SEMANTIC CONTROL OF GENERATIVE MUSICAL ATTRIBUTES

Stewart Greenhill **Majid Abdolshah** **Vuong Le**
Sunil Gupta **Svetha Venkatesh**
Applied Artificial Intelligence Institute, Deakin University, Australia
s.greenhill@deakin.edu.au

ABSTRACT

Deep generative neural networks have been successful in tasks such as composing novel music and rendering expressive performance. Controllability is essential for building creative tools from such models. Recent work in this area has focused on disentangled latent space representations, but this is only part of the solution. Efficient control of semantic attributes must handle non-linearities and holes that occur in latent spaces, whilst minimising unwanted changes to other attributes. This paper introduces SeNT-Gen, a neural traversal algorithm that uses a secondary neural network to model the complex relationships between latent codes and musical attributes. This enables precise editing of semantic attributes that adapts to context. We demonstrate the method using the dMelodies dataset, and show strong performance for several VAE models.

1. INTRODUCTION

Deep generative models show promise for music, with systems that can compose melodies and accompaniments, render expressive performances, or synthesise instrumental sounds and singing voices [1, 2]. While many of these work as "black boxes," *controllability* is an essential factor in creative tools for musicians. This raises important challenges such as controlling musically meaningful aspects of the composition, balancing creativity versus imitation, providing interactivity and refinement, and producing a convincing temporal structure that has a sense of direction [3].

In generative applications a neural network is trained to find a low-dimensional representation for complex data. A common approach uses a Variational Auto-Encoder (VAE) in which an *encoder* learns to transform the data space X into a simpler "latent space" Z. For music X is a representation of a score such as a piano-roll that defines the pitch and duration of notes over time. A *decoder* recovers the original representation from the latent samples. In learning the latent representation the network identifies the essential attributes needed to construct the melody. The dimensions of the latent space represent semantic attributes such as note density [4], syncopation [5], genre [6] or arousal [7]. Novel melodies can be generated by decoding samples drawn from the latent space. Moving in the latent space adjusts the semantic attributes, for example: to smoothly "morph" between two melodies by interpolating a path between their latent positions.

Recent work on control of generative music has focused on regularisation of the latent space, and disentanglement of the semantic attributes. The aim is to relate each semantic attribute of the melody to a unique dimension of the latent space, which allows attributes to be adjusted by adding an "attribute vector" [8, 9] in latent space. This approach involves some challenges. First, there is often a tradeoff between regularisation of the latent space and reconstruction quality. Second, effective disentanglement can only be achieved through supervision [10] and unsupervised approaches are sensitive to inductive biases in both the data set and learning model (such as network model, hyperparameters, random seeds). Third, the relationship between the latent dimension and corresponding semantic attribute value may be non-linear. Fourth, latent spaces may contain "holes" where the decoder produces invalid results [11] and these must be avoided when adjusting attributes, or interpolating paths in the latent space. This means that some additional work is required beyond disentanglement to properly control the values of semantic attributes.

This paper proposes a new method to efficiently traverse latent spaces frequently used in generative music. We introduce the Semantic Neural Latent Traversal method for Generative models (SeNT-Gen), which employs a secondary neural network to model the complex relationship between latent codes and semantic attributes. The SeNT-Gen traversal function predicts the new latent position given a starting position and attribute change, supporting non-linear relationships and adapting to the context of the traversal. We evaluate the performance of SeNT-Gen for musical control using the dMelodies [12] data set.

Our key contributions are:

1. A neural method to traverse the latent space, targeting precise changes to musical attributes, and supporting non-linear contextual relationships between the VAE latent space and semantic attributes;

2. Experiments and analysis of the proposed method using the established dMelodies data set. The method is independent of learning model, and shows best performance for strongly regularised models;

3. Objective quantitative metrics to measure performance of the method through targeted attribute changes, which improves on notions of "controllability" that are tied to interpolation heuristics.

2. RELATED WORK

Control of generative music is usually based on latent space representations. Some of these approaches are surveyed here, while alternative approaches for images do not directly translate between image and musical domains [12]. Music-VAE [5] uses a hierarchical recurrent VAE to model temporal structure in music. Attribute vectors are shown to partially control attributes such as *note density*, *melodic interval* and *rhythmic syncopation*. MIDI-VAE [6] learns songs using a a parallel VAE with shared latent space, and an attached style classifier enables a chosen *style* (classic, jazz, pop, Bach, Mozart) to be applied to the output. GLSR-VAE [4] uses a novel regularisation method to control *note density* of chorale-style melodies. Music FaderNets [7] proposes a Gaussian mixture VAE for learning piano performance, with control over the *arousal* or "energy level" based on rhythm and note density. AR-VAE [13] defines a supervised regularisation method which is applied to controlling *rhythmic complexity*, *pitch range*, *note density*, and *contour* for melodies trained from chorales and folk tunes. Kawai and colleagues [14] use a VAE with adversarial classifier-discriminator to condition the decoder on semantic attributes. This model is trained on folk tunes to control orthogonal attributes such as *number of notes*, *pitch variability*, *chromatic motion* and *amount of arpeggiation*.

