Attribute-based Regularization of Latent Spaces for Variational Auto-Encoders

Ashis Pati · **Alexander Lerch**

Abstract Selective manipulation of data attributes using deep generative models is an active area of research. In this paper, we present a novel method to structure the latent space of a Variational Auto-Encoder (VAE) to encode different continuous-valued attributes explicitly. This is accomplished by using an attribute regularization loss which enforces a monotonic relationship between the attribute values and the latent code of the dimension along which the attribute is to be encoded. Consequently, post training, the model can be used to manipulate the attribute by simply changing the latent code of the corresponding regularized dimension. The results obtained from several quantitative and qualitative experiments show that the proposed method leads to disentangled and interpretable latent spaces which can be used to effectively manipulate a wide range of data attributes spanning image and symbolic music domains.

1 Introduction

Over the last few years, deep generative models have emerged as powerful tools to create and manipulate data. These models have been applied to tasks in several domains such as image generation [42, 40, 19, 34], text generation [56, 15], speech generation [2, 23] and music creation [24, 41, 47, 46]. More recently, there has been considerable focus on improving the controllability of these models by enabling selective manipulation of data attributes such as changing the gender of the person in an image [33], changing the speaking style in a

Ashis Pati & Alexander Lerch
Center for Music Technology,
Georgia Institute of Technology,
Atlanta, USA
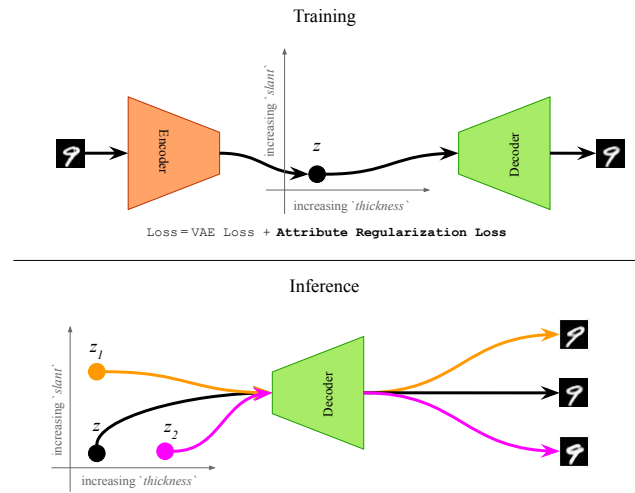E-mail: ashis.pati@gatech.edu & alexander.lerch@gatech.edu

Fig. 1 Motivation for the AR-VAE model which uses a novel attribute regularization loss (see Sect. 3.2) during the training step to force the latent space to encode specific attributes along specific dimensions of the latent space of a VAE. During inference, individual data attributes can be manipulated by simply traversing along these regularized dimensions

given speech excerpt [53], or increasing the density of notes in a musical excerpt [47, 21].

Latent representation-based models such as the Variational Auto-Encoders (VAEs) [28] have shown promise in this direction as they are able to encode certain hidden attributes of the data [10]. VAE-based latent spaces show interesting properties such as semantic interpolation [46] and attribute vector arithmetic [38, 47]. However, one of the key limitations of the vanilla-VAE framework is that the encoded attributes in the latent space cannot be explicitly controlled and the learnt attributes are often not interpretable by humans. This necessitates the need to rely on attribute vectors [10]. In order to circumvent this limitation, there has been substantial research on modifying the VAE training procedure to learn representations which are able to disentangle differ-
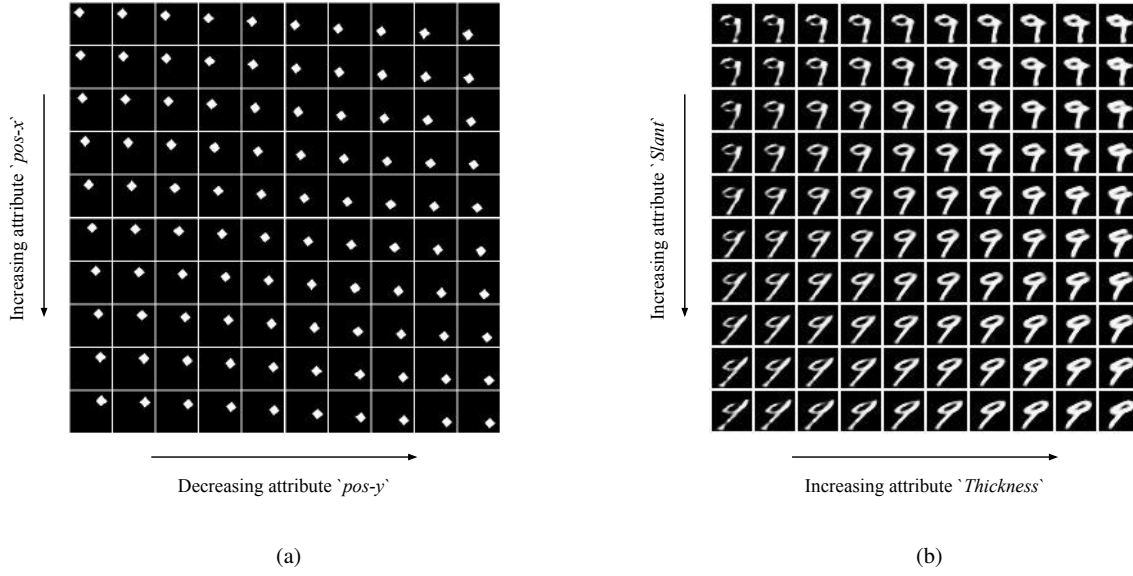
Increasing attribute `pos-x`

Decreasing attribute `pos-y`

Increasing attribute `Slant`

Increasing attribute `Thickness`

(a)   (b)

Fig. 2 Manipulating two attributes independently by traversing along the corresponding regularized dimensions of the AR-VAE latent space. (a) 2-d sprites image [37], (b) Morpho-MNIST digit [11]

ent data attributes using either unsupervised methods [22, 12, 48, 32] or supervised techniques [33, 21, 18, 5, 30]. However, there are certain limitations associated with both categories of methods which we discuss in Sect. 2.

In this paper, we propose Attribute-Regularized VAE (AR-VAE) which uses a new supervised training method to create structured latent spaces where specific attributes are forced to be encoded along specific dimensions of the latent space. In order to achieve this, we formulate a novel regularization loss which forces each specific attribute of interest to have a monotonic relationship with the latent code of the dimension along which we want to encode the attribute (hereafter referred to as the *regularized dimension*). Fig. 1 demonstrates this overall idea. The proposed approach has the key advantages that it:

(a) has a simple formulation with few hyperparameters,
(b) works with continuous data attributes, and
(c) is agnostic to how the attributes are computed/obtained.

Apart from the above, the supervised nature of the proposed method eliminates the need for any post-training analysis to interpret the relationship between the latent code and the data attributes.

We show that using the proposed method we are able to learn disentangled latent representations useful for manipulating a wide range of attributes across two different domains: images and music. The superior performance of AR-VAE compared to the baseline models is demonstrated using several objective and subjective experiments. Fig. 2 shows an example of the capability of our model to control a couple of attributes of 2-d sprites [37] and Morpho-MNIST digits [11].

The remainder of this paper is organized as follows: firstly, we present an overview of the related work in Sect. 2 followed by the description of our method in Sect. 3. We then provide details of our experimental set-up and results in Sect. 4 and 5, respectively. Finally, Sect. 6 concludes the paper and outlines avenues for future research.

2 Related Work

The approaches for attribute-based control over the output of generative models can be grouped into two broad categories. The first group of methods attempts to disentangle different factors of variation in the data [4]. The majority of the methods in this category are unsupervised techniques. Approaches from the second category rely on identifying certain attributes of interest and using supervised techniques in order to enable control during the generation process.

2.1 Unsupervised Disentanglement Learning

Unsupervised methods for disentanglement learning attempt to separate the distinct factors of variation in data [4] and learn a representation where changes to a single underlying factor of variation leads to changes in a single factor of the learned representation [36]. Most of the current approaches to unsupervised disentanglement are based on variations of the VAE framework [28] (see Sect. 3.1 for a short introduction). The main idea behind these approaches is that forcing the latent representation to have a factorized aggregated posterior should result in disentanglement [36]. This can be achieved

using different means such as imposing constraints on the information capacity of the latent space [22,9,48], maximizing the mutual information between a subset of the latent code and the observations [13], and maximizing the independence between the latent variables [12,26].

While many of these methods show good performance (based on one or more objective metrics for measuring disentanglement) on artificially generated image datasets (such as dSprites [37]), a recent study by Locatello et al. shows that not only are these approaches sensitive to inductive biases such as choice of network, hyperparameters, and random seeds, but also that some amount of supervision is necessary for learning effective disentanglement [36]. In addition, since these methods seek to learn a factorized latent representation they work well for low-level data attributes. However, as we show in our experiments, they do not extend well for complex data attributes (which are usually some combination of low-level attributes). Consequently, their practical usefulness is limited. Using unsupervised methods for manipulating attributes also requires a post-training analysis to determine how different attributes are encoded along the different dimensions of the latent space.

