the explainability of the AI models.

## 5   Experimental setting: Comparing VAE model architectures

MeasureVAE and adversarialVAE were compared using the Irish folk dataset for training to explore RQ1. Of the 20 000 monophonic Irish folk melodies, 14 000 were used as the training set, 3 000 as test sets, and 3 000 as validation sets. Models were compared in terms of the evaluation metrics outlined in Section 5.1.

The experiment learning rate was set to $1 \times 10^{-4}$ (optimized using Adam[52]). Both models were trained on a GPU for a total of 50 iterations with a batch size of 64 for all data. $\alpha = 0.1$, $\beta = 0.1$, $\gamma = 0.2$ were used in the VAE loss function. As measureVAE uses 256 dimensions of latent space whereas adversarialVAE uses 128 dimensions testing was undertaken with both 128 and 256 dimensions[41]. Both models had 4 musical features imposed on them (from Section 4.3).

We use musical measures for generative output and training in keeping with state-of-the-art music generation research[11, 27] and typical of the musical elements used in current generative AI tasks. Each measure is represented by 24 characters which include note names such as A3, G5 and so on, continuation tokens and rest tokens.

---

## 5.1 Evaluation metrics

We evaluate the AI models in terms of standard measures of:

1) **Reconstruction accuracy.** How well the model can reconstruct any given input. We aim to maximise this. This is calculated by comparing the difference between the input melody and generated melody, and averaging this over the whole dataset. Reconstruction accuracy is defined as follows:

$$RA(\boldsymbol{x}, \widehat{\boldsymbol{x}}) = \frac{100}{N} \sum_{i=1}^{N} \frac{1}{M_i} \sum_{j=1}^{M_i} Check(x_{ij}, \hat{x}_{ij}) \qquad (9)$$

where $x_i$ is the input sequence, $\hat{x}_i$ is the reconstructed sequence, $N$ is the number of samples in a dataset and $M$ is the sequence length. The check function compares two corresponding elements in the original and the reconstructed sequences as follows:

$$Check(x_{ij}, \hat{x}_{ij}) = \begin{cases} 1, & \text{if } x_{ij} = \hat{x}_{ij} \\ 0, & \text{otherwise.} \end{cases} \qquad (10)$$

2) **Reconstruction efficiency.** How well the model generates music with respect to the characteristics of its training dataset and also the provided input sequence when musical parameters are changed. We aim to maximise this. To calculate this measure, we split our data into two categories, where an attribute $a_r \geq 0$ and $a_r < 0$. Then, we calculate the mean latent vectors $\boldsymbol{z_a}$ and $\boldsymbol{z_{a0}}$ for each of these subsets, respectively. These procedures provide us with a general picture of latent vectors with respect to the presence of the musical attribute. Then, using these mean vectors, for each sample in our data, we get their latent vectors, $\boldsymbol{z}$ and apply the following interpolation.

$$\boldsymbol{z_{\textbf{resulting}}} = \boldsymbol{z} + \mu(\boldsymbol{z_a} - \boldsymbol{z_{a0}}) \qquad (11)$$

where $\mu \in \{-0.5, -0.4, \cdots, 0.4, 0.5\}$ with 11 possible values. Then, $\boldsymbol{z_{\textbf{resulting}}}$ vectors are decoded into generated music sequences and for each generated music sequence, $\hat{\boldsymbol{x}}$, and input music sequence, $\boldsymbol{x}$, we check the cosine similarity between these sequences using the following formula:

$$CS(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{\boldsymbol{x} \cdot \hat{\boldsymbol{x}}}{\|\boldsymbol{x}\|\|\hat{\boldsymbol{x}}\|}. \qquad (12)$$

Each musical attribute is interpolated separately, and then the average similarity of each interpolation is calculated.

3) **Attribute independence.** How resilient an attribute is to change by other attributes. We aim to maximise this. This is calculated as getting the maximum value for Spearman′s correlation coefficient between the attribute value, $a(\boldsymbol{x})$, and each dimension of the latent space, $z_d$[53]. Then, these correlation coefficients are averaged for all of the musical attributes.

Implementations of the reconstruction accuracy and attribute independence are included in these repositories[10, 11] as in [11, 32].

## 5.2 Results

### 5.2.1 Reconstruction accuracy

Table 2 shows the reconstruction accuracy scores for 128 and 256 latent space dimensions for measureVAE and adversarialVAE. Results show that measureVAE reconstruction′s accuracy outperformed the adversarialVAE in both 128 and 256 dimension configurations, achieving a high of 99.6% for the validation set with 256 dimensions. Both models performed better in 256 dimensions than in 128 dimensions. This may be because the higher number of dimensions makes it easier to decompress the latent space.

Table 2    The reconstruction accuracy of measureVAE and adversarialVAE models on training, test and validation data from Irish folk dataset

| Dims. | Dataset | Reconstruction accuracy (%) | |
| --- | --- | --- | --- |
| | | MeasureVAE | AdversarialVAE |
| | Train | 97.359 | 94.721 |
| 128 | Test | 96.743 | 93.626 |
| | Validate | 96.207 | 93.572 |
| | Train | 99.824 | 95.866 |
| 256 | Test | 99.459 | 95.030 |
| | Validate | 99.674 | 95.268 |

### 5.2.2 Reconstruction efficiency

Fig. 4 illustrates the comparative reconstruction efficiency for measureVAE and adversarialVAE with 128 and 256 latent dimensions, as summarised in Table 3. Results show that the number of dimensions of latent space (128 or 256) did not have a noticeable effect on the reconstruction efficiency. Regardless of the number of dimensions, measureVAE had higher reconstruction efficiency than adversarialVAE, with largest difference between reconstruction efficiency at $\mu = 0.0$.

**Generated outputs.** Figs. 5 and 6 illustrate example outputs of the measureVAE and adversarialVAE models respectively for 256 latent dimensions. Figs. 5(a) and 6(a) show the input notes for the AI model, and Figs. 5(b) and 6(b) to Figs. 5(e) and 6(e) show the melody produced by the model after interpolating for one musical attribute at $\mu = 0.3$. Each shows the melody generated for a different musical feature: Figs. 5(b) and 6(b)
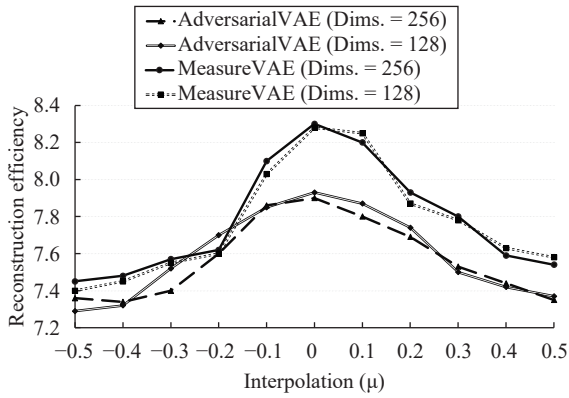
Fig. 4    Reconstruction efficiency for measureVAE and adversarialVAE with 128 and 256 latent dimensions

Table 3    The mean and standard deviation of reconstruction efficiency for measureVAE and adversarialVAE models with 128 and 256 latent dimensions

| Dims. | Evaluation | Reconstruction efficiency | |
|---|---|---|---|
| | | MeasureVAE | AdversarialVAE |
| 128 | Mean | 7.760 | 7.589 |
| | S.D. | 0.307 | 0.234 |
| 256 | Mean | 7.775 | 7.567 |
| | S.D. | 0.306 | 0.213 |

show note range increase; Figs. 5(c) and 6(c) show note density increase; Figs. 5(d) and 6(d) show rhythmic complexity increase; Figs. 5(e) and 6(e) show average interval jump increase.