Apart from latent space approaches, other methods generally require ad-hoc modifications to the probability distribution of a model. Transformers are used to generate music with long-term structure [15], but control is limited to providing a prompt stimulus for the system to extend. Pop Music Transformer [16] uses a Transformer-XL model to learn expressive piano performances. Control over *tempo* and *chord* is achieved by masking out corresponding event probabilities in the model output. Coconet [17] uses a convolutional model to generate chorales in the style of Bach. Cococo [18] adds "semantic sliders" for *conventional/surprising* and *happy/sad* output using "soft priors" to modify the model's sampling distribution.

Several factors make it difficult to compare the results of these studies. Firstly, there are a wide variety of model architectures, and each implements its own encoding of the training data into a form suitable for model training. This in turn depends on the data-sets used, which vary from simple scores in ABC notation, to traditional scores, to MIDI transcriptions of actual performances either recorded from a digital keyboard or automatically extracted from audio recordings. Secondly, while there is a focus on disentangled representations, these are not in themselves control methods. Even in a disentangled latent space, more work is required to accurately achieve a specific value of an attribute, dealing with potential non-linearities and holes in the latent space. In the absence of a particular control algorithm some works assess "controllability" through interpolation [7, 11, 14] though there is no consistency in the metrics used.

Feature	Values	Description
Tonic	12	Notes C, C#, D ..., B
Octave	3	Octave 4, 5, or 6
Scale	3	major, minor, or blues
Rhythm Bar 1, 2	28	$\binom{8}{6}$ codes for 6 note onsets
Arp Chord 1, 2, 3, 4	2	arpeggio direction up or down

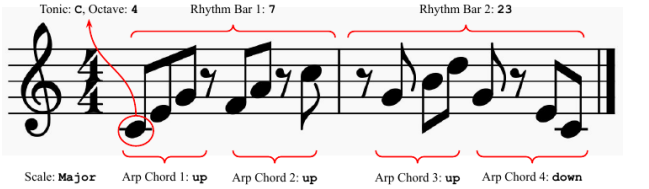

Figure 1. Each dMelodies two-bar melody is described by 9 attributes. *Tonic*, *Octave* and *Scale* apply to the entire melody. Separate *Rhythm* attributes apply to each bar. *Arpeggio* attributes apply to each of the 4 chords. The table (top) shows the number of values for each attributes, and an example from [12] is shown below.

The dMelodies dataset [12] has been proposed for evaluation of musical disentanglement learning, similar to dSprites in the image domain [19]. dMelodies includes 1,354,752 unique two-bar melodies, algorithmically constructed with independent factors of variation, with each comprised of 4 arpeggios in a I-IV-V-I chord pattern (see Table 1). Attributes are discrete, and most (except for *Tonic* and *Octave*) are categorical. dMelodies provides three unsupervised reference models including β-VAE [20], and a subsequent paper [11] adds three supervised models: I-VAE [21], AR-VAE [13], and S2-VAE [22]. Three measures of disentanglement are implemented [12]: *Mutual Information Gap* [23], *Modularity* [24], and *Separated Attribute Predictability* [25]. On these measures, supervised learning is shown to give better disentanglement, without sacrificing reconstruction quality [11].

2.1 Metrics

Suppose the VAE decoder $\mathcal{G}$ generates an object $x = \mathcal{G}(z)$ for latent sample z, and let $c = [c_1, c_2, ..., c_K]$ be the K semantic attributes of x. Disentanglement is formalised through the concepts of *consistency* and *restrictiveness* [26] and establishes a relationship between a latent dimension z_i and a corresponding semantic attribute c_i. *Consistency* says that when z_i is fixed, c_i of the generated x never changes. *Restrictiveness* says that when only z_i is changed, the change is restricted to c_i and there is no change to other attributes c_j for $j \neq i$.

Attributes may be controlled through a traversal function $\mathcal{T}_k(z, c_k, c_k^*)$ which predicts the latent value z^* required to change attribute k of z from c_k to c_k^*. The accuracy of $\mathcal{T}_k$ can be evaluated for a set of changes (z, c_k, c_k^*) by measuring the deviation of the result from the target, and any side effects on non-target attributes. Alternative approaches assess "controllabilty" in the absence of such a traversal function. Instead, interpolation is used to tra-

VAE	To	Oc	Sc	R1	R2	A1	A2	A3	A4
β	1	.95	.14	.64	.64	.28	.23	.28	.24
AR	1	.95	.34	.09	.09	.02	.02	.02	.02
I	1	.99	.36	.73	.77	.50	.52	.54	.53
S2	1	.94	.12	.65	.70	.29	.23	.34	.25