2.2 Supervised Regularization Methods

There has also been some research on fully supervised methods to control attributes by learning the transformations [55, 43]. However, these methods require availability of specifically annotated data regarding how each transformation changes a given data-point. Obtaining such data is costly and hence, such methods only have been shown to work with artificially created datasets.

An alternative is to use any available attribute information as conditioning inputs without needing explicit supervision. Methods within this group can broadly be differentiated into two categories, (a) methods relying on learning attribute-dependent latent spaces, and (b) methods relying on learning attribute-invariant latent spaces.

Attribute-dependent latent spaces attempt to explicitly encode different attributes along different dimensions of the latent space. These either encode individual attributes along individual dimensions [21,30], or decompose the latent space into different parts where each part corresponds to specific attributes [16,7] . There have also been attempts to use latent spaces of vanilla-VAEs and learn transformation and traversal methods to improve interpretability [1] and learn conditional generation [18,41].

Attribute-invariant latent spaces are those where the latent representation is independent of the attribute values. The idea is to learn a generalized latent representation which, when combined with the attribute specific information, generates a data-point with that attribute. Conditional VAEs and Generative Adversarial Networks (GANs) were early attempts at

this [49,54,39]. The more recent Fader networks proposed by Lample et al. demonstrated the effectiveness of this idea for image generation by using an adversarial training scheme [33].

Supervised methods also have their limitations. On the one hand, some methods are constrained to work only with certain types of data attributes. For instance, the Fader network, although designed for categorical attributes, has been shown to only work well with binary attributes [33]. On the other hand, some methods might impose additional constraints. For example, they might require the ability to generate or sample data-points by independently varying attributes [30], or require the ability to group data-points based on certain attributes [5,16]. In addition, very few supervised approaches are designed to work with continuous-valued attributes. To our best knowledge, the only model which is designed to exclusively work with continuous-valued data attributes is the Geodesic Latent Space Regularization VAE (GLSR-VAE) [21]. However, it requires differentiable computation of data attributes and careful tuning of hyperparameters.

3 Method

The goal of the proposed method is to train a structured latent space in which individual attributes are encoded along specific dimensions of the latent space. Such a latent structure enables controllable generation by selectively changing the latent code of the regularized dimension. For instance, if the attribute represents 'thickness' of an MNIST digit, and the regularized dimension corresponds to the first dimension of the latent space, then sampling latent vectors with increasing values of the first dimension should result in digits with increasing thickness as illustrated in Fig. 1. We first present a brief background on VAEs before describing the regularization loss formulation and the learning algorithm for AR-VAE.

3.1 Background on Variational Auto-Encoders

A Variational Auto-Encoder (VAE) [28] is a generative model which uses an auto-encoding [52] framework; during training, the model is forced to reconstruct its input. In a typical auto-encoder, the encoder learns to map data-points $\mathbf{x}$ from a high-dimensional data-space X to points in a low-dimensional space Z. This low-dimensional space is referred to as the *latent* space and points $\mathbf{z}$ in the latent space are called *latent vectors*. The decoder learns to map the latent vectors back to the data-space. VAEs treat the latent vector as a random variable and model the generative process as a sequence of sampling operations: $\mathbf{z} \sim p(\mathbf{z})$, and $\mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z})$, where $p_\theta(\mathbf{x}|\mathbf{z})$ is the decoder network parametrized by θ, and $p(\mathbf{z})$

is a prior distribution over the latent space. The posterior $q_\phi(\mathbf{z}|\mathbf{x})$ is approximated by the encoder with parameters ϕ. Variational Inference is used to approximate the posterior by minimizing the Kullback-Leibler (KL)-divergence [31] between the approximate posterior and the true posterior by maximizing the evidence lower bound (ELBO)

$$\log p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \quad (1)$$

where $\mathbb{E}[\cdot]$ is the mathematical expectation, $D_{\mathrm{KL}}(\cdot||\cdot)$ is the KL-divergence.

The first term of Eq. (1) can be interpreted as maximizing the reconstruction accuracy while the second term ensures that realistic samples are generated when latent vectors are sampled using the prior $p(\mathbf{z})$ [47]. In practice, the VAE optimization minimizes the loss function

$$L_{\mathrm{VAE}}(\theta, \phi) = L_{\mathrm{recons}}(\theta, \phi) + L_{\mathrm{KLD}}(\theta, \phi) \quad (2)$$

where $L_{\mathrm{recons}}(\theta, \phi)$ and $L_{\mathrm{KLD}}(\theta, \phi)$ are VAE reconstruction loss and the KL-Divergence regularization, respectively. The reconstruction loss is:

$$L_{\mathrm{recons}}(\theta, \phi) = \frac{1}{N} \sum_{i=1}^{N} \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2 \quad (3)$$

where N is the number of examples, $\hat{\mathbf{x}}$ is the reconstruction of $\mathbf{x}$ obtained using the encoder and decoder of the VAE. The L-2 norm in the above equation is replaced by cross-entropy loss for class prediction tasks. The regularization loss is

$$L_{\mathrm{KLD}}(\theta, \phi) = D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})). \quad (4)$$

As mentioned in Sect. 2, most unsupervised methods for learning disentangled representations use a slight modification of this objective. For instance, the β-VAE model [22,6] uses the following formulation

$$L_{\mathrm{VAE}}(\theta, \phi) = L_{\mathrm{recons}}(\theta, \phi) + \beta L_{\mathrm{KLD}}(\theta, \phi). \quad (5)$$

The core idea here is that using $\beta > 1$ encourages the independence of the dimensions of the latent space and leads to better disentanglement. However, the trade off is that increasing β might result in reduced reconstruction quality.

3.2 Attribute Regularization Loss

Let us consider a $\mathbb{D}$-dimensional latent space where latent vectors are represented as $\mathbf{z} : \{z^k\}, k \in [0, \mathbb{D})$. Formally, the objective is to encode an attribute a along a dimension r of the latent space such that, as we traverse along r, the attribute value a of the generated data increases. Mathematically, if $a(\mathbf{x}_i) > a(\mathbf{x}_j)$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two data-points generated using latent vectors $\mathbf{z}_i$ and $\mathbf{z}_j$, then $z_i^r > z_j^r$ should hold for any arbitrary i and j.

This is accomplished by adding an attribute-specific regularization loss to the VAE training objective. To compute this loss, we use a mini-batch containing m training examples and follow a three step process:

1. An attribute distance matrix $D_a \in \mathbb{R}^{m \times m}$ is computed for all examples in the training mini-batch:

$$D_a(i, j) = a(\mathbf{x}_i) - a(\mathbf{x}_j) \quad (6)$$

   where $i, j \in [0, m)$.

2. Next, a similar distance matrix $D_r \in \mathbb{R}^{m \times m}$ is computed for the regularized dimension r of the latent vectors:

$$D_r(i, j) = z_i^r - z_j^r. \quad (7)$$

3. The regularization loss is finally formulated as:

$$L_{r,a} = \mathrm{MAE}(\tanh(\delta D_r) - \mathrm{sgn}(D_a)), \quad (8)$$

   where $\mathrm{MAE}(\cdot)$ is the mean absolute error, $\tanh(\cdot)$ is the hyperbolic tangent function, $\mathrm{sgn}(\cdot)$ is the sign function, and δ is a tunable hyperparameter which decides the spread of the posterior distribution.

The $\mathrm{sgn}(\cdot)$ function is used with the attribute distance matrix D_a since we are only interested in whether the attribute value of a data-point is higher or lower than the others in the mini-batch and do not care about the magnitude of the differences. The $\tanh(\cdot)$ function is used for the regularized dimension distance matrix D_r since it has the same range as $\mathrm{sgn}(D_a)$, i.e, $[-1, 1]$, and we want to minimize the mean absolute error between $\tanh(\delta D_r)$ and $\mathrm{sgn}(D_a)$. In addition, $\tanh(\cdot)$ is a differentiable function which ensures that the loss is differentiable with respect to the latent vectors (and consequently the encoder parameters). Thus, we can use gradient descent for optimizing the network parameters. Other functions having such properties could also be potentially used to replace $\tanh(\cdot)$.

Overall, this formulation forces the latent code of the regularized dimension to have a monotonic relationship with the attribute values. Note that, unlike the GLSR-VAE formulation [21] which requires that the attributes should be computed using differentiable functions, our formulation is agnostic to the way in which the attributes are computed/obtained.

3.3 Learning Algorithm

If our attribute set $\mathscr{A} : \{a_l\}, l \in [0, \mathbb{L})$, contains $\mathbb{L}$ attributes, ($\mathbb{L} \leq \mathbb{D}$), then the overall loss function for AR-VAE is formulated as

$$L_{\mathrm{AR\text{-}VAE}} = L_{\mathrm{recons}} + \beta L_{\mathrm{KLD}} + \gamma \sum_{l=0}^{\mathbb{L}-1} L_{r_l, a_l} \quad (9)$$

where L_{r_l, a_l} is the regularization loss for the attribute a_l with r_l as the index of the regularized dimension, and γ is a tunable hyperparameter which we call as the regularization strength. We omit θ and ϕ in the above equation for brevity. The overall learning algorithm for AR-VAE is shown in Algorithm 1.