Inspecting Figs. 5 and 6 suggests that the two models

produce different sounding music to each other for increased note range (b) with measureVAE producing a measure with larger changes between the notes. Furthermore, increasing the note range measureVAE also increased the average interval jump in Fig. 5(b), whereas adversarialVAE produced a measure which shifted most of the original melody upwards in pitch except for the final note which was shifted down to produce the required increase in note range in Fig. 6(b). This difference is illus-
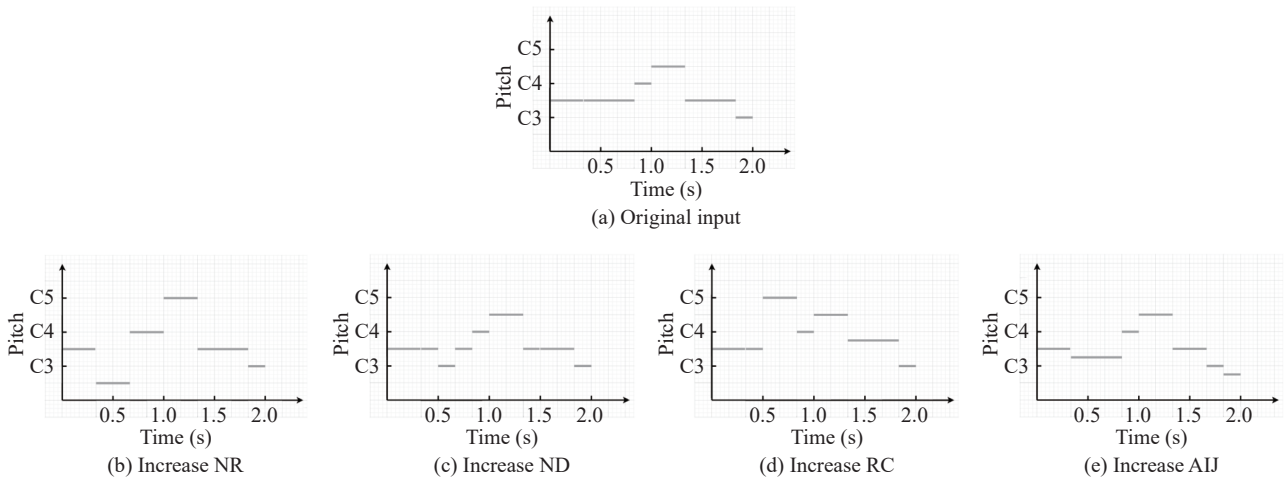


(a) Original input



(b) Increase NR      (c) Increase ND      (d) Increase RC      (e) Increase AIJ

Fig. 5    Music generated by measureVAE for the input melody is shown in (a); (b) to (e) show variation of four musical attributes for $\mu = 0.3$.



(a) Original input



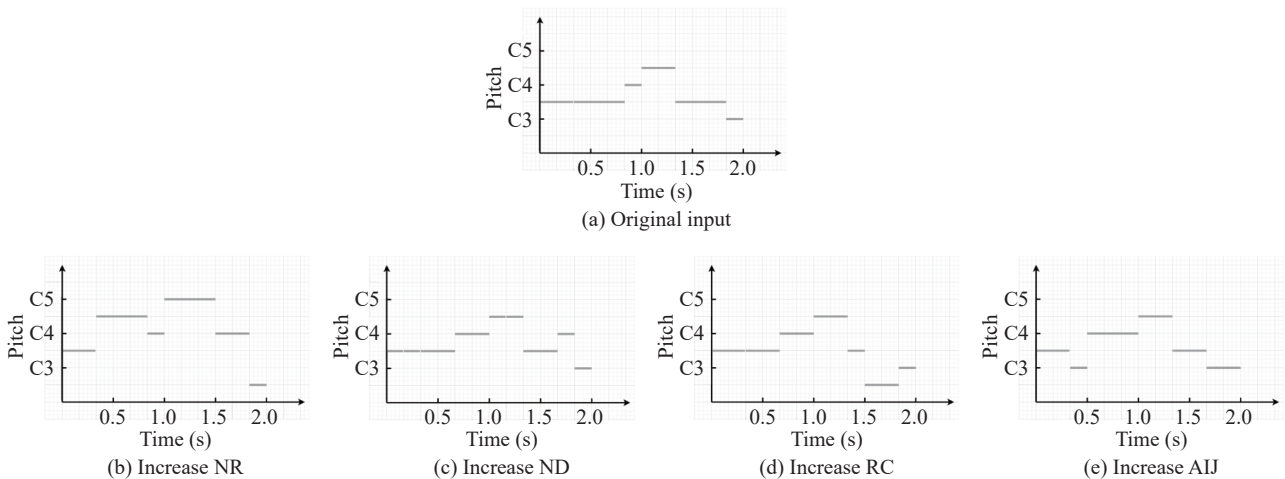(b) Increase NR      (c) Increase ND      (d) Increase RC      (e) Increase AIJ

Fig. 6    Music generated by adversarialVAE for the input melody is shown in (a); (b) to (e) show variation of four musical attributes for $\mu = 0.3$.

trated by calculating the Spearman′s correlation $r$ between the musical attribute values. In this case, we see that the correlation between increase in note range and the generated measure′s AIJ is $r = 0.286$ for measure-VAE (weak correlation) and $r = 0.154$ for adversarial-VAE (no correlation), i.e., AIJ increases weakly with increases in NR for measureVAE but not for adversarial-VAE.

For note density, measureVAE produces music with a higher note density (Fig. 5(c)) than adversarialVAE (Fig. 6(c)) when there is an increase in note density attribute applied to the generation. Interestingly, measure-VAE achieves the increased note density by adding an upward run of notes to the measure which also increases the rhythmic complexity, whereas adversarialVAE′s increase in note density reduces rhythmic complexity compared to the original. Calculating Spearman′s correlation again, we see that in this case of increasing note density, the correlation of increased ND to output RC for measureVAE is $r = 0.341$ (weak correlation) and adversarial-VAE $r = 0.178$ (no correlation). This suggests that RC increases weakly when ND is increased with measure-VAE, but not with adversarialVAE.

In contrast to NR and ND, both AI models generate similar music to each other for increased rhythmic complexity, as shown in Figs. 5(d) and 6(d) and also for increased average interval jump, as shown in Figs. 5(e) and 6(e).