Table 1. Latent Density Ratio (LDR) for four dMelodies VAE models. Attributes are Tonic (To), Octave (Oc), Scale (Sc), Rhythm Bar (R) and Arp Chord (A).

verse from a position z in the latent space, along the corresponding dimension z_d between a minimum and maximum value over a number of steps. This is repeated over a small batch of points. Three measures are defined by [7] based on the work of [26]: *Consistency* measures how constant the attribute is across the batch for the same value of z_d, *Restrictiveness* measures how constant other attributes are when z_d changes, and *Linearity* measures how linear the attribute is with respect to the latent dimension z_d. An *attribute change matrix* $A(d, n)$ is proposed by [11] which computes the net change in the n^{th} attribute while traversing regularised dimension d. When an attribute is well controlled A should be high on the diagonal, and low elsewhere, and these relationships can be inspected by visualising the matrix. Another measure of controllability is the correlation between the interpolated value z_d and the resulting attribute [14], which measures linearity but not restrictiveness.

Another important factor is the *Latent Density Ratio* (or *LDR*), the proportion of a batch of random latent samples that decode with valid attributes [11]. As shown in Table 1, errors in dMelodies occur most frequently with the *Scale*, *Rhythm Bar* and *Arp Chord* attributes. If notes are generated that are outside the three defined scales (major, minor, blues) the *Scale* attribute is invalid. If a bar does not have exactly 6 note onsets the *Rhythm Bar* attribute does not match one of the 28 codes, and is invalid. If the chord notes are not consistently ascending or descending, the *Arp Chord* attribute is invalid. A traversal function should aim to be more accurate than these base-line LDR rates.

3. METHOD

3.1 Neural Latent Traversal

This section describes SeNT-Gen as applied to musical VAE models. Further details are available for a GAN-based image synthesis application [27].

Let x be a sample of the data space $X = \mathbb{R}^N$, a piano-roll encoding of a melody. Let z be a sample of the latent space $Z = \mathbb{R}^D$ of the VAE model. Let $c = [c_1, c_2, ..., c_K]$ be the vector of K semantic attributes of x, and $R_1(.)$, ..., $R_k(.)$ be K functions that compute the value of attribute k of x in the normalised range $[0, 1]$: $c_k = R_k(x)$. Let $\mathcal{F}(.)$ be the encoder, and $\mathcal{G}(.)$ be the decoder of the VAE, so that $\mathcal{G}(\mathcal{F}(x)) \approx x$. We use x^* for the *target value* of x, and $\hat{x}$ for its *predicted* or *achieved* value.

Given a sample x, we aim to find the modified x^* such that the value of attribute k is changed from c_k to c_k^* while all other attributes remain unchanged: $c_i^* = c_i, i \neq k$.

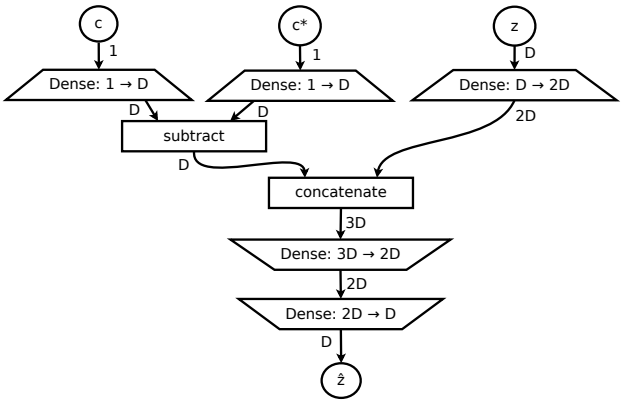

Figure 2. Implementation of SeNT-Gen neural traversal function $\mathcal{T}_k(z, c_k, c_k^*)$. Input is the latent code z, and the desired attribute change from c_k to c_k^*. Each Dense layer includes ReLU activation, except for the Tanh activation on the final layer. Output is the latent code $\hat{z}$ which approximates the solution z^*.

This adjustment will be done in the latent space. Given the latent encoding $z = \mathcal{F}(x)$, we seek a modified z^* that generates $x^* = \mathcal{G}(z^*)$, with the desired attribute value $c_k^* = R_k(\mathcal{G}(z^*))$.

SeNT-Gen implements contextual traversal of the latent space for each of the attributes. The traversal function $\mathcal{T}_k$ predicts the new value $\hat{z}$ given the old value z, and the old and new values of the attribute c_k and c_k^*:

$$\hat{z} = \mathcal{T}_k(z, c_k, c_k^*)$$

The traversal function $\mathcal{T}_k$ is implemented by training the neural network shown in Figure 2 which outputs the value $\hat{z}$, an approximate solution for z^*.