Algorithm 1: Learning algorithm for AR-VAE

Input: observations and attribute labels $(\mathbf{x}_i, \{a_l\}_i)_{i=1}^N$ (N is the number of examples in the dataset, $l \in [0, \mathbb{L})$ where $\mathbb{L}$ is the number of attributes), batch-size m, indices of the latent dimensions to be regularized $\{r_l\}_{l=0}^{\mathbb{L}-1}$, initialized VAE encoder and decoder parameters θ, ϕ, neural network optimizer g

repeat
    Randomly sample a batch of m data-points $(\mathbf{x}_i, \{a_l\}_i)$
    Compute $L_{\text{recons}}(\theta, \phi)$ using Eq. (3)
    Compute $L_{\text{KLD}}(\theta, \phi)$ using Eq. (4)
    for $l \in [0, \mathbb{L})$ **do**
        Compute L_{r_l, a_l} using Eq. (8)
    end
    Compute $L_{\text{AR-VAE}}(\theta, \phi)$ using Eq. (9)
    Update VAE encoder and decoder parameters:
        $\theta, \phi \leftarrow g(L_{\text{AR-VAE}}(\theta, \phi))$
until *convergence of objective*

---

4 Experimental Setup

In order to demonstrate the capabilities of the AR-VAE model, several quantitative and qualitative experiments are conducted. The quantitative experiments comprise of evaluating the degree of disentanglement of the latent space, reconstruction fidelity, ability to preserve content, and sensitivity to the two hyperparameters. The qualitative experiments comprise of manipulating data attributes using latent interpolations and traversals and visualizing the latent space with respect to the different attributes. Before delving deeper into the experimental results, we first present a brief description of the different datasets and attributes used, the baseline considered and some of the details related to the implementation of the models.

4.1 Datasets and Attributes

Most of the current research on disentanglement learning use image-based datasets such as 2-d sprites [37], 3-d shapes [8], CelebA [35], and 3-D chairs [3] for evaluation. This might be limiting since most methods are developed, tested and validated on a single data domain (images). In addition, some of these datasets such as 2-d sprites and 3-d shapes are artificially generated. While artificially created datasets are useful for benchmarking different methods, they have relatively simpler factors of variation. Considering this, we use datasets from two different domains (images and music), include both artificial and real-world data, and consider a diverse set of attributes.

The first dataset from the image domain is the 2-d sprites dataset which contains (≈ 0.7 million) two dimensional shapes having 5 simple factors of variation: *shape*, *scale*, *orientation*, *x-position*, and *y-position* [37]. This is standard dataset for evaluating disentanglement methods. The second dataset is

the Morho-MNIST dataset which contains the 70000 handwritten MNIST digits along with complex morphological attributes for each digit obtained using computational methods [11]. The attributes are: *area*, *length*, *thickness*, *slant*, *width*, and *height*.

From the music domain, two datasets consisting of single measures (bars) of monophonic melodies are used. The first dataset consists of measures extracted from the soprano parts of the J.S. Bach Chorales dataset [14] (≈ 350 chorales). The second dataset consists of measures extracted from (≈ 20000) folk melodies in the Scottish and Irish style [50]. Since pitch and rhythm are the two primary features of a monophonic melody, the following four attributes are considered for both datasets: (a) *rhythmic complexity*, based on Toussaints metrical complexity measure [51], (b) *pitch range*, the difference of lowest pitch value (in MIDI) in the measure from the highest pitch value, (c) *note density*, the count of the number of notes per measure, and (d) *contour*, the degree to which the melody moves up or down measured by summing up the difference in pitch values of all the notes in the measure. All these attributes are generated using computational methods (see Appendix A for details).

4.2 Baseline

Most of the unsupervised methods for disentanglement learning perform at par with one another [36]. Since the AR-VAE is effectively an extension of the β-VAE [22] (compare Eq. (5) to Eq. (9)), the latter is chosen as a baseline for comparison. Consequently, it also doubly serves as an ablation case. In addition, the β-VAE model is shown to perform at par with other supervised models [22].

Other supervised methods such as GLSR-VAE [21] and Fader networks [33] were also considered as potential baselines. However, the former requires differentiable computation of attributes (which cannot to applied to the datasets and attributes considered in this paper) and the later was designed for binary/categorical attributes. We investigated adapting the Fader network design to work with continuous attributes, however, the attempt did not lead to good results.

4.3 Implementation Details

Different model architectures are chosen for the different data domains. Convolutional architectures are used for the VAEs trained on images whereas recurrent architectures are used with the music-based datasets. To ensure consistency, all models for a particular dataset are trained for the same number of epochs using the same optimizer and learning rate. The model architectures and other details are provided in Appendix B and in our GitHub repository.[1].

---

[1] https://github.com/ashispati/ar-vae

For each dataset, both the β-VAE and the AR-VAE models were trained with 10 different random initializations. Based on initial experiments, for the β-VAE models, β was chosen as 4.0 and 0.001 for the image and music datasets, respectively (the low value of β was necessary to train the music VAE models [41,47]). For AR-VAE, the models for the image datasets were trained with $\gamma = 10.0$ and $\delta = 1.0$, whereas those for the music datasets were trained with $\gamma = 1.0$ and $\delta = 10.0$.

5 Results

5.1 Disentanglement

This experiment looks to evaluate the degree of disentanglement of the latent space with respect to the different data attributes.

There has been considerable work in the recent past to define objective metrics for measuring disentanglement of latent spaces [1,45,12,32,22,26,17]. Apart from disentangling the latent space, we are also interested in the degree to which AR-VAE is able to enforce a monotonic relationship between a given attribute and the regularized dimension. Taking this into account, the following metrics are used for this experiment: (a) *Interpretability* [1], which measures the ability to predict a given attribute using only one dimension of the latent space, (b) *Modularity* [45], which measures if each dimension of the latent space depends on only one attribute, (c) *Mutual Information Gap (MIG)* [12], which measures the difference of mutual information between a given attribute and the top two dimensions of the latent space which share maximum mutual information with the attribute, (d) *Separated Attribute Predictability (SAP)* [32], which measures the difference in the prediction error of the two most predictive dimensions of the latent space for a given attribute, and (e) *Spearman Correlation Coefficient (SCC)* score, which computes maximum value of the Spearman's correlation coefficient between an attribute and each dimension of the latent space. All metrics are aggregated by computing the mean across all attributes.

Fig. 3 shows the box plots with the performance of the models across all datasets and metrics. The following observations can be made. Firstly, AR-VAE clearly outperforms β-VAE across all metrics (except *Modularity*). Furthermore, this superior performance extends across all datasets in spite of the different domains and the varying degree of complexity in the attributes. The relatively poor performance for the *Modularity* metric can be explained by looking at the normalization method used: the score is divided by the maximum value of the mutual information of all the attributes with a latent dimension. This can result in high *Modularity* scores even with relatively low values of mutual information. Secondly,

the highest improvement is seen for the *Interpretability* metric and the *SCC* score. This is also expected since AR-VAE forces a monotonic relationship between a given attribute and the latent code of the regularized dimension. Thirdly, there is lower improvement in *MIG* which would suggest that there are other dimensions (apart from the regularized dimension) which share high mutual information with different attributes. This would also explain why the *SAP* score does not improve as much as the *Interpretability* metric. Finally, for the image-based datasets, the metrics for 2-d sprites are generally higher when compared to the morpho-MNIST dataset. This could be due to the artificial nature of the former and it's simpler attributes. For the music-based datasets, Folk music has better performance than Bach chorales. This might be due to the significantly larger size of the Folk dataset.

The superior performance of AR-VAE is also seen in the pairwise scatter plots shown in Fig. 4 where the results of datasets belonging to the same domain are aggregated. There is a clear difference in the performance of the two models when either the *Interpretability* metric or the *SCC* score are considered. This is evident from the separation in the distributions for the two models in the diagonal plots for these metrics. The distributions for the *MIG* and *SAP* score show inter-domain separation although there seems to some overlap for datasets within each domain.

5.2 Reconstruction Fidelity

The reconstruction quality is another important criterion. While a high degree of disentanglement is desirable, it should not come at the cost of reduced quality of the generation.

Fig. 5 shows the box plots for the reconstruction accuracies. While the performance of the two models is in the same range for the 2-d sprites dataset, AR-VAE performs better for the Morpho-MNIST dataset. A few example reconstructions for the 2-d sprites and Morpho-MNIST datasets are shown in Fig. 6. The better performance of AR-VAE can be gauged from are sharper reconstructions of MNIST digits.

For the music datasets, there is drop in performance for AR-VAE. The slight drop in case of the Folk dataset is expected since the same value of β is used for both AR-VAE and β-VAE models. AR-VAE thus has more constraints than β-VAE during training. The larger drop in performance for the Bach Chorales might be due to the smaller size of the dataset.