### 5.2.3 Attribute independence

Table 4 shows the results for attribute independence tests of measureVAE and adversarialVAE for musical attributes note range (NR), note density (ND), rhythmic complexity (RC) and average interval jump (AIJ). Results indicate that adversarialVAE performs better than measureVAE for attribute independence for all attributes and latent dimensions except for note range with 128 latent dimensions where measureVAE performs marginally better. Note density had the highest attribute independence for both models and for both 128 and 256 dimensions. This may be because ND is a measure of the num-

ber of notes in a measure, meaning that it is easier to distinguish compared to other metrics such as rhythmic complexity which relies on the ability to differentiate between different musical beat types. Rhythmic complexity shows the largest difference between measureVAE (0.878) and adversarialVAE (0.943) for 128 dimensions, whereas average interval jump shows the largest difference between measureVAE (0.765) and adversarialVAE (0.875) for 256 dimensions. The higher independence of adversarialVAE attributes may be due to the use of the adversarial classifier-discriminator to impose musical attributes and the additional phase in the training process that optimises the decisions rather than trying to lower the loss function′s value.

## 6 Experimental setting: Latent space configuration and training datasets

In this work, we are interested in contributing towards understanding how generative models which create music in given styles can be better interpreted and manipulated by users. To this end, we now explore the performance of measureVAE in more details as it has higher reconstruction accuracy and reconstruction efficiency than adversarialVAE (Section 5).

In this section, we examine the impact that different configurations of musical (explainable) features (RQ2), sizes of latent spaces (RQ3) and different training datasets (RQ4) might have on the performance and explainability of measureVAE. To examine this systematically, we undertook a combinatorial experiment examining the effect of musical dataset ($n = 4$), number of latent dimensions ($n = 7$), and number of regularized musical attributes ($n = 2$) on evaluation metrics (Section 6.1):

1) **Datasets:** Muse Bach, Lakh clean, Turkish makam, Irish folk datasets (Section 4.2) – to compare a range of musical genres;

2) **Latent dimensions:** 4, 8, 16, 32, 64, 128 and 256 – to capture a typical range of latent space sizes;

3) **Regularised dimensions:** 2 or 4 – using musical features (Section 4.3) in the latent space – ND & RC, NR & AIJ or ND & NR & RC & AIJ.

For each combination of the above, we trained a measureVAE model for 25 epochs. We use Adam[52] as the optimizer of the model with learning rate $= 1 \times 10^{-5}$, $\beta_1 = 0.999\,9$ and $\epsilon = 1 \times 10^{-8}$. The model is trained on a single rtx 6000 GPU following a similar setting of [42], taking on average of 2.5 hours per epoch.

### 6.1 Evaluation metrics

We evaluate the combinations of datasets, latent space dimensions and regularised dimensions in terms of standard measures of:

1) **Reconstruction accuracy.** How well the model can reconstruct a given input – as in Section 5.1.

Table 4   The attribute independence of measureVAE and adversarialVAE models for note range, note density, rhythmic complexity and average interval jump attributes

| Dims. | Attributes | Attribute independence | |
| --- | --- | --- | --- |
| | | MeasureVAE | AdversarialVAE |
| 128 | NR | 0.945 | 0.941 |
| | ND | 0.976 | 0.986 |
| | RC | 0.878 | 0.943 |
| | AIJ | 0.870 | 0.899 |
| 256 | NR | 0.928 | 0.936 |
| | ND | 0.969 | 0.981 |
| | RC | 0.812 | 0.938 |
| | AIJ | 0.765 | 0.875 |

2) **Loss.** Loss scores are calculated by the sum of VAE loss (KL-divergence and reconstruction loss, typical to VAE architectures[54]) and the loss of the latent space regularization[11, 19, 42] – as in Section 4.1. We aim to minimize this.

3) **Attribute interpretability.** How well a musical attribute can be predicted using only one LSR dimension in the latent space[11, 42], we aim to maximise this. We suggest that higher interpretability scores contribute to better explainability, it indicates less entangled semantic dimensions[55].

## 6.2 Results

Tables 5 and 6 show the results for the combinatorial experiment, including datasets (Muse Bach, Lakh clean, Turkish makam, Irish folk), latent dimensions (4, 8, 16, 32, 64, 128, 256), regularised dimensions (ND, NR, RC, AIJ), loss scores and reconstruction accuracy scores

(Table 5), and musical attribute interpretability scores (Table 6).

### 6.2.1 Reconstruction accuracy and loss

Table 5 shows that with 32 or more latent space dimensions measureVAE achieves reconstruction accuracy scores above 99% and loss scores below 0.2 for all datasets and number of regularised dimensions. This suggests that measureVAE is capable of generating music across folk, pop, rock, jazz and blues, R&B and classical music.

Figs. 7 and 8 illustrate the average reconstruction accuracy and loss respectively for each dataset across all dimensions and with 2 (ND & RC) or 4 regularised dimensions.

The results indicate that measureVAE performed best with the Lakh clean dataset with the lowest loss scores and the highest reconstruction accuracy scores for both 2 and 4 regularised dimensions. This may be a result of the Lakh clean dataset being less musically complex with the lowest mean note density, note range, and rhythmic com-

Table 5　Loss and reconstruction accuracy scores for measureVAE

| Latent dimensions | Loss scores | | | | Reconstruction accuracy scores (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | Muse | Lakh | Turkish | Irish | Muse | Lakh | Turkish | Irish |
| 2 regularised dimensions (ND&RC) | | | | | | | | |
| 4 | 0.296 | 0.264 | 0.658 | 0.492 | 91.009 | 93.928 | 79.925 | 86.488 |
| 8 | 0.122 | 0.106 | 0.364 | 0.118 | 96.873 | 97.909 | 91.151 | 97.185 |
| 16 | 0.037 | 0.036 | 0.083 | 0.026 | 99.284 | 99.517 | 98.260 | 99.688 |
| 32 | 0.018 | 0.018 | 0.027 | 0.012 | 99.894 | 99.900 | 99.853 | 99.942 |
| 64 | 0.040 | 0.025 | 0.031 | 0.012 | 99.203 | 99.777 | 99.827 | 99.939 |
| 128 | 0.059 | 0.014 | 0.029 | 0.014 | 99.828 | 99.958 | 99.837 | 99.951 |
| 256 | 0.011 | 0.017 | 0.027 | 0.013 | 99.947 | 99.920 | 99.879 | 99.912 |
| *Mean* | *0.084* | *0.068* | *0.174* | *0.098* | *98.005* | *98.701* | *95.533* | *97.586* |
| 2 regularised dimensions (NR&AIJ) | | | | | | | | |
| 4 | 0.372 | 0.265 | 0.823 | 0.706 | 89.972 | 94.162 | 75.627 | 82.661 |
| 8 | 0.163 | 0.145 | 0.512 | 0.190 | 95.788 | 97.924 | 89.187 | 96.552 |
| 16 | 0.088 | 0.067 | 0.256 | 0.083 | 98.532 | 99.109 | 96.168 | 99.389 |
| 32 | 0.049 | 0.116 | 0.137 | 0.058 | 99.649 | 98.340 | 99.181 | 99.796 |
| 64 | 0.041 | 0.047 | 0.128 | 0.055 | 99.799 | 99.662 | 99.496 | 99.841 |
| 128 | 0.042 | 0.029 | 0.146 | 0.054 | 99.764 | 99.909 | 99.323 | 99.876 |
| 256 | 0.040 | 0.033 | 0.136 | 0.052 | 99.806 | 99.893 | 99.448 | 99.870 |
| *Mean* | *0.114* | *0.100* | *0.306* | *0.171* | *97.616* | *98.428* | *94.061* | *96.855* |
| 4 regularised dimensions | | | | | | | | |
| 4 | 0.803 | 0.514 | 1.333 | 1.283 | 88.317 | 92.031 | 76.837 | 78.294 |
| 8 | 0.213 | 0.194 | 0.615 | 0.470 | 94.796 | 97.176 | 86.843 | 93.658 |
| 16 | 0.154 | 0.092 | 0.438 | 0.100 | 98.705 | 99.135 | 95.967 | 99.346 |
| 32 | 0.063 | 0.077 | 0.176 | 0.076 | 99.638 | 99.510 | 99.061 | 99.838 |
| 64 | 0.057 | 0.063 | 0.166 | 0.072 | 99.733 | 99.656 | 99.495 | 99.894 |
| 128 | 0.061 | 0.053 | 0.151 | 0.074 | 99.740 | 99.844 | 99.540 | 99.886 |
| 256 | 0.051 | 0.056 | 0.170 | 0.078 | 99.900 | 99.781 | 99.495 | 99.871 |
| *Mean* | *0.200* | *0.150* | *0.436* | *0.308* | *97.261* | *98.162* | *93.891* | *95.827* |