Three constraints are imposed during training. The first constraint is to minimise the perceptual loss by ensuring that the changed attribute value $\hat{c}_k = R_k(\mathcal{G}(\hat{z}))$ is close to the target c_k^*:

$$\mathcal{L}_c = \mathbb{E}_{z \sim P(Z)} \left[\| R_k(\mathcal{G}(\hat{z})) - c_k^* \|_2^2 \right] \quad (1)$$

The term $\| R_k(\mathcal{G}(\hat{z})) - c_k^* \|_2$ represents the deviation between $\hat{c}_k$ and the target c_k^*, and is in the range [0,1]. When R_k is invalid for $\mathcal{G}(\hat{z})$ a deviation of 1 is used instead.

The other two constraints are required to align the attributes of the traversal function with the relevant dimensions of the latent space. In a disentangled latent space each semantic attribute corresponds to one latent dimension, but in general there will likely be a small number of relevant dimensions that are strongly related to each attribute. For *relevant* dimensions r, $\hat{z}_r$ should be be close to the correct solution z_r^*. For the other *irrelevant* dimensions i, $\hat{z}_i$ should be close to the original location z_i.

Relevance is expressed using the vector $\rho_k \in \mathbb{R}^D$, which is 1 for relevant dimensions and 0 otherwise. Under supervised training, this relation will be known and for a consistent numbering of attributes and dimensions $\rho_k[i] = 1$ for $i = k$, and 0 for $i \neq k$. Otherwise, for unsupervised models ρ_k can be calculated using mutual

information as described below. The two remaining loss functions are thus:

$$\mathcal{L}_I = \mathbb{E}_{z,z^* \sim P(Z)} \left[\| {\rho_k}^T (\hat{z} - z^*) \|_2^2 \right] \quad (2)$$

$$\mathcal{L}_{\neg I} = \mathbb{E}_{z,z^* \sim P(Z)} \left[\| (1 - \rho_k)^T (\hat{z} - z) \|_2^2 \right] \quad (3)$$

Equation 2 pulls $\hat{z}$ towards z^* for relevant dimensions, and Equation 3 pulls $\hat{z}$ towards z for irrelevant dimensions.

When ρ_k is not known a-priori it can be calculated using the mutual information (MI) between the dimensions $d \in \{1, ..., D\}$ of z and the c_k values:

$$\rho_k = \text{threshold}\left(\text{softmax}(\text{MI}(z_d, c_k)), \gamma\right) \quad (4)$$

Where γ is a hyper-parameter controlling the number of latent dimensions related to each semantic attribute. Equation 4 selects γ dimensions of z that have the most mutual information with each attribute c_k.

3.2 Latent Traversal Training

The traversal function $\mathcal{T}_k$ is trained on similar pairs of latent vectors (z, z^*) which differ only in attribute c_k. Such pairs are good examples of attribute modifications. We sample pairs of the data space (x, x^*), and compute their attributes (c_k, c_k^*), where $c_k = R_k(x)$, and $c_k^* = R_k(x^*)$. We use the encoder compute their latent representation (z, z^*), where $z = \mathcal{F}(x)$ and $z^* = \mathcal{F}(x^*)$. We discard pairs where there is a significant difference in attributes other than k. A training pair is considered valid if:

$$\sum_{j \neq k} \| c_j - c_j^* \|_2 \leq \epsilon, \quad \text{for } j \in \{1, ..., K\} \quad (5)$$

Where $\epsilon \geq 0$ is a slack parameter. One implementation is to enumerate all pairs from a set of candidate samples, and adjust ϵ to be as small as possible while also yielding enough similar pairs.

After generating the training pairs, we train the traversal model $\mathcal{T}_k$ by minimising the loss:

$$\mathcal{L}_k \triangleq \lambda_1 \mathcal{L}_I + \lambda_2 \mathcal{L}_{\neg I} + \lambda_3 \mathcal{L}_c \quad (6)$$

where λ_1, λ_2, and λ_3 are hyper-parameters that balance perceptual loss versus semantic relevance.

4. EXPERIMENTS

We train each of the four dMelodies learning models (β-VAE, AR-VAE, I-VAE and S2-VAE) for three different hyperparameter settings and 3 different random seeds, a total of 36 models. Input is a two-bar melody, with a latent space of $D = 32$ dimensions. For β-VAE we vary $\beta \in \{0.2, 1, 4\}$, and for the other models we vary the regularisation strength $\Gamma \in \{0.1, 1, 10\}$ for $\beta = 0.2$. The default settings use 1016k melodies (75%) for training the original models, which leaves 338k melodies (25%) as candidates for training and testing the SeNT traversal function $\mathcal{T}_k$. Setting $\epsilon = 0$ yields between 42k and 157k pairs (see Equation 5) and from these we allocate 80% for training and 20% for testing. Since dMelodies is generated by combinatorial expansion, attributes with the most states (*Tonic*, *Rhythm Bar*) have the most pairs, while those with the fewest states (*Arp Chord*) have the least. Latent codes z are normalised to the range $[-1, 1]$, and attributes c_k to $[0, 1]$. Neural traversal is trained for $\gamma = 3$ (see Equation 4), $\lambda_1 = \lambda_2 = \lambda_3 = 1$ (Equation 6).