5.3 Content Preservation

In order to ascertain AR-VAE's ability to retain the content during interpolation, we conduct an additional experiment using the morpho-MNIST dataset using the identity of the digit as a proxy to the image content. Different variations

(a) *Interpretability*

(b) *Modularity*

(c) *MIG*

(d) *SAP*

(e) *SCC*

Fig. 3 Box plots for various disentanglement metrics (higher is better) across different datasets for ten different random seeds. The circular dots denote results for each random seed. AR-VAE scores better than β-VAE across the different datasets for all metrics except *Modularity*

of an input digit are generated by changing the attributes (via appropriate manipulation of the latent code) and then a pre-trained model is used to predict the digit class. The higher the classification accuracy, the better the model is at preserving the identity of the input. The experiment is run using the 10000 digits in the MNIST test set. For each digit, 60 variations are generated by interpolating along the regularized dimensions (10 for each of the six attributes). The pre-trained model has a ResNet-based architecture with an accuracy of 96.15% on the unmodified MNIST test set. The results of this experiment are shown in Fig. 7. While there is a only small difference in the performance of the two models in retaining the identity of the reconstructed digits, the AR-VAE model significantly outperforms the β-VAE for the interpolations. This indicates that AR-VAE is better for manipulating attributes while retaining the underlying content.

5.4 Hyperparameter Sensitivity

The next experiment assesses the sensitivity of the proposed model to the different hyperparameters (γ and δ). Fig. 8 shows the trade-off between reconstruction accuracy and the *Interpretability* metric as the hyperparameters are varied for the morpho-MNIST dataset while keeping β fixed at 1.0.

For lower values of γ, an increase in δ results in only marginal improvements in the *Interpretability* metric with-

out any loss in reconstruction accuracy. However, after γ crosses 1.0, increasing δ leads to significant improvement in the *Interpretability* metric accompanied by a slight drop in the reconstruction accuracy. Note that the β-VAE model performs considerably worse in comparison. Choosing γ in the $[5.0, 10.0]$ range and δ in the $[1.0, 10.0]$ range seems to give the best results.

5.5 Attribute Manipulation

This experiment presents a qualitative evaluation of the degree to which AR-VAE provides control over individual data attributes during the generation process. Given a data-point with latent code z, different variations for an attribute a_l are generated by changing the latent code of the corresponding regularized dimension z^{r_l} from -4 to 4. For comparison, similar variations are generated for the β-VAE model by using the dimension which has the highest mutual information with the attribute.

Images: Fig. 9 and 10 show the results of controlling attributes for the 2-d sprites and the Morpho-MNIST datasets, respectively. While both models are able to control the individual attributes to a similar degree for the 2-d sprites dataset (Fig. 9), the AR-VAE model performs better for the Morpho-MNIST dataset (Fig. 10). Not only are the interpolations meaningful with respect to the attribute of interest, but the
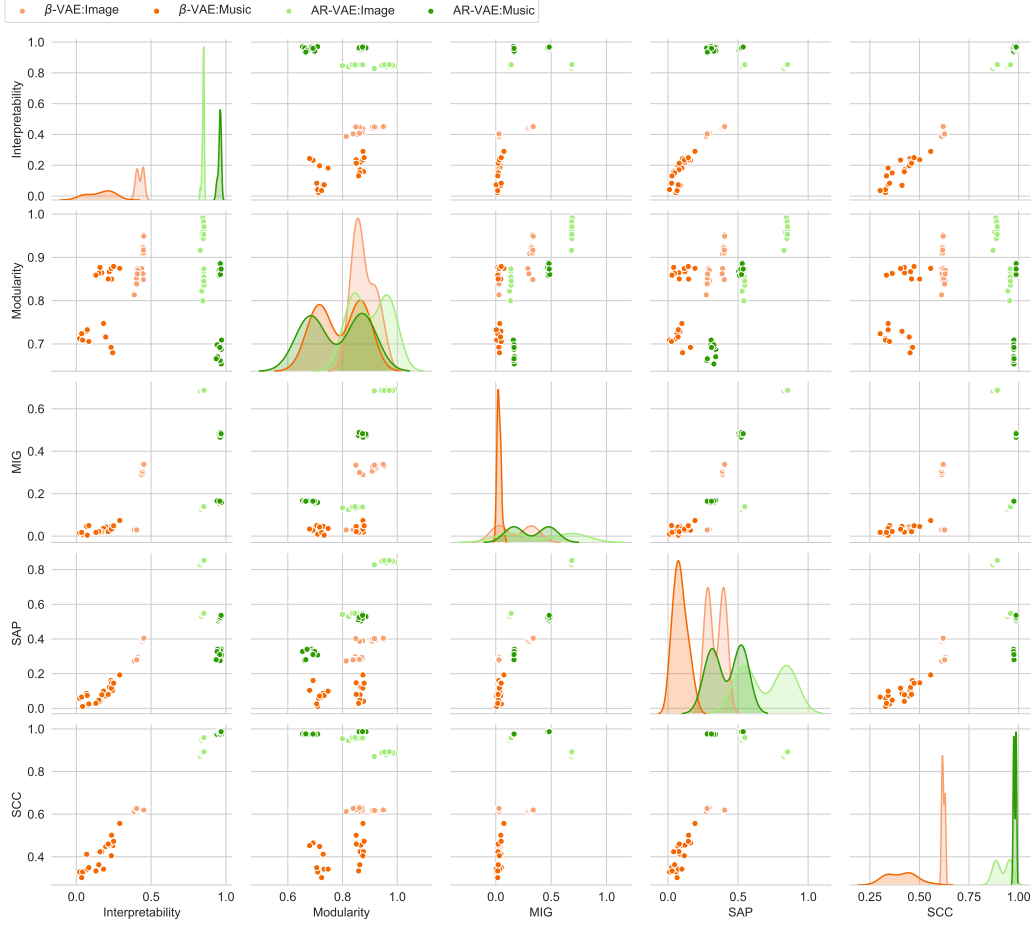
Fig. 4 Pairwise scatter plots of the different metrics aggregated over the image and music datasets. For *Interpretability* metric and *SCC* score, the performance of the AR-VAE model is clearly separable from that of β-VAE even across different data domains. There is some overlap for other metrics, with *Modularity* having the highest overlap
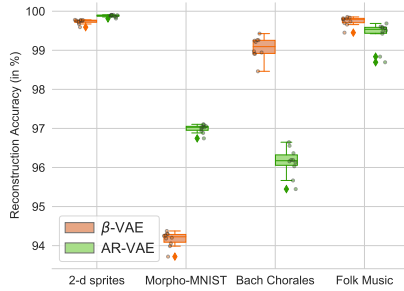

Fig. 5 Box plots of reconstruction accuracies (higher is better) for different datasets. AR-VAE and β-VAE shows better performance in image and music datasets respectively
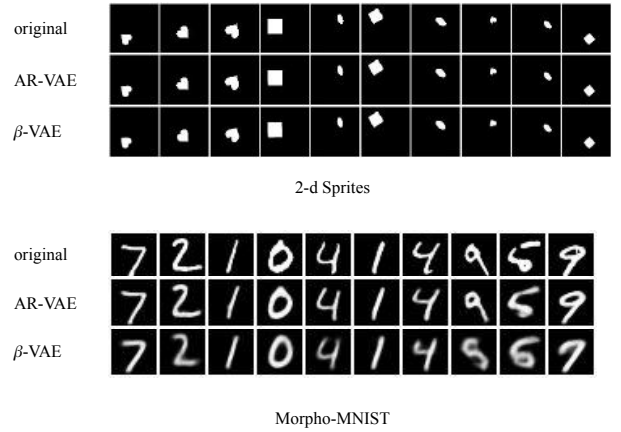

Fig. 6 Few examples of reconstructions in the 2-d sprites and Morpho-MNIST datasets. The AR-VAE model seems to have better reconstructions compared to the β-VAE model

AR-VAE is able to retain the identity of the original digit in most cases while β-VAE fails to do so (Sect. 5.3 further supports this observation). See Appendix C for more examples.

Music: Fig. 11 shows the results of manipulating different musical attributes in the Bach Chorales and Folk Music

datasets. For AR-VAE, in most cases, traversal along the regularized dimensions lead to measures with increasing val-

Fig. 7 Distribution of accuracy (higher is better) in predicting the MNIST test set digits. Higher values indicate greater ability of the model in preserving the image content. 'Reconstructed': prediction accuracy on reconstructed images from the MNIST test set, 'Interpolated': prediction accuracy on interpolations generated by traversing along the regularized dimensions for different attributes. The prediction accuracy on the original MNIST test set is shown with a ×
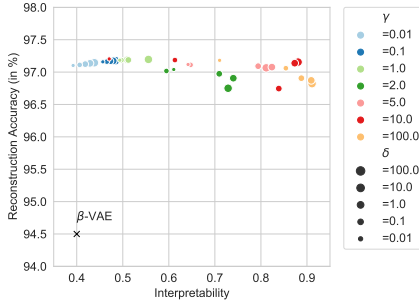

Fig. 8 Effect of the hyperparameters γ and δ on the performance of the AR-VAE model (β is fixed at 1.0) on the Morpho-MNIST dataset. Each dot corresponds to a unique combination of γ and δ which are indicated by the color and the size of the dots respectively. The β-VAE model ($\beta = 4$) is also shown (with a ×) for reference. The ideal model should lie on the top right corner of the plot (with high values of both reconstruction accuracy and *Interpretability* metric)
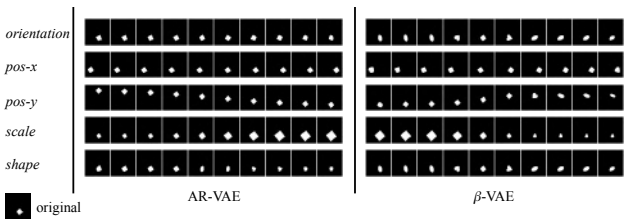

Fig. 9 Controlling different attributes of a 2-d sprites image (shown in bottom left). AR-VAE model is able to manipulate each attribute independently. However, while the β-VAE model is also able to separately control *pos-x*, *pos-y*, and *scale*, it changes the *shape* of the sprite while manipulating *orientation* and vice versa

ues of the attributes. These can be seen more clearly in the attribute value plots to the right of the piano-rolls.