Table 6    Interpretability scores for measureVAE. Bold indicates the highest score of 2 and 4 dimensions.

| Latent dimensions | Muse | Lakh | Turkish | Irish | Muse | Lakh | Turkish | Irish |
|---|---|---|---|---|---|---|---|---|
| | 2 regularised dimensions (ND&RC) | | | | 4 regularised dimensions | | | |
| | Note density (ND) interpretability scores | | | | | | | |
| 4 | **0.999** | **0.978** | **0.996** | **0.992** | 0.982 | 0.956 | 0.940 | 0.931 |
| 8 | 0.996 | **0.992** | **1.000** | **0.996** | **0.998** | 0.977 | 0.998 | 0.991 |
| 16 | **0.997** | **0.990** | **0.999** | **0.997** | 0.708 | 0.986 | 0.584 | 0.996 |
| 32 | **0.996** | **0.983** | **0.998** | 0.994 | 0.992 | 0.975 | 0.994 | **0.995** |
| 64 | **0.996** | **0.977** | **0.997** | 0.992 | 0.990 | 0.975 | 0.994 | **0.995** |
| 128 | 0.987 | **0.981** | **0.998** | 0.990 | **0.989** | 0.969 | 0.994 | **0.997** |
| 256 | **0.990** | **0.978** | **0.999** | 0.989 | 0.989 | 0.977 | 0.994 | **0.994** |
| *Mean* | *0.995* | *0.983* | *0.998* | *0.993* | *0.950* | *0.974* | *0.928* | *0.985* |
| | Rhythmic complexity (RC) interpretability scores | | | | | | | |
| 4 | **0.987** | **0.956** | **0.970** | **0.732** | 0.946 | 0.927 | 0.925 | 0.712 |
| 8 | 0.985 | **0.977** | **0.991** | **0.832** | 0.989 | 0.959 | 0.989 | 0.774 |
| 16 | **0.980** | 0.966 | **0.983** | **0.815** | 0.955 | 0.974 | 0.580 | 0.802 |
| 32 | **0.989** | 0.952 | **0.977** | **0.806** | 0.985 | 0.965 | 0.977 | 0.800 |
| 64 | **0.981** | 0.960 | **0.973** | 0.769 | 0.975 | 0.967 | 0.971 | **0.801** |
| 128 | 0.909 | 0.937 | **0.981** | 0.761 | 0.980 | 0.945 | 0.976 | 0.800 |
| 256 | 0.975 | 0.955 | 0.970 | 0.769 | 0.976 | 0.961 | 0.975 | 0.802 |
| *Mean* | *0.972* | *0.957* | *0.978* | *0.783* | *0.972* | *0.957* | *0.913* | *0.784* |
| | 2 regularised dimensions (NR&AIJ) | | | | 4 regularised dimensions | | | |
| | Note range (NR) interpretability scores | | | | | | | |
| 4 | **0.961** | **0.864** | **0.953** | **0.950** | 0.922 | 0.811 | 0.460 | 0.736 |
| 8 | **0.984** | 0.950 | **0.993** | 0.955 | 0.981 | **0.961** | 0.993 | **0.991** |
| 16 | **0.975** | 0.958 | 0.945 | 0.963 | 0.540 | **0.971** | **0.974** | **0.993** |
| 32 | 0.977 | **0.969** | 0.976 | **0.990** | 0.983 | 0.966 | **0.988** | 0.987 |
| 64 | 0.975 | 0.965 | **0.983** | 0.988 | 0.976 | **0.975** | 0.982 | 0.988 |
| 128 | 0.971 | 0.968 | 0.970 | 0.989 | **0.976** | **0.971** | **0.980** | **0.991** |
| 256 | 0.974 | 0.968 | **0.983** | 0.957 | **0.978** | **0.971** | 0.977 | **0.990** |
| *Mean* | *0.974* | *0.949* | *0.972* | *0.970* | *0.908* | *0.947* | *0.908* | *0.954* |
| | Average interval jump (AIJ) interpretability scores | | | | | | | |
| 4 | **0.949** | **0.714** | 0.621 | **0.844** | 0.914 | 0.651 | **0.672** | 0.506 |
| 8 | 0.960 | **0.869** | **0.929** | **0.964** | **0.970** | 0.863 | 0.881 | 0.934 |
| 16 | **0.976** | 0.868 | 0.886 | **0.973** | 0.957 | **0.880** | **0.916** | 0.915 |
| 32 | **0.981** | **0.896** | **0.951** | **0.969** | 0.974 | 0.888 | 0.916 | 0.918 |
| 64 | 0.967 | 0.882 | **0.958** | **0.970** | **0.970** | **0.899** | 0.928 | 0.922 |
| 128 | **0.974** | **0.897** | 0.887 | **0.971** | 0.969 | 0.894 | **0.898** | 0.918 |
| 256 | **0.976** | **0.897** | **0.894** | **0.968** | 0.964 | 0.879 | 0.865 | 0.916 |
| *Mean* | *0.969* | *0.860* | *0.875* | *0.951* | *0.960* | *0.851* | *0.868* | *0.861* |

plexity (Table 1) arguably making the modelling task easier and suggesting that measureVAE may be more suited to less complex musical styles. MeasureVAE performed least well for the Turkish makam dataset which had the highest average loss scores and lowest reconstruction accuracy scores. MeasureVAE′s poor performance with the Turkish makam dataset may be due to the higher complexity of the music in the dataset with high mean note density and rhythmic complexity, and lower average interval jump than other datasets. Moreover, there are similar poor reconstruction accuracy and loss scores found for the Irish folk dataset which also has high mean
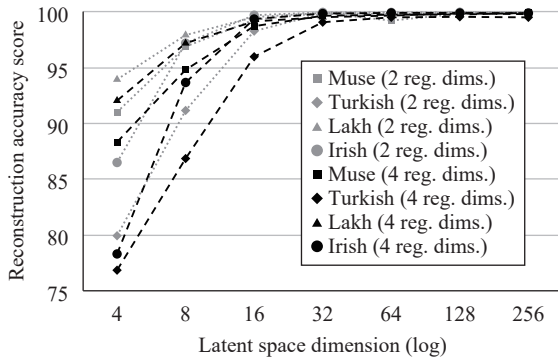
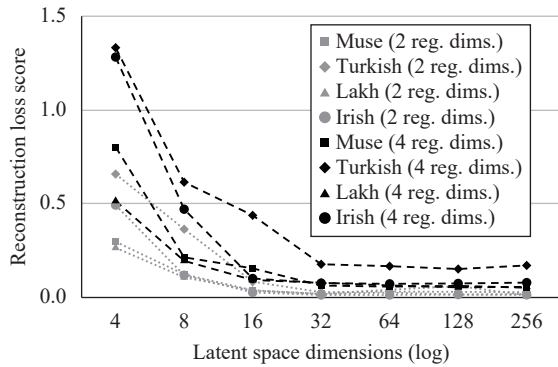Fig. 7    Reconstruction accuracy for measureVAE



Fig. 8    Loss for measureVAE

ND, NR and RC. The poor performance of the Turkish makam dataset may also be due to it being the smallest of the datasets used in this experiment, or the complex tonal features of Turkish makam music[48] which may not be captured in the musical metrics used in this experiment.