4.1 Performance Metrics

Performance of the traversal function $\mathcal{T}_k$ is evaluated by measuring accuracy on a set of attribute changes. A set of testing pairs (z, z^*) is generated using the same method that generated the training pairs (see Equation 5). These define a starting state z and an attribute change from c_k to c_k^* that leaves other attributes unchanged. Using the traversal function we predict the new latent code $\hat{z} = \mathcal{T}_k(z, c_k, c_k^*)$, and compute its attributes $\hat{c}_i = R_i(\mathcal{G}(\hat{z}))$ for all i. This approach ensures that only feasible attribute changes are tested, and that these have been unseen during training.

For a targeted change of an attribute, the important measures are: (1) How far is the edited attribute value $\hat{c}_k$ from the desired target c_k^*? and (2) How far are the unedited attributes $\hat{c}_u$ from the original values c_u? The *target deviation* Δ_k for target attribute k is defined as:

$$\Delta_k = |\hat{c}_k - c_k^*| \quad (7)$$

where $\hat{c}_k = R_k(\mathcal{G}(\hat{z}))$. The *non-target deviation* ∇_u for non-target attribute u is defined as:

$$\nabla_u = |\hat{c}_u - c_u| \quad (8)$$

When Δ_k is low, the traversal achieves the desired attribute k value. When ∇_u is low, the traversal avoids unintended changes to other attributes u. When combined, these measures are similar to the attribute change matrix of [11], except that Δ_k is with respect to specific target values, rather than arbitrary interpolation points.

Occasionally errors occur in the generated melodies, due to the decoder quality and the proximity of samples to holes in the latent space. Ideally $\mathcal{T}_k$ should avoid these holes, and to measure this we define the *Target Density Ratio* (or TDR) to be the proportion of the decoded target $\hat{z}$ values with valid attributes. We aim for TDR to be substantially higher than the underlying LDR (defined in 2.1).

5. RESULTS

For brevity we summarise the 36 models by selecting the best performing hyper-parameters, and aggregate over the three random seeds. For supervised models these are the settings with the most regularisation $\Gamma = 10$, and for β-VAE, the median $\beta = 1$. Source code is available online with further results and technical details. [1]

Figure 3 shows Mutual Information between semantic attributes (vertical axis) and latent dimensions (horizontal axis). The supervised S2-VAE (top) shows good disentanglement, with each attribute uniquely related to one

[1] https://github.com/stewartgreenhill/sentgen

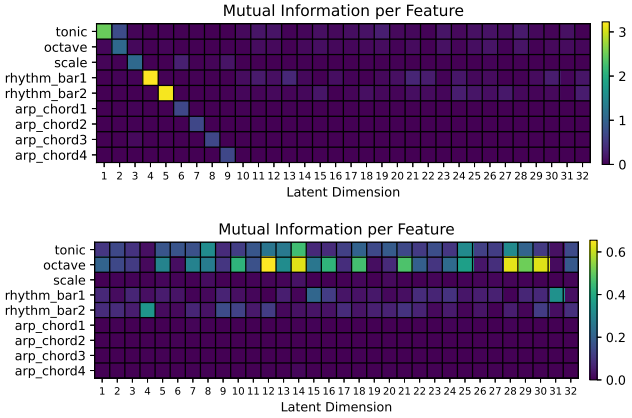

Figure 3. Mutual Information between semantic attributes and dimensions of the latent space for supervised S2-VAE (top), and unsupervised β-VAE (bottom).

of the first 9 latent dimensions. The strongest relationships are for the attributes with the most states: *Tonic* and *Rhythm Bar*. The unsupervised β-VAE (bottom) shows some strong but less well separated relationships, and some very weak relationships: *Scale*, *Arp Chord*. Mutual Information is important in SeNT-Gen for determining ρ_k which aligns semantic attributes to the latent space (Equations 2–4).

Figure 4 shows SeNT-Gen traversal accuracy for the S2-VAE. The bottom chart shows mean target deviation Δ_k for target attributes k. Smaller deviations are better. For most attributes the deviation is 2% or less, and the worst performance is 8% for attribute *Tonic*. The top chart shows the mean non-target deviation ∇_k, with non-target attributes on the vertical axis, and target attributes on the horizontal axis. The *Scale* attribute is most influenced by changes to other attributes, particularly *Tonic*, and *Octave* where the influence approaches 60%. This makes sense since these attributes are both changing the overall pitch of the melody, and it only requires one transposition "error" amongst the 12 melody notes to cause the scale to be invalid or altered. To a lesser extent the *Arp Chord* attributes are also susceptible for the same reason. Changes to *Arp Chord* attributes are the most accurate, with no deviation in the target or non-target attributes other than *Scale*.