On the contrary, for β-VAE, the progression of the attributes is not as uniform. In fact, in the Bach Chorales example, there is no change in the values of two out of the four attributes. This is because, in the case of β-VAE, on
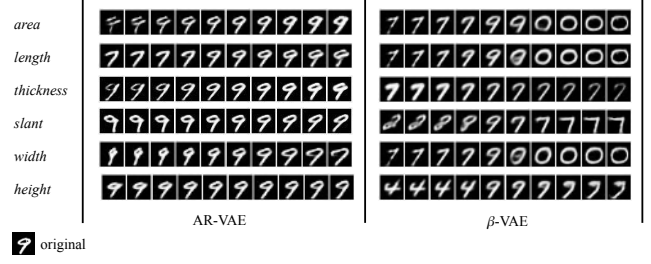

Fig. 10 Controlling different attributes of an MNIST digit (shown in bottom left). The AR-VAE model is able to separately control the individual attributes of the original image and retains the identity of the digit in most cases. The β-VAE model, on the other hand, fails to retain the identity of original digit and the interpolations are not meaningful

many occasions there is a single dimension of the latent space which has maximum mutual information with two or more attributes. This could be either due to a high degree of correlation between attributes (e.g., *Rhythmic Complexity* and *Note Density*) or due to very poor disentanglement where all attributes are poorly encoded. Even in cases where the attributes have a nice progression (e.g., see bottom right row for the *Contour* attribute in the Folk Music examples in Fig. 11), the order of the encoding is reversed (the attribute decreases with increasing value of the latent code). This warrants a post-hoc analysis of the latent space to understand the relationship between the attributes and the latent dimensions.

It is important to point out that while AR-VAE allows better control over the different attributes for both datasets, it sometimes struggles to maintain the musical coherency of the generated measures. For instance, the generated measures shown in Fig. 11 are not always in the same key. Although this seems to the case for measures generated with the β-VAE model as well, the problem seems to be more pronounced for AR-VAE. We observed that training separate models for the different attributes, where each model regularizes a single attribute, tends to alleviate this limitation. Traversing along the regularized dimensions in these models typically results in better musical coherence.

5.6 Latent Space Visualization

In this experiment, the interpretability of the latent space with respect to different attributes is investigated by generating attribute surface plots. For each attribute, A 2-dimensional plane on the latent space is considered which comprises of the regularized dimension (*x*-axis) for the attribute and a non-regularized dimension (*y*-axis). The latent code for the other dimensions are drawn from a normal distribution and kept fixed. The latent vectors thus obtained are passed through the decoder and the attributes of the generated data are computed. Fig. 12 and 13 show the result of this visualization for the Morpho-MNIST and the Folk Music dataset, respectively.
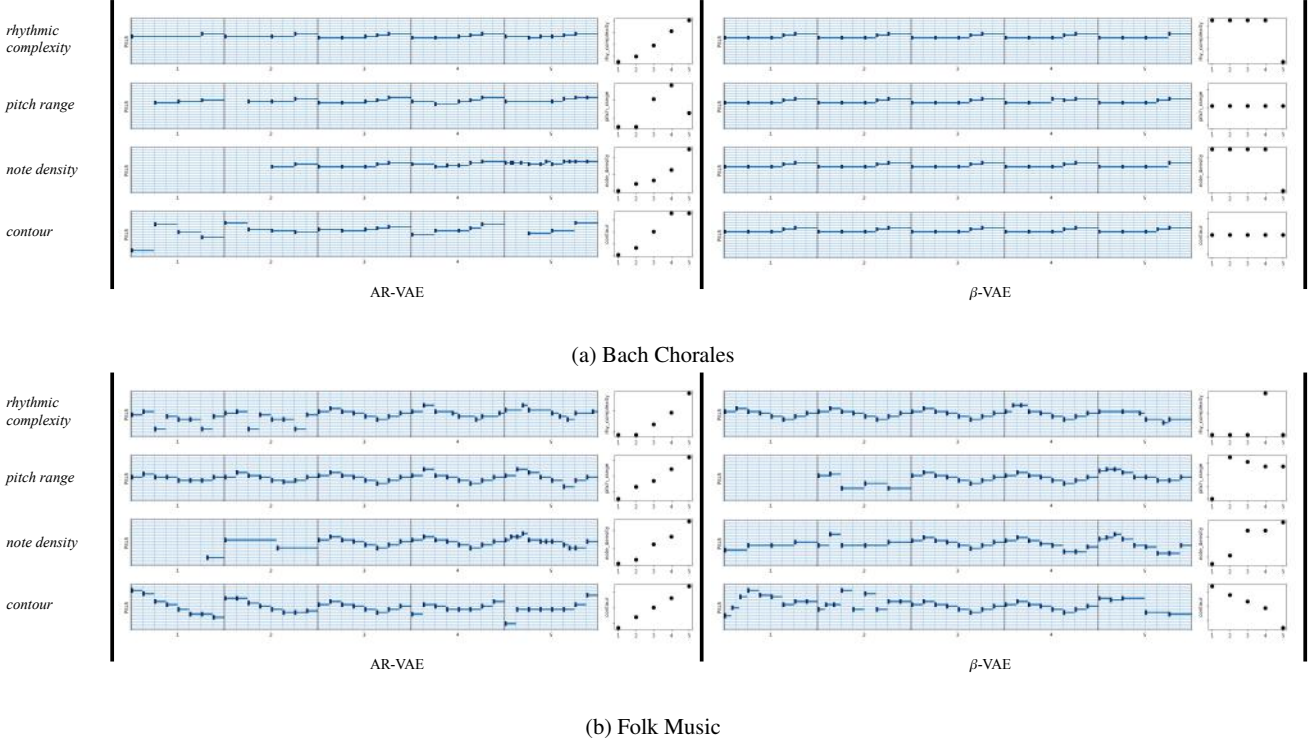
(a) Bach Chorales

(b) Folk Music

Fig. 11 Controlling different attributes of musical measures. The piano-rolls show measures generated by increasing latent code of the regularized dimension (or in the case of β-VAE, the dimension with the highest mutual information) for the respective attribute. The light vertical lines within each measure denote the location of the eighth-notes. The y-axis of the piano rolls show pitch in semi-tones. The plots on the right show how the attribute values change with the increase in the latent code. For AR-VAE, in most cases, traversing along the regularized dimension leads to increase in attribute values

For both datasets, most attributes show an increasing trend as the latent code of the regularized dimension is increased. This is seen by the gradual transition from purple to yellow color along the x-axis. The change in color is minimal for the y-axis which indicates independence of the attribute from the non-regularized dimension. Thus, the AR-VAE latent space is interpretable with respect to the attribute values and traversing along the regularized dimension will lead to an increase in the corresponding attribute. The only attributes for which this does not hold are *Length* and *Rhythmic Complexity* in the morpho-MNIST and Folk Music datasets, respectively. The reason for the poor performance for *Rhythmic Complexity* could be that it is strongly correlated with *Note Density* (≈ 0.89). This is also reflected in the relatively poor *Interpretability* score for *Rhythmic Complexity* (0.83) compared to the other attributes (≈ 0.99). The choice of attributes and their relationship to each other, thus, becomes an important consideration to create meaningful latent spaces.

6 Conclusion

This paper investigates the problem of selective manipulation of data attributes in deep generative models by structuring the latent space of a VAE to encode specific attributes along specific dimensions of the latent space. The proposed AR-

VAE model uses a novel regularization loss to enforce a monotonic relationship between the attributes and the latent code of the respective regularized dimension. The resulting latent spaces are easily interpretable and allow manipulation of individual attributes by simple traversals along the regularized dimensions. The regularization loss works for continuous data attributes, and has a simple formulation with only two hyperparameters. In addition, contrary to previous supervised methods [21, 30], the loss formulation is agnostic to how the attributes are computed or obtained. Using both image and music-based datasets, we show that AR-VAE can work with different types of data, model architectures, and a wide range of data attributes. We show that AR-VAE leads to better disentanglement of the latent space, and creates meaningful attribute-based interpolations while preserving the content of the original data. This superior performance is achieved without any significant drop in the reconstruction quality of the VAE. While AR-VAE is designed to work with VAE-based architectures, the proposed regularization method should also work for other types of generative models such as Auto-Encoders [52], GANs [20], and flow-based models [44].