The results in Table 5 suggest that 2 regularised dimensions perform better than 4 for loss and reconstruction accuracy scores. The pair of ND&RC regularised dimensions performing better than NR&AIJ for both loss and reconstruction accuracy scores. The results also indicate that reconstruction accuracy and loss scores improve up to 32 dimensional latent space and plateau for larger latent spaces.

### 6.2.2    Attribute interpretability

Table 6 shows that measureVAE is capable of generating music with musical attribute interpretability scores of at least 0.8 for all of the tested musical attributes, though not for all latent space sizes. Interpretability scores for 4 regularised dimensions are over 0.95 for 69 of the possible 112 combinations, suggesting that measureVAE is able to generate music with good musical interpretability with 4 regularised dimensions. With 2 regularised dimensions interpretability scores are above 0.95 for note density, rhythmic complexity and note range for all datasets and the majority of latent dimensions (89 of 98) except RC for the Irish folk dataset which reached a maximum RC interpretability of 0.832 and AIJ for the Lakh

dataset (maximum 0.897). Overall, this suggests that measureVAE can successfully generate music across folk, classical and popular music with interpretable musical dimensions of control. The results also indicate that different datasets have different interpretability scores for different musical attributes, though it is not possible at this stage to say whether these interpretability scores are good or not.[55]

**Number of regularised dimensions.** For ease of inspection, Fig. 9 illustrates the average interpretability scores for all attributes for each dataset and 4 and 2 regularised dimensions (ND&RC). As suggested in Fig. 9 and detailed in Table 6, interpretability scores for each regularised attribute were in general higher for 2 regularised dimensions than 4 which is to be expected as it is easier to achieve successful and linearly independent regularisation in fewer dimensions. The exception to this are the rhythmic complexity interpretability scores for Muse Bach, Lakh clean and Irish folk datasets. For Muse Bach and Lakh clean datasets, the mean RC interpretability scores were equal. The Irish folk dataset′s mean rhythmic complexity interpretability scores are marginally higher for 4 regularised dimensions than 2 regularised due to the performance for latent spaces of 64, 128 and 256 dimensions. Inspecting the interpretability scores for each dimension, the data suggests that 2 regularised dimensions perform best compared to 4 regularised dimensions for note density and average interval jump interpretability scores. The higher mean interpretability scores for 2 regularised dimensions versus 4 may be due to only regularising 2 dimensions rather than the nature of the regularisation itself. The poor performance of rhythmic complexity for the Irish folk dataset may be a reflection of higher note density, note range and rhythmic complexity means for the Irish folk dataset, or it may be a reflection of the larger dataset size.

**Interpretability score performance.** Rankings of
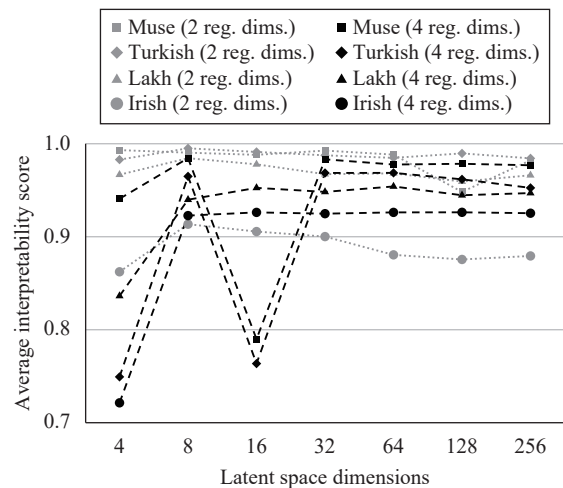


Fig. 9    Average    interpretability    for    measureVAE    latent regularised dimensions

interpretability scores for datasets are not consistent across the number of dimensions in the latent space. For example, for 4 regularised dimensions the highest rhythmic complexity interpretability score with 16 latent space dimensions is for the Lakh clean dataset (0.974) whereas for 32 latent space dimensions the highest RC interpretability score is for the Muse Bach dataset (0.985). For 4 regularised dimensions, the highest scoring attribute interpretability scores are consistent for 64, 128 and 256 dimensions, e.g., the Muse Bach dataset has the highest note range interpretability scores for latent spaces with 64, 128 and 256 dimensions. For 2 regularised dimensions, there is no consistently highest ranked attribute for interpretability across the range of latent space dimensions.

**Optimal configurations.** Given that high reconstruction accuracy scores and low loss scores reach at a 32 dimensional latent space for both 2 and 4 regularised dimensions, and that rankings of interpretability scores stabilise at 64 dimensions and above, the results suggest that a 32 or 64 dimensional latent space would be when optimal applying measureVAE across genres as it minimises latent space size and loss whilst maximising reconstruction accuracy and providing similar interpretability scores to higher dimensional spaces. However, careful selection of latent space size is recommended when measureVAE is to be used to generate specific genres of music. For example, 16 or even 8 latent dimensions are likely to be optimal for Irish folk music generation with 2 regularised dimensions given that its best interpretability performance is with an 8 dimensional latent space.

**Dataset performance.** Taking the average interpretability score across all latent space sizes, results suggest that for 4 regularised dimensions, the Irish folk dataset has the highest average interpretability scores for ND and NR, whereas Muse Bach has the highest for RC and AIJ. It is worth noting that the Irish folk dataset itself has the highest mean ND and NR and the second highest mean RC (Table 1), whereas the Muse Bach dataset has the second lowest mean RC and lowest mean AIJ, suggesting that there is not a correlation between the musical attributes of the datasets and the average interpretability scores of the measureVAE models. In terms of lowest average interpretability scores, the Turkish makam dataset has the lowest for ND, Irish folk dataset has the lowest for RC, the Muse Bach and Turkish makam datasets equally have the lowest for NR, and the Lakh clean dataset has the lowest average AIJ interpretability score.

For 2 regularised dimensions, the Turkish makam dataset has the highest average ND and RC interpretability scores whereas Lakh clean has the lowest ND, and Irish folk dataset has the lowest RC. Note that the Irish folk dataset has the lowest rhythmic complexity interpretability scores for both 2 and 4 regularised dimensions and for all sizes of latent spaces. Whilst the Irish folk dataset has a high mean rhythmic complexity, this does

not explain the poor RC interpretability score for the Irish folk dataset as the Turkish makam dataset has the highest mean RC and also highest RC interpretability score for 2 regularised dimensions, suggesting that there is not a correlation between the musical attributes of the datasets and the interpretability scores of the measureVAE models.