Figure 5 shows SeNT-Gen traversal accuracy for the β-VAE which is the worst performing model. Rhythm attributes show a good target deviation of 3%, with the pitch based attribute deviations ranging from 7 to 22%. Non-target errors are lowest for changes to *Rhythm Bar 1 & 2*, but are generally much higher than for the S2-VAE. Here *Scale* and *Arp Chord* attributes are most influenced by changes to other attributes. This is expected since these attributes are only weakly related to dimensions of the latent space (see Figure 3) so are essentially invisible to the traversal constraints.

Another way to evaluate traversal accuracy is to look at correlation R^2 between target and achieved attribute values. Figure 6 shows the target c_k^* value (horizontal) versus the achieved $\hat{c}_k$ (vertical) for rhythm_bar1, the attribute

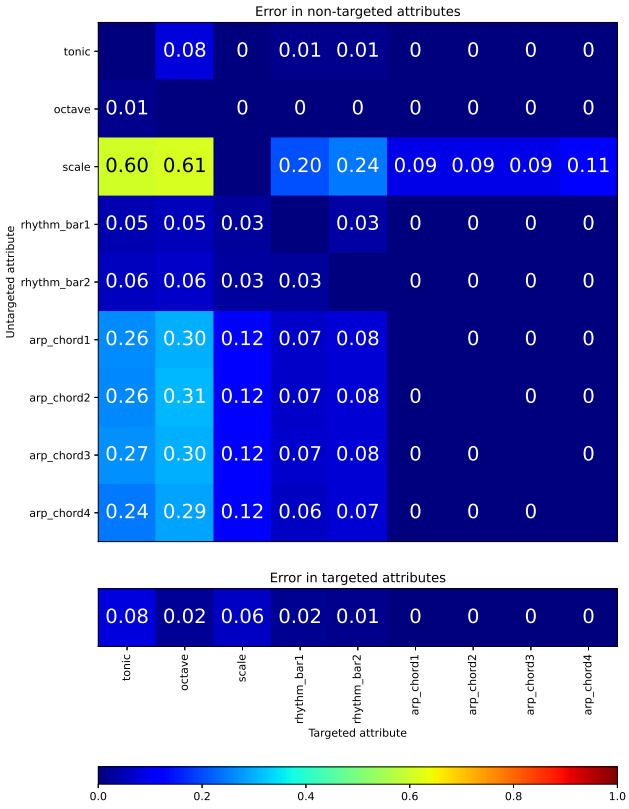

Figure 4. Accuracy of S2-VAE traversal, showing *target deviation* Δ (bottom), and *non-target deviation* ∇ (top). Target attributes are on the horizontal axis, and non-targets on the vertical.

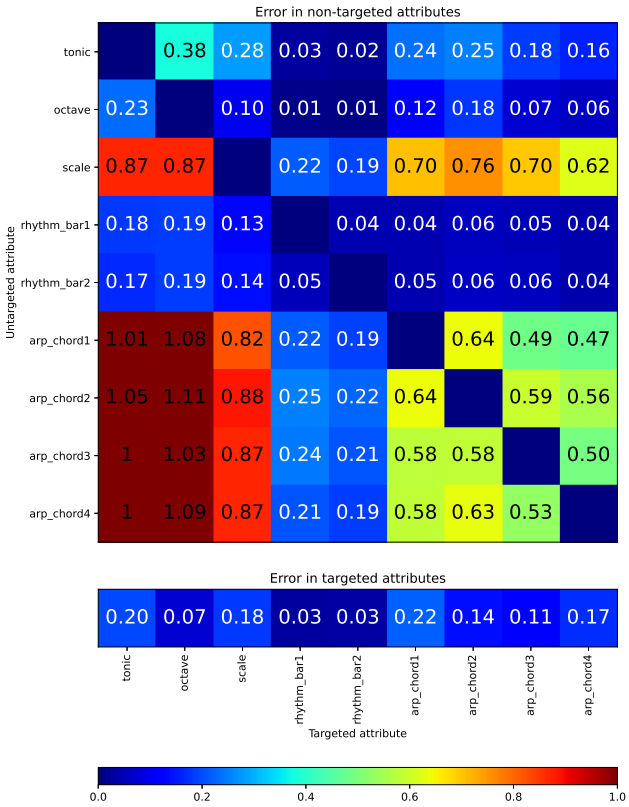

Figure 5. Accuracy of β-VAE traversal, showing *target deviation* Δ (bottom), and *non-target deviation* ∇ (top). Target attributes are on the horizontal axis, and non-targets on the vertical.