The proposed method can be used as a building block to create several interesting and useful applications for context-driven data generation. For instance, it can be used to ma-
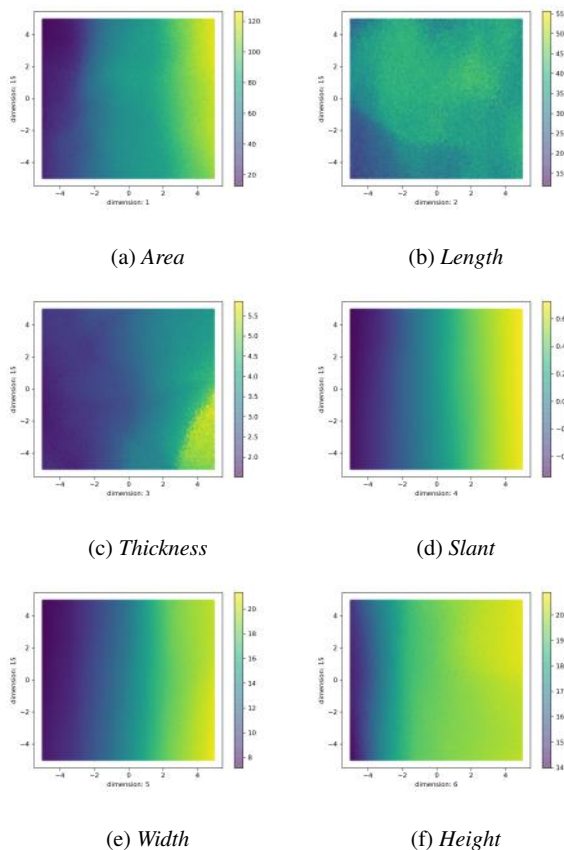
(a) *Area* (b) *Length*

(c) *Thickness* (d) *Slant*

(e) *Width* (f) *Height*

Fig. 12 Attribute surface plots for the Morpho-MNIST dataset. Each plots shows the attribute values for the decoded latent vectors on a 2-dimensional surface of the latent space (the latent code for the other dimensions is kept fixed). The *x*-axis corresponds to the regularized dimension for the attribute and the *y*-axis corresponds to a non-regularized dimension. The attribute values increase (gradual transition from purple to yellow color) as the latent code for the regularized dimension is increased for most attributes

(a) *Rhythmic Complexity* (b) *Pitch Range*
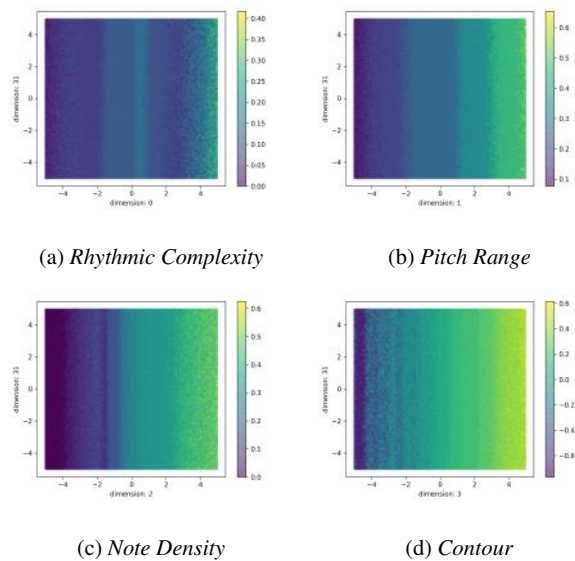
(c) *Note Density* (d) *Contour*

Fig. 13 Attribute surface plots for the Folk Music dataset. Each plot shows the attribute values for the decoded latent vectors on a 2-dimensional surface of the latent space (the latent code for the other dimensions is kept fixed). The *x*-axis corresponds to the regularized dimension for the attribute and the *y*-axis corresponds to a non-regularized dimension. The attribute values increase (gradual transition from purple to yellow color) as the latent code for the regularized dimension is increased for most attributes

nipulate attributes in image or photo-based applications such as FaceApp[2] or Prisma.[3] In the context of music, it can be used to build interactive tools or plugins to aid composers and music creators. For instance, composers would be able to manipulate different attributes (such as rhythmic complexity) of the composed music to try different ideas and meet specific compositional requirements. This would allow fast iteration and would be especially useful for novices and hobbyists. Since the method is agnostic to how the attributes are computed, it can potentially be useful to manipulate high-level musical attributes such as tension and emotion. This will be particularly useful for music generation in the context of video games where the background music can be suitably changed to match the emotional context of the game and the actions of the players. There is also the possibility of using this for speech generation applications. For instance,

the ability to manipulate the prosody of the generated speech can make mobile voice assistants more realistic.

There are, however, some limitations of the approach which can open up avenues for future research. The regularization loss is currently designed to work with continuous attributes. While we present evidence that it also works for discrete categorical attributes (e.g., *shape* in the 2-d sprites dataset), additional experiments will be needed to ascertain this. Furthermore, the current formulation is not suitable for binary attributes such as the ones used in Fader networks [33]. We also observe that the choice of attributes seems to play an important role in the training process. While we are able to jointly regularize multiple attributes in most cases (as seen in Sect. 5.5 and 5.6), strongly correlated attributes can lead to latent spaces which are not interpretable with respect to every single attribute. This results in poor control over some attributes and reduced content preservation and coherence in the interpolations. While this is a limitation, we think that independent control over strongly correlated attributes is probably not a strict requirement for a useful generative model. Even in cases where such control is necessary, two or more separately trained AR-VAE models can be used which regularize those attributes individually.

---

[2] https://faceapp.com/app, last accessed: 20th July 2020
[3] https://prisma-ai.com, last accessed: 20th July 2020

Acknowledgements

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Adel, T., Ghahramani, Z., Weller, A.: Discovering Interpretable Representations for Both Deep Generative and Discriminative Models. In: 35th International Conference on Machine Learning (ICML), pp. 50–59. Stockholm, Sweden (2018)
2. Akuzawa, K., Iwasawa, Y., Matsuo, Y.: Expressive Speech Synthesis via Modeling Expressions with Variational Autoencoder. In: 19th Interspeech. Graz, Austria (2018)
3. Aubry, M., Maturana, D., Efros, A.A., Russell, B.C., Sivic, J.: Seeing 3D Chairs: Exemplar Part-based 2D-3D Alignment using a Large Dataset of CAD Models. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3762–3769. Columbus, Ohio, USA (2014)
4. Bengio, Y., Courville, A., Vincent, P.: Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(8), 1798–1828 (2013)
5. Bouchacourt, D., Tomioka, R., Nowozin, S.: Multi-Level Variational Autoencoder: Learning Disentangled Representations From Grouped Observations. In: 32nd AAAI Conference on Artificial Intelligence. New Orleans, USA (2018)
6. Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Jozefowicz, R., Bengio, S.: Generating Sentences from a Continuous Space. In: SIGNLL Conference on Computational Natural Language Learning. Berlin, Germany (2016)
7. Brunner, G., Konrad, A., Wang, Y., Wattenhofer, R.: MIDI-VAE: Modeling Dynamics and Instrumentation of Music with Applications to Style Transfer. In: 19th International Society for Music Information Retrieval Conference (ISMIR). Paris, France (2018)
8. Burges, C., Hyunjik, K.: 3d-shapes Dataset (2020). URL https://github.com/deepmind/3d-shapes. Last accessed, 2nd April 2020
9. Burgess, C.P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., Lerchner, A.: Understanding disentangling in β-VAE. arXiv:1804.03599 [cs, stat] (2018)
10. Carter, S., Nielsen, M.: Using Artificial Intelligence to Augment Human Intelligence. Distill 2(12), e9 (2017). DOI 10.23915/distill.00009
11. Castro, D.C., Tan, J., Kainz, B., Konukoglu, E., Glocker, B.: Morpho-MNIST: Quantitative Assessment and Diagnostics for Representation Learning. Journal of Machine Learning Research 20 (JMLR) (2019)
12. Chen, R.T.Q., Li, X., Grosse, R., Duvenaud, D.: Isolating Sources of Disentanglement in Variational Autoencoders. In: Advances in Neural Information Processing Systems 31 (NeurIPS) (2018)
13. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In: Advances in Neural Information Processing Systems 29 (NeurIPS), pp. 2172–2180 (2016)
14. Cuthbert, M.S., Ariza, C.: music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data. In: 11th International Society of Music Information Retrieval Conference (ISMIR). Utrecht, The Netherlands (2010)
15. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. In: Assoiation of Computational Linguistics (ACL). Florence, Italy (2019)
16. Donahue, C., Lipton, Z.C., Balsubramani, A., McAuley, J.: Semantically Decomposing the Latent Spaces of Generative Adversarial Networks. In: 6th International Conference on Learning Representations (ICLR). Vancouver, Canada (2018)
17. Eastwood, C., Williams, C.K.I.: A Framework for the Quantitative Evaluation of Disentangled Representations. In: 6th International Conference on Learning Representations (ICLR). Vancouver, Canada (2018)
18. Engel, J., Hoffman, M., Roberts, A.: Latent Constraints: Learning to Generate Conditionally from Unconditional Generative Models. In: 5th International Conference on Learning Representations (ICLR). Toulon, France (2017)
19. Gatys, L.A., Ecker, A.S., Bethge, M.: Image Style Transfer Using Convolutional Neural Networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2414–2423. Las Vegas, USA (2016)
20. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets. In: Advances in Neural Information Processing Systems 27 (NeurIPS), pp. 2672–2680 (2014)
21. Hadjeres, G., Nielsen, F., Pachet, F.: GLSR-VAE: Geodesic latent space regularization for variational autoencoder architectures. In: IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–7. Hawaii, USA (2017)
22. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M.M., Mohamed, S., Lerchner, A.: β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In: 5th International Conference on Learning Representations (ICLR). Toulon, France (2017)
23. Hsu, W.N., Zhang, Y., Glass, J.: Learning Latent Representations for Speech Generation and Transformation. In: 18th Interspeech. Stockholm, Sweeden (2017)
24. Huang, C.Z.A., Vaswani, A., Uszkoreit, J., Simon, I., Hawthorne, C., Shazeer, N., Dai, A.M., Hoffman, M.D., Dinculescu, M., Eck, D.: Music Transformer: Generating Music with Long-Term Structure. In: 6th International Conference on Learning Representations (ICLR). Vancouver, Canada (2018)
25. Jozefowicz, R., Zaremba, W., Sutskever, I.: An Empirical Exploration of Recurrent Network Architectures. In: 32nd International Conference on Machine Learning (ICML). Lille, France (2015)
26. Kim, H., Mnih, A.: Disentangling by Factorising. In: 35th International Conference on Machine Learning (ICML). Stockholm, Sweeden (2018)
27. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: 3rd International Conference on Learning Representations (ICLR). San Diego, USA (2015)
28. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: 2nd International Conference on Learning Representations (ICLR). Banff, Canada (2014)
29. Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S.: Self-Normalizing Neural Networks. In: Advances in Neural Information Processing Systems 30 (NeurIPS), pp. 971–980 (2017)
30. Kulkarni, T.D., Whitney, W.F., Kohli, P., Tenenbaum, J.: Deep Convolutional Inverse Graphics Network. In: Advances in Neural Information Processing Systems 28 (NeurIPS), pp. 2539–2547 (2015)
31. Kullback, S., Leibler, R.A.: On Information and Sufficiency. The Annals of Mathematical Statistics 22(1), 79–86 (1951)