Fig. 9 and Table 6 indicate some anomalies in the interpretability scores. For 2 regularised dimensions, there is an outlying rhythmic complexity interpretability score for 128 latent dimensions. For 4 regularised dimensions in a 16 dimensional latent space, there are outlying interpretability scores for the Muse Bach dataset (ND and NR) and the Turkish makam dataset (ND and RC). On investigation of the data, no obvious reasons for these outlying results emerge. Instead, these anomalies suggest potential inconsistent performance of measureVAE for different datasets and latent dimension size, necessitating careful selection of latent space size for musical style.

## 7   Conclusions

This is the first time that two VAE models with semantic features for control of music generation have been systematically compared in terms of performance, latent space features, musical attributes and training datasets. In doing this, we help researchers to understand the properties of state-of-the-art generative models and help to inform generative model research and design by providing a detailed analysis of current systems. We found that measureVAE has higher reconstruction accuracy and reconstruction efficiency than adversarialVAE but lower musical attribute independence (Section 5).

The results also show that measureVAE is capable of generating music across folk, pop, rock, jazz and blues, R&B and classical music styles, and performs best with lower complexity musical styles such as pop and rock. Furthermore, results show that measureVAE was able to generate music across these genres with interpretable musical dimensions of control.

The measureVAE generated output was found to have different musical interpretability scores for different datasets, but there was not a correlation between the musical features of datasets and the related interpretability scores of the generated music. For 4 regularised dimensions, the Irish folk dataset has the highest average interpretability scores for note density and note range, whereas Muse Bach has the highest for rhythmic complexity and average interval jump interpretability scores.

Interpretability metrics were in general higher when only two dimensions of the latent space were regularised. Similarly, loss and reconstruction accuracy scores were better for two regularised dimensions than four. These findings are to be expected as it is easier to achieve successful and linearly independent regularisation in fewer dimensions. For loss and reconstruction accuracy scores,

measureVAE performed better with the pair of note density and rhythmic complexity regularised dimensions than when it is trained with note range and average interval jump regularised dimensions. This may be because measureVAE is better at generating the tonal and rhythmic aspects of the music which are captured by ND and RC.

In terms of recommendations for use, results suggest that a 32 or 64 dimensional latent space would be optimal using measureVAE to generate music across a range of genres as this minimises latent space size whilst maximising reconstruction performance and providing similar interpretability scores to those offered by higher dimensional spaces. However, careful selection of latent space size is required for generation of specific genres of music. For example, Irish folk music may be optimally generated with 16 or even 8 latent dimensional space.

These results show that when explainable features are added to the measureVAE system, it performs well across genres of musical generation. For XAI and the arts more broadly, our approach demonstrates how complex AI models can be compared and explored in order to identify optimal configurations for the required styles of music generation. The work also demonstrates the complex relationships between datasets, explainable attributes, and AI model music generation performance. This complex relationship has some wider implications for generative AI models. For example, it highlights the bias built in to models which makes them more amenable to certain datasets rather than others – a key concern of human-centred AI[56]. In our case, the structure of measureVAE biased it towards lower complexity musical styles such as pop and rock at the expense of more complex forms of music such as Turkish makam, it is worth noting there are more marginalised forms of music.

The research presented here is a first step and is limited in scope. Future research needs to explore the effect that other genres and datasets, dataset sizes, musical attributes and training regimes have on the performance of explainable AI models. This would provide a more in-depth analysis of the landscape of generative AI models from which to inform future AI model research and design. For example, we chose two sets of musical attributes to use in this experiment based on frequently used attributes in research papers, but the utility of musical attributes to control musical generation very much depends on the context of use. We also need to compare a wider range of generative models and explainability techniques across datasets and musical attributes to identify which combinations of explainable AI model and dataset offer the best generative performance for the musical features of interest to musicians. For example, using information dynamic measures to compare generative models following [29]. It would also be important to examine longer-term music generation such as song structure generation, e.g., [57], and to use subjective listening tests to better understand the quality of the music generated (ibid.). Exploring how the robustness and interpretability of the models tested could be improved, for example, [58] would be especially important for real-time music generation settings such as live performance. Moreover, it would be useful to explore how the evaluation approach deployed in this paper could be applied to other domains such as image generation. For example, how comparative evaluations of image generating VAEs[37] could be undertaken to compare interpretable features as we have done in this paper, or to apply our approach to comparing the effect of different interpretable features on the robustness of image generation[58] instead of music generation.

Finally, we need to start to explore how the explainable features of the models tested in this paper could be used to make more interactive generative systems that move beyond being an empirical tool for researching AI models to become more of a creative tool used in musical practice and performance. As a first step, we will build the findings of this research into audio plugins which can be embedded into musician's musical tool chains as part of their artistic practice, starting with an MIDI music processor[59].

## Acknowledgements

## Author contributions

Bryan-Kinns instigated this work, led the research, supervised the student projects, and led the data analysis and writing. Zhao and Zhang contributed equally to the implementation of AI models, data collection and analysis in this work. Banar developed the original implementation in [19] which formed the basis for this work, contributed to the supervision of the student projects, contributed to the data analysis, and led the technical writing in this paper.

## Declarations of conflict of interest

The authors declared that they have no conflicts of interest to this work.

## Open Access

## References

[1]  B. L. Sturm, O. Ben-Tal, Ú. Monaghan, N. Collins, D. Herremans, E. Chew, G. Hadjeres, E. Deruty, F. Pachet. Machine learning research that matters for music creation: A case study. *Journal of New Music Research*, vol. 48, no. 1, pp. 36–55, 2019. DOI: 10.1080/09298215.2018.1515233.

[2]  D. Herremans, C. H. Chuan, E. Chew. A functional taxonomy of music generation systems. *ACM Computing Surveys*, vol. 50, no. 5, Article number 69, 2018. DOI: 10.1145/3108242.

[3]  F. Carnovalini, A. Rodà. Computational creativity and music generation systems: An introduction to the state of the art. *Frontiers in Artificial Intelligence*, vol. 3, Article number 14, 2020. DOI: 10.3389/frai.2020.00014.

[4]  P. M. Todd. A connectionist approach to algorithmic composition. *Computer Music Journal*, vol. 13, no. 4, pp. 27–43, 1989. DOI: 10.2307/3679551.

[5]  D. Eck, J. Schmidhuber. A First Look at Music Composition Using LSTM Recurrent Neural Networks, Technical Report No. IDSIA-07-02, Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, Manno, Switzerland, 2002.

[6]  J. P. Briot, G. Hadjeres, F. D. Pachet. Deep learning techniques for music generation-a survey, [Online], Available: https://arxiv.org/abs/1709.01620, 2017.

[7]  G. Hadjeres, F. Pachet, F. Nielsen. DeepBach: A steerable model for Bach chorales generation. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, vol. 70, pp. 1362–1371, 2017.

[8]  H. Y. Zhu, Q. Liu, N. J. Yuan, C. Qin, J. W. Li, K. Zhang, G. Zhou, F. R. Wei, Y. C. Xu, E. H. Chen. XiaoIce band: A melody and arrangement generation framework for pop music. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* London, UK, pp. 2837–2846, 2018. DOI: 10.1145/3219819.3220105.