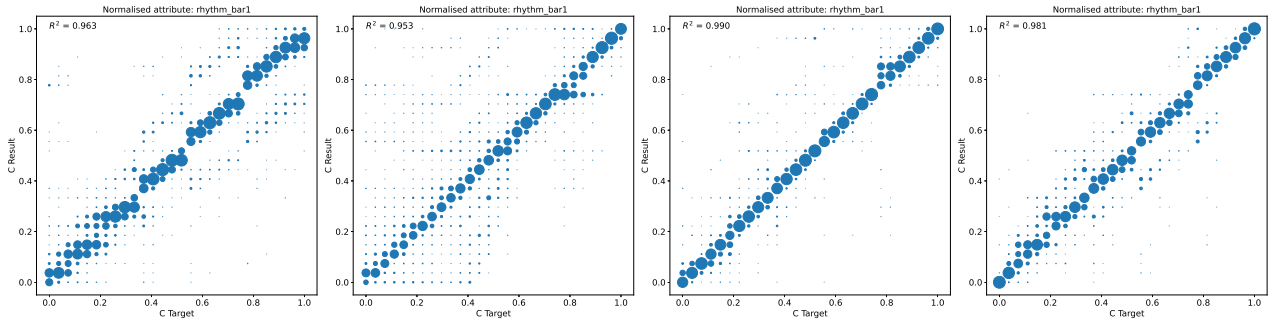

Figure 6. Correlation R^2 between normalised target c_k^* (horizontal axis) and result $\hat{c}_k$ (vertical axis) for $k = \text{rhythm\_bar1}$. Dot area is proportional to number of samples. Models are (left to right) β-VAE, AR-VAE, I-VAE, and S2-VAE.

VAE	To	Oc	Sc	R1	R2	A1	A2	A3	A4
β	.09	.78	.19	.96	.97	.11	.45	.56	.32
AR	.07	0	-1.3	.95	.97	.98	.99	1	1
I	.26	.34	.98	.99	.97	1	1	1	1
S2	.77	.93	.78	.98	.99	1	1	1	1

Table 2. Correlation R^2 between $\hat{c}_k$ versus c_k^* for four dMelodies VAE models. Attributes are Tonic (To), Octave (Oc), Scale (Sc), Rhythm Bar (R) and Arp Chord (A).

VAE	To	Oc	Sc	R1	R2	A1	A2	A3	A4
β	1	1	.30	.92	.92	.57	.52	.64	.62
AR	1	1	.49	.76	.82	.72	.73	.73	.73
I	1	1	.92	.95	.91	.90	.91	.90	.92
S2	1	.99	.65	.93	.93	.91	.90	.91	.91

Table 3. Target Density Ratio (TDR) for four dMelodies VAE models. See Table 1 for Latent Density Ratio (LDR).

with the most states. The frequency and location of errors can be visualised by inspecting the off-diagonal elements. Table 2 shows the R^2 values for each of the attributes over four dMelodies VAE models, excluding samples which yield invalid results. As expected, this measure shows high correlations where target deviation Δ is low. The three supervised models show strong control over the *Rhythm Bar* and *Arp Chord* attributes with weaker control over *Tonic*, *Octave* and *Scale*.

Table 3 shows the Target Density Ratio (TDR), the proportion of the decoded targets $\hat{x} = \mathcal{G}(\hat{z})$ that have valid attributes, thus avoiding potential holes in the latent space. The best performer from this perspective is the I-VAE which exhibits good TDR for all attributes. All models score better than the corresponding latent density ratio (LDR, Table 1). Notably, the AR-VAE shows good control over *Arp Chord* with TDR=0.73 despite scoring very low LDR=0.02 for these attributes.

6. CONCLUSION

This paper presents a novel algorithm to control musical attributes of deep generative models. The SeNT-Gen method implements a neural traversal function $\mathcal{T}_k(z, c_k, c_k^*)$ that predicts the latent position z^* required to change attribute k of z from c_k to c_k^*. Previous works in this field focus on disentanglement but do not implement traversal functions and instead assess controllability via latent space interpolation which does not allow the specification of a particular target value c_k^* for the controlled attribute.

The SeNT-Gen method is demonstrated using the dMelodies data set and various VAE models. Performance is strongest for highly regularised models. The best performance was obtained using the S2-VAE model, which shows strong control over most attributes, and few side-effects except for *Scale*. Some attributes that depend on note pitch (*Scale* and *Arp Chord*) are significantly less stable when adjusting overall pitch via *Tonic* or *Octave*, and also show low LDR scores. The definition of these attributes may be fragile to small changes in the melody notes. The Target Density Ratio (TDR) measures the robustness of the method to holes in the latent space, aggregating factors related to the latent space regularisation, the traversal algorithm, and the fidelity of the decoder. TDR for I-VAE and S2-VAE is good for most attributes, and AR-VAE shows very strong improvement over a poor baseline LDR.