32. Kumar, A., Sattigeri, P., Balakrishnan, A.: Variational Inference of Disentangled Latent Concepts from Unlabeled Observations. In: 5th International Conference on Learning Representations (ICLR). Toulon, France (2017)

33. Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L., Ranzato, M.A.: Fader Networks:Manipulating Images by Sliding Attributes. In: Advances in Neural Information Processing Systems 30 (NeurIPS), pp. 5967–5976 (2017)

34. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4681–4690. Hawaii, USA (2017)

35. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep Learning Face Attributes in the Wild. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 3730–3738. Santiago, Chile (2015)

36. Locatello, F., Bauer, S., Lucic, M., Rtsch, G., Gelly, S., Schlkopf, B., Bachem, O.: Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. In: 36th International Conference on Machine Learning (ICML). Long Beach, California, USA (2019)

37. Matthey, L., Higgins, I., Hassabis, D., Lerchner, A.: dSprites: Disentanglement testing Sprites dataset (2017). URL https://github.com/deepmind/dsprites-dataset. Last accessed, 2nd April 2020

38. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: Advances in Neural Information Processing Systems 26 (NeurIPS), pp. 3111–3119 (2013)

39. Mirza, M., Osindero, S.: Conditional Generative Adversarial Nets. arXiv:1411.1784 [cs, stat] (2014). URL http://arxiv.org/abs/1411.1784

40. van den Oord, A., Kalchbrenner, N., Espeholt, L., kavukcuoglu, k., Vinyals, O., Graves, A.: Conditional Image Generation with Pixel-CNN Decoders. In: Advances in Neural Information Processing Systems 29 (NeurIPS), pp. 4790–4798 (2016)

41. Pati, A., Lerch, A., Hadjeres, G.: Learning to Traverse Latent Spaces for Musical Score Inpainting. In: 20th International Society for Music Information Retrieval Conference (ISMIR). Delft, The Netherlands (2019)

42. Razavi, A., van den Oord, A., Vinyals, O.: Generating Diverse High-Fidelity Images with VQ-VAE-2. In: Advances in Neural Information Processing Systems 32 (NeurIPS), pp. 14866–14876 (2019)

43. Reed, S.E., Zhang, Y., Zhang, Y., Lee, H.: Deep Visual Analogy-Making. In: Advances in Neural Information Processing Systems 28 (NeurIPS), pp. 1252–1260 (2015)

44. Rezende, D.J., Mohamed, S.: Variational Inference with Normalizing Flows. In: 32nd International Conference on Machine Learning (ICML). Lille, France (2015). ArXiv: 1505.05770

45. Ridgeway, K., Mozer, M.C.: Learning Deep Disentangled Embeddings With the F-Statistic Loss. In: Advances in Neural Information Processing Systems 31 (NeurIPS), pp. 185–194 (2018)

46. Roberts, A., Engel, J., Oore, S., Eck, D.: Learning Latent Representations of Music to Generate Interactive Musical Palettes. In: Intelligent User Interfaces Workshops (IUI). Tokyo, Japan (2018)

47. Roberts, A., Engel, J., Raffel, C., Hawthorne, C., Eck, D.: A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. In: 35th International Conference on Machine Learning (ICML). Stockholm, Sweeden (2018)

48. Rubenstein, P., Scholkopf, B., Tolstikhin, I.: Learning Disentangled Representations with Wasserstein Auto-Encoders. In: 6th International Conference on Learning Representations (ICLR), Workshop Track. Vancouver, Canada (2018)

49. Sohn, K., Lee, H., Yan, X.: Learning Structured Output Representation using Deep Conditional Generative Models. In: Advances in Neural Information Processing Systems 28 (NeurIPS) (2015)

50. Sturm, B.L., Santos, J.F., Ben-Tal, O., Korshunova, I.: Music transcription modelling and composition using deep learning. In: 1st International Conference on Computer Simulation of Musical Creativity (CSMC). Huddersfield, UK (2016)

51. Toussaint, G.: A Mathematical Analysis of African, Brazilian and Cuban Clave Rhythms. In: BRIDGES: Mathematical Connections in Art, Music and Science, pp. 157–168 (2002)

52. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: 25th International Conference on Machine Learning (ICML), pp. 1096–1103. Helsinki, Finland (2008)

53. Wang, Y., Stanton, D., Zhang, Y., Skerry-Ryan, R.J., Battenberg, E., Shor, J., Xiao, Y., Ren, F., Jia, Y., Saurous, R.A.: Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis. In: 35th International Conference on Machine Learning (ICML). Stockholm, Sweden (2018)

54. Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2Image: Conditional Image Generation from Visual Attributes. In: B. Leibe, J. Matas, N. Sebe, M. Welling (eds.) European Conference for Computer Vision (ECCV), pp. 776–791. Amsterdam, The Netherlands (2016)

55. Yang, J., Reed, S.E., Yang, M.H., Lee, H.: Weakly-supervised Disentangling with Recurrent Transformations for 3D View Synthesis. In: Advances in Neural Information Processing Systems 28 (NeurIPS), pp. 1099–1107 (2015)

56. Zhang, Y., Gan, Z., Fan, K., Chen, Z., Henao, R., Shen, D., Carin, L.: Adversarial feature matching for text generation. In: 34th International Conference on Machine Learning (ICML), pp. 4006–4015. Sydney, Australia (2017)

APPENDIX

A Computation of Musical Attributes

The data representation scheme from [41] is chosen where each mono-phonic measure of music M is a sequence of N symbols $\{m_t\}$, $t \in [0, N)$, where $N = 24$. The set of symbols consists of note names (e.g., A#, Eb, B, C), a continuation symbol '$\_\_$', and a special token for Rest. The computation steps for the musical metrics are as follows:

(a) *Rhythmic Complexity (r)*: This attribute measures the rhythmic complexity of the a given measure. To compute this, a complexity coefficient array $\{f_t\}$, $t \in [0, N)$ is first constructed which assigns weights to different metrical locations based on Toussaints metrical complexity measure [51]. Metrical locations which are on the beat are given low weights while locations which are off-beat are given higher weights. The attribute is computed by taking a weighted average of the note onset locations with the complexity coefficient array f. Mathematically,

$$r(M) = \frac{\sum_{t=0}^{N-1} \text{ONSET}(m_t).f_t}{\sum_{t=0}^{N-1} f_t}, \qquad (10)$$

where $\text{ONSET}(\cdot)$ detects if there is a note onset at location t, i.e., it is 1 if m_t is a note name symbol and 0 otherwise.

(b) *Pitch Range (p)*: This is computed as the normalized difference between the maximum and minimum MIDI pitch values:

$$p(M) = \frac{1}{R} \left(\max_{t \in [0, N)} (\text{MIDI}(m_t)) - \min_{t \in [0, N)} (\text{MIDI}(m_t)) \right), \qquad (11)$$

where $\text{MIDI}(\cdot)$ computes the pitch value in MIDI for the note symbol. The MIDI pitch value for Rest and '$\_\_$' symbols are set to zero. The normalization factor R is based on the range of the dataset.