[9]  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 *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, USA, 2019.

[10]  D. Gunning. Explainable Artificial Intelligence (XAI). DARPA/I2O Proposers Day, [Online], Available: https://www.darpa.mil/attachments/XAIIndustryDay_Final.pptx, 2016.

[11]  A. Pati, A. Lerch. Attribute-based regularization of latent spaces for variational auto-encoders. *Neural Computing and Applications*, vol. 33, no. 9, pp. 4429–4444, 2021. DOI: 10.1007/s00521-020-05270-2.

[12]  R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys*, vol. 51, no. 5, Article number 93, 2019. DOI: 10.1145/3236009.

[13]  G. Ciatto, M. I. Schumacher, A. Omicini, D. Calvaresi. Agent-based explanations in AI: Towards an abstract framework. In *Proceedings of the 2nd International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, Auckland, New Zealand, pp. 3–20, 2020. DOI: 10.1007/978-3-030-51924-7_1.

[14]  Q. V. Liao, D. Gruen, S. Miller. Questioning the AI: Informing design practices for explainable AI user experiences. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, Honolulu, USA, pp. 1–15, 2020. DOI: 10.1145/3313831.3376590.

[15]  M. T. Ribeiro, S. Singh, C. Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, USA, pp. 1135–1144, 2016. DOI: 10.1145/2939672.2939778.

[16]  G. Quellec, H. Al Hajj, M. Lamard, P. H. Conze, P. Massin, B. Cochener. ExplAIn: Explanatory artificial intelligence for diabetic retinopathy diagnosis. *Medical Image Analysis*, vol. 72, Article number 102118, 2021. DOI: 10.1016/j.media.2021.102118.

[17]  N. Du, J. Haspiel, Q. N. Zhang, D. Tilbury, A. K. Pradhan, X. J. Yang, L. P. Robert Jr. Look who′s talking now: Implications of AV′s explanations on driver′s trust, AV preference, anxiety and mental workload. *Transportation Research Part C*: Emerging Technologies, vol. 104, pp. 428–442, 2019. DOI: 10.1016/j.trc.2019.05.025.

[18]  Y. Shen, S. D. J. Jiang, Y. L. Chen, E. Yang, X. L. Jin, Y. L. Fan, K. Driggs-Campbell. To explain or not to explain: A study on the necessity of explanations for autonomous vehicles, [Online], Available: https://arxiv.org/abs/2006.11684, 2020.

[19]  N. Bryan-Kinns, B. Banar, C. Ford, C. N. Reed, Y. X. Zhang, S. Colton, J. Armitage. Exploring XAI for the arts: Explaining latent space in generative music. In *Proceedings of the 1st Workshop on eXplainable AI Approaches for Debugging and Diagnosis*, 2021.

[20]  G. Vigliensoni, L. McCallum, R. Fiebrink. Creating latent spaces for modern music genre rhythms using minimal training data. In *Proceedings of the 11th International Conference on Computational Creativity*, Coimbra, Portugal, pp. 259–262, 2020.

[21]  J. McCormack, T. Gifford, P. Hutchings, M. T. L. Rodriguez, M. Yee-King, M. d′Inverno. In a silent way: Communication between AI and improvising musicians beyond sound. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, Glasgow, UK, Article number 38, 2019. DOI: 10.1145/3290605.3300268.

[22]  P. Pasquier, A. Eigenfeldt, O. Bown, S. Dubnov. An introduction to musical metacreation. *Computers in Entertainment*, vol. 14, no. 2, Article number 2, 2016. DOI: 10.1145/2930672.

[23]  G. Widmer. Getting closer to the essence of music: The *Con espressione* manifesto. *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 2, Article number 19, 2017. DOI: 10.1145/2899004.

[24]  J. P. Briot, F. Pachet. Deep learning for music generation: Challenges and directions. *Neural Computing and Applications*, vol. 32, no. 4, pp. 981–993, 2020. DOI: 10.1007/s00521-018-3813-6.

[25] F. Colombo, A. Seeholzer, S. P. Muscinelli, J. Brea, W. Gerstner. Algorithmic composition of melodies with deep recurrent neural networks. In *Proceedings of the 1st Conference on Computer Simulation of Musical Creativity*, Huddersfield, UK, 2016. DOI: 10.13140/RG.2.1.2436.5683.

[26] B. L. Sturm, J. F. Santos, O. Ben-Tal, I. Korshunova. Music transcription modelling and composition using deep learning, [Online], Available: https://arxiv.org/abs/1604.08723, 2016.

[27] A. Pati, A. Lerch, G. Hadjeres. Learning to traverse latent spaces for musical score inpainting. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, pp. 343–351, 2019.

[28] A. Roberts, J. H. Engel, C. Raffel, C. Hawthorne, D. Eck. A hierarchical latent vector model for learning long-term structure in music. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, vol. 80, pp. 4361–4370, 2018.

[29] E. S. Koh, S. Dubnov, D. Wright. Rethinking recurrent latent variable model for music composition. In *Proceedings of the 20th International Workshop on Multimedia Signal Processing*, Vancouver, Canada, pp. 1–6, 2018. DOI: 10.1109/MMSP.2018.8547061.

[30] C. Ames. The Markov process as a compositional model: A survey and tutorial. *Leonardo*, vol. 22, no. 2, pp. 175–187, 1989. DOI: 10.2307/1575226.

[31] R. Whorley, R. Laney. Generating subjects for pieces in the style of Bach′s two-part inventions. In *Proceedings of the Joint Conference on AI Music Creativity*, Stockholm, Sweden, 2020.

[32] L. Kawai, P. Esling, T. Harada. Attributes-aware deep music transformation. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*, Montreal, Canada, pp. 670–677, 2020.

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. Generative adversarial networks. *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020. DOI: 10.1145/3422622.

[34] D. P. Kingma, M. Welling. Auto-encoding variational Bayes, [Online], Available: https://arxiv.org/abs/1312.6114, 2013.

[35] R. H. Yang, D. S. Wang, Z. Y. Wang, T. Y. Chen, J. Y. Jiang, G. Xia. Deep music analogy via latent representation disentanglement. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, pp. 596–603, 2019.

[36] Z. Y. Wang, Y. Y. Zhang, Y. X. Zhang, J. Y. Jiang, R. H. Yang, G. Xia, J. B. Zhao. PIANOTREE VAE: Structured representation learning for polyphonic music. In *Proceedings of the 21th International Society for Music Information Retrieval Conference*, Montreal, Canada, pp. 368–375, 2020.

[37] R. Q. Wei, C. Garcia, A. El-Sayed, V. Peterson, A. Mahmood. Variations in variational autoencoders – A comparative evaluation. *IEEE Access*, vol. 8, pp. 153651–153670, 2020. DOI: 10.1109/ACCESS.2020.3018151.

[38] R. Louie, A. Cohen, C. Z. A. Huang, M. Terry, C. J. Cai. Cococo: AI-steering tools for music novices co-creating with generative models. In *Proceedings of the Human-AI Co-creation with Generative Models and User-aware Conversational Agents Co-located, the 25th International Conference on Intelligent User Interfaces*, Cagliari, Italy, 2020.