Although each SeNT-Gen traversal function controls only one attribute, more complex operations involving multiple attributes could be performed using a sequence of separate attribute changes. Future improvements to the algorithm could include better support for categorical variables, as well as mixtures different variable types. Categorical variables would normally be one-hot encoded, but alternative distance metrics might be required, for example Jaccard distance where Euclidean distance is currently used (Equations 1 and 5).

While demonstrated here using VAE models, SeNT-Gen can also be used in other latent space models such as Generative Adversarial Networks (GAN). There is an underlying assumption that the relationship between semantic attributes and the latent dimensions will be fairly sparse. This is normally true for supervised models, but can also apply in other models too. In any case the hyper-parameter γ should be chosen to reflect this. An extensive study of the other hyper-parameters λ_1, λ_2, and λ_3 was outside the scope of this work, but fine-tuning these values through meta-optimisation may improve the overall performance.

7. REFERENCES

[1] S. Ji, J. Luo, and X. Yang, "A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions," *arXiv preprint arXiv:2011.06801*, 2020.

[2] G. Peeters and G. Richard, "Deep learning for audio and music," in *Multi-faceted Deep Learning*. Springer, 2021, pp. 231–266.

[3] J.-P. Briot and F. Pachet, "Music generation by deep learning – challenges and directions," *arXiv preprint arXiv:1712.04371*, 2017.

[4] G. Hadjeres, F. Nielsen, and F. Pachet, "GLSR-VAE: Geodesic latent space regularization for variational autoencoder architectures," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–7.

[5] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *International conference on machine learning*. PMLR, 2018, pp. 4364–4373.

[6] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, "MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer," in *Proceedings 19th International Society for Music Information Retrieval Conference*, 2018.

[7] H. H. Tan and D. Herremans, "Music FaderNets: Controllable music generation based on high-level features via low-level feature modelling," in *Proc. of the International Society for Music Information Retrieval Conference*, 2020.

[8] T. White, "Sampling generative networks," *arXiv preprint arXiv:1609.04468*, 2016.

[9] D. T. Chang, "Latent variable modeling for generative concept representations and deep generative models," *arXiv preprint arXiv:1812.11856*, 2018.

[10] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4114–4124.

[11] A. Pati and A. Lerch, "Is disentanglement enough? on latent representations for controllable music generation," in *Proceedings 22nd International Society for Music Information Retrieval Conference*, 2021.

[12] A. Pati, S. Gururani, and A. Lerch, "dMelodies: A music dataset for disentanglement learning," in *Proceedings 21st International Society for Music Information Retrieval Conference*, 2020.

[13] A. Pati and A. Lerch, "Attribute-based regularization of latent spaces for variational auto-encoders," *Neural Computing and Applications*, vol. 33, no. 9, pp. 4429–4444, 2021.

[14] L. Kawai, P. Esling, and T. Harada, "Attributes-aware deep music transformation," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020.

[15] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," *arXiv preprint arXiv:1809.04281*, 2018.

[16] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1180–1188.

[17] C.-Z. A. Huang, T. Cooijmans, A. Roberts, A. Courville, and D. Eck, "Counterpoint by convolution," in *Proceedings 18th International Society for Music Information Retrieval Conference*, 2017.

[18] R. Louie, A. Cohen, C.-Z. A. Huang, M. Terry, and C. J. Cai, "Cococo: AI-steering tools for music novices co-creating with generative models." in *Proceedings 25th International Conference on Intelligent User Interfaces*, 2020.

[19] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner, "dSprites: Disentanglement testing sprites dataset," 2017. [Online]. Available: https://github.com/deepmind/dsprites-dataset/

[20] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[21] T. Adel, Z. Ghahramani, and A. Weller, "Discovering interpretable representations for both deep generative and discriminative models," in *International Conference on Machine Learning*. PMLR, 2018, pp. 50–59.

[22] F. Locatello, M. Tschannen, S. Bauer, G. Rätsch, B. Schölkopf, and O. Bachem, "Disentangling factors of variation using few labels," in *Proceedings 8th International Conference on Learning Representations*, 2020.

[23] R. T. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[24] K. Ridgeway and M. C. Mozer, "Learning deep disentangled embeddings with the f-statistic loss," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[25] A. Kumar, P. Sattigeri, and A. Balakrishnan, "Variational inference of disentangled latent concepts from unlabeled observations," *arXiv preprint arXiv:1711.00848*, 2017.

[26] R. Shu, Y. Chen, A. Kumar, S. Ermon, and B. Poole, "Weakly supervised disentanglement with guarantees," in *International Conference on Learning Representations*, 2019.

[27] M. Abdolshah, H. Le, T. K. George, V. Le, S. Gupta, S. Rana, and S. Venkatesh, "Neural latent traversal with semantic constraints," 2022. [Online]. Available: https://openreview.net/forum?id=ODdaICh-7dK