[39] N. J. W. Thelle, P. Pasquier. Spire muse: A Virtual musical partner for creative brainstorming. In *Proceedings of the 21th International Conference on New Interfaces for Musical Expression*, Shanghai, China, 2021. DOI: 10.21428/92fbeb44.84c0b364.

[40] T. Murray-Browne, P. Tigas. Latent mappings: Generating open-ended expressive mappings using variational autoencoders. In *Proceedings of the 21th International Conference on New Interfaces for Musical Expression*, Shanghai, China, 2021. DOI: 10.21428/92fbeb44.9d4bcd4b.

[41] A. K. Gillette, T. H. Chang. ALGORITHMS: Assessing Latent Space Dimension by Delaunay Loss, Technical Report LLNL-CONF-814930, Lawrence Livermore National Laboratory, Livermore, USA, 2020.

[42] A. Pati, A. Lerch. Latent space regularization for explicit control of musical attributes. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, USA, 2019.

[43] G. Hadjeres, F. Nielsen, F. Pachet. GLSR-VAE: Geodesic latent space regularization for variational autoencoder architectures. In *Proceedings of the IEEE Symposium Series on Computational Intelligence*, Honolulu, USA, pp. 1–7, 2017. DOI: 10.1109/SSCI.2017.8280895.

[44] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, M. Ranzato. Fader networks: Manipulating images by sliding attributes. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, USA, pp. 5969–5978, 2017.

[45] H. H. Tan, D. Herremans. Music FaderNets: Controllable music generation based on high-level features via low-level feature modelling. In *Proceedings of the 21th International Society for Music Information Retrieval Conference*, Montreal, Canada, pp. 109–116, 2020.

[46] B. Banar, S. Colton. A systematic evaluation of GPT-2-based music generation. In *Proceedings of the 11th International Conference on Artificial Intelligence in Music, Sound, Art and Design*, Madrid, Spain, pp. 19–35, 2022. DOI: 10.1007/978-3-031-03789-4_2.

[47] M. K. Karaosmanoglu. A Turkish makam music symbolic database for music information retrieval: SymbTr. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, pp. 223–228, 2012.

[48] G. Dzhambazov, A. Srinivasamurthy, S. Şentürk, X. Serra. On the use of note onsets for improved lyrics-to-audio alignment in Turkish makam music. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*, New York, USA, pp. 716–722, 2016.

[49] C. Raffel. Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching, Ph. D. dissertation, Columbia University, USA, 2016.

[50] C. McKay, I. Fujinaga. jSymbolic: A feature extractor for MIDI files. In *Proceedings of the International Computer Music Conference*, New Orleans, USA, 2006.

[51] G. T. Toussaint. A mathematical analysis of African, Brazilian, and Cuban *clave* rhythms. In *Proceedings of the BRIDGES: Mathematical Connections in Art, Music, and Science*, Towson, USA, pp. 157–168, 2002.

[52] D. P. Kingma, J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, USA, 2015.

[53] L. Myers, M. J. Sirois. Spearman correlation coefficients, differences between. *Encyclopedia of Statistical Sciences*, S. Kotz, C. B. Read, N. Balakrishnan, B. Vidakovic, N. L. Johnson, Eds., Hoboken: John Wiley & Sons, Inc., 2006. DOI: 10.1002/0471667196.ess5050.pub2.

[54] D. P. Kingma, M. Welling. Auto-encoding variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations*, Banff, Canada, 2014.

[55] T. Adel, Z. Ghahramani, A. Weller. Discovering interpretable representations for both deep generative and discriminative models. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, pp. 50–59, 2018.

[56] O. O. Garibay, B. Winslow, S. Andolina, M. Antona, A. Bodenschatz, C. Coursaris, G. Falco, S. M. Fiore, I. Garibay, K. Grieman, J. C. Havens, M. Jirotka, H. Kacorri, W. Karwowski, J. Kider, J. Konstan, S. Koon, M. Lopez-Gonzalez, I. Maifeld-Carucci, S. McGregor, G. Salvendy, B. Shneiderman, C. Stephanidis, C. Strobel, C. Ten Holter, W. Xu. Six human-centered artificial intelligence grand challenges. *International Journal of Human–computer Interaction*, vol. 39, no. 3, pp. 391–437, 2023. DOI: 10.1080/10447318.2022.2153320.

[57] K. Chen, G. Xia, S. Dubnov. Continuous melody generation via disentangled short-term representations and structural conditions. In *Proceedings of the 14th International Conference on Semantic Computing*, San Diego, USA, pp. 128–135, 2020. DOI: 10.1109/ICSC.2020.00025.

[58] D. Y. Liu, L. Wu, H. F. Zhao, F. Boussaid, M. Bennamoun, X. H. Xie. Jacobian norm with selective input gradient regularization for improved and interpretable adversarial defense, [Online], Available: https://arxiv.org/abs/2207.13036, 2022.

[59] B. Banar, N. Bryan-Kinns, S. Colton. A tool for generating controllable variations of musical themes using variational autoencoders with latent space regularisation. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, Vancouver, Canada, 2023.

**Nick Bryan-Kinns** received the B. Sc. degree in computer science and the M.Sc degree in human-computer interaction from King′s College London, UK in 1993 and 1994, respectively, and the Ph. D. degree in human-computer interaction from Queen Mary and Westfield College, University of London, UK in 1998. He is a professor of creative computing at the Creative Computing Institute, University of the Arts London, UK. He is a Fellow of the Royal Society of Arts, Fellow of the British Computer Society, Turing Fellow at the Alan Turing Institute, Senior Member of the Association of Computing Machinery (ACM), and Chartered Engineer.

His research interests include explainable AI, interaction design, mutual engagement, interactive art and cross-cultural design.

E-mail: n.bryankinns@arts.ac.uk (Corresponding author)
ORCID iD: 0000-0002-1382-2914

**Bingyuan Zhang** received the B. Sc. degree in software engineering from Anhui Normal University, China in 2020. She is a master student in artificial intelligence from School of Electronic Engineering and Computer Science, Queen Mary University of London, UK.

Her research interests include interpretability of music generation models, anime faces generation models and machine learning.

E-mail: b.zhang@se21.qmul.ac.uk
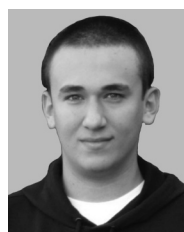ORCID iD: 0009-0005-7623-836X

**Songyan Zhao** received the B. Sc. degree in computer science and mathematics at Carleton College, USA in 2023. He is a master student in computer science at UCLA Samueli School of Engineering, USA.

His research interests include machine learning and AI music composition.

Email: zhaosongyan@g.ucla.edu
ORCID iD:0009-0000-6116-5325

**Berker Banar** received the B. Sc. degree in electrical and electronics engineering from Bilkent University, Turkey in 2016, and the M. Sc. degree in electronic production and design from Berklee College of Music, USA in 2019. He is currently a Ph. D. researcher (Comp. Sci.) at the AI and Music CDT, Queen Mary University of London, UK, and an enrichment student at the Alan Turing Institute, UK. He has previously done research internships at Sony, Bose and Northwestern University.

His research interests include machine learning, deep learning, optimisation, generative modelling, music generation and computational creativity.

E-mail: b.banar@qmul.ac.uk
ORCID iD: 0000-0003-1808-2203