(c) *Note Density (d)*: This measures the count of the number of notes per measure normalized by the total length of the measure sequence:

$$d(M) = \frac{1}{N} \sum_{i=0}^{N-1} \text{ONSET}(m_t), \qquad (12)$$

where $\text{ONSET}(\cdot)$ has the same meaning as in Eq. (10) above.

(d) *Contour (c)*: This measures the degree to which the melody moves up or down and is measured by summing up the difference in pitch values of all the notes in the measure. Mathematically,

$$c(M) = \frac{1}{R} \sum_{t=0}^{N-2} [\text{MIDI}(m_{t+1}) - \text{MIDI}(m_t)], \qquad (13)$$

where $\text{MIDI}(\cdot)$ and R have same meaning as in Eq. (11) above.

B Implementation Details

Image-based Models: For the image-based models, a stacked con-volutional VAE architecture is used. The encoder consists of a stack of N 2-dimensional convolutional layers followed by a stack of linear layers. The decoder mirrors the encoder and consists of a stack of linear layers followed by a stack of N 2-dimensional transposed convolutional layers. The configuration details are given in Table 2.

Music-based Models: For the music-based models, the model archi-tecture is based on our previous work of musical score inpainting [41]. A hierarchical recurrent VAE architecture is used. Fig. 14 shows the over-all schematic of the architecture and Table 1 provides the configuration details.
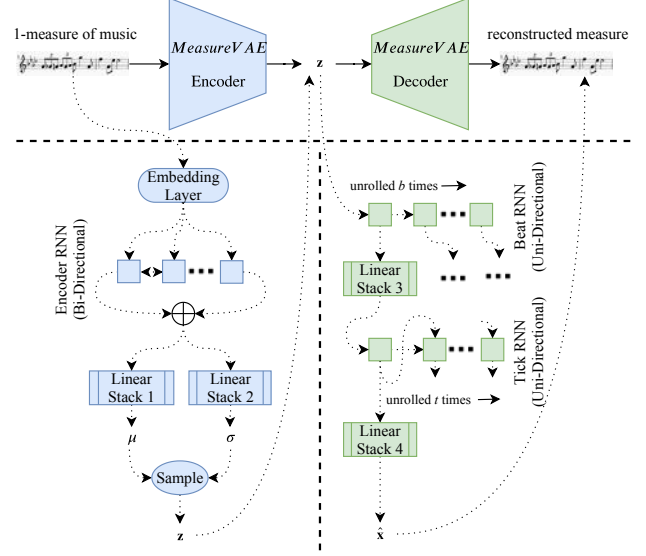

Fig. 14 MeasureVAE schematic. Individual components of the encoder and decoder are shown below the main blocks (dotted arrows indicate data flow within the individual components). $\mathbf{z}$ denotes the latent vector and $\hat{\mathbf{x}}$ denotes the reconstructed measure, $b = 4$ denotes the number of beats in a measure and $t = 6$ denotes the number of symbols/ticks in a beat. Fig. taken from [41]

Measure VAE	
Embedding Layer	i=dict size, o=10
EncoderRNN	n=2, i=10, h=128, d=0.5 , type=GRU
Linear Stack 1 Linear Stack 2	i=512, o=32, n=2, non-linearity=SELU
BeatRNN	n=2, i=1, h=128, d=0.5, type=GRU
TickRNN	n=2, i=138, h=128, d=0.5, type=GRU
Linear Stack 3	i=128, o=256, n=1, non-linearity=SELU
Linear Stack 4	i=128, o=dict size, n=1, non-linearity=ReLU

Table 1 Table showing configurations of the MeasureVAE architecture. n: number of layers, i: input size, o: output size, h: hidden size, d: dropout probability, SELU: Scaled Exponential Linear Unit [29], ReLU: Rectifier Linear Unit, GRU: Gated Recurrent Units [25]

Training Details: All models for the same dataset are trained for the same number of epochs (models for both image-based datasets and Bach Chorales are trained for 100 epochs, models for the Folk Music dataset are trained for 30 epochs). The optimization is carried out using the ADAM optimizer [27] with a fixed learning rate of $1e{-}4$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 1e{-}8$.

All the models are implemented using the Python programming language and the Pytorch[4] library.

C Additional Results

Some additional examples from the image-based datasets are shown in Fig. 15 and 16. The musical scores for AR-VAE generated interpolations from Fig. 11 is shown in Fig. 17.

---

[4] https://pytorch.org, last accessed: 20th July 2020

Model Type	2-d Sprites VAE	Mnist VAE
Encoder Convolutional Stack	4-layer Convolutional Network: Conv(i=1, o=32, k=4, s=2, p=1) + ReLU Conv(i=32, o=32, k=4, s=2, p=1) + ReLU Conv(i=32, o=32, k=4, s=2, p=1) + ReLU Conv(i=32, o=32, k=4, s=2, p=1) + ReLU	3-layer Convolutional Network: Conv(i=1, o=64, k=4, s=1, p=0) + SELU + Dropout(d=0.5) Conv(i=64, o=64, k=4, s=1, p=0) + SELU + Dropout(d=0.5) Conv(i=64, o=8, k=4, s=1, p=0) + SELU + Dropout(d=0.5)
Encoder Linear Stack	3-layer Linear Network: Linear(i=512, o=256) + ReLU Linear(i=256, o=256) + ReLU Linear(i=256, o=10) × 2 (in parallel)	2-layer Linear Network: Linear(i=2888, o=256) + SELU Linear(i=256, o=16) × 2 (in parallel)
Decoder Linear Stack	3-layer Linear Network: Linear(i=10, o=256) + ReLU Linear(i=256, o=256) + ReLU Linear(i=256, o=512) + ReLU	2-layer Linear Network: Linear(i=16, o=256) + SELU Linear(i=256, o=2888) + SELU
Decoder Convolutional Stack	4-layer Transposed Convolutional Network: TrConv(i=32, o=32, k=4, s=2, p=1) + ReLU TrConv(i=32, o=32, k=4, s=2, p=1) + ReLU TrConv(i=32, o=32, k=4, s=2, p=1) + ReLU TrConv(i=32, o=1, k=4, s=2, p=1)	3-layer Transposed Convolutional Network: TrConv(i=8, o=64, k=4, s=1, p=0) + SELU + Dropout(d=0.5) TrConv(i=64, o=64, k=4, s=1, p=0) + SELU + Dropout(d=0.5) TrConv(i=64, o=1, k=4, s=1, p=0)

Table 2 Table showing configurations of the VAEs for the image-based datasets. In the Encoder Linear Stack, the last layer has two parallel linear layers for computing the mean and log standard deviation of the latent vectors respectively. Conv: 2-dimensional convolutional layer, TrConv: 2-dimensional transposed convolutional layer, i: input channels, o: output channels, k: kernel size, s: stride, p: padding, d: dropout probability, SELU: Scaled Exponential Linear Unit [29], ReLU: Rectifier Linear Unit
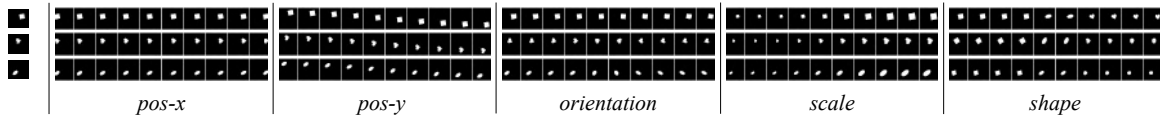

pos-x   *pos-y*   *orientation*   *scale*   *shape*

Fig. 15 Manipulating attributes for three different shapes from the 2-d sprites dataset images using AR-VAE. The interpolations for each attribute are generated by changing the latent code of the corresponding regularized dimension for the original shapes shown on the extreme left. Attributes can be manipulated independently
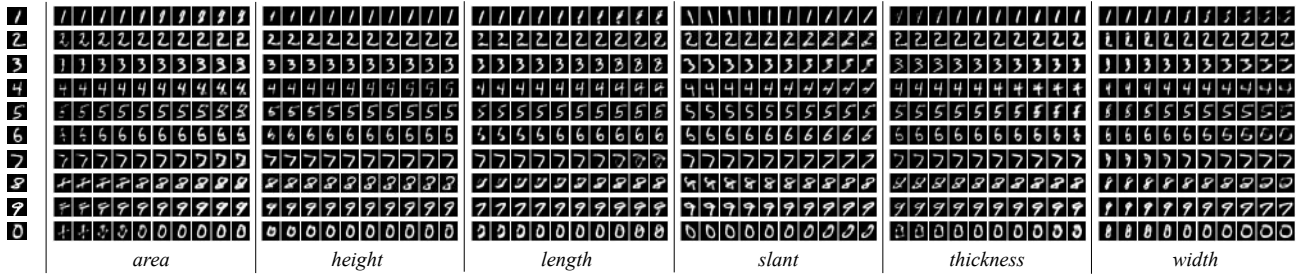

area   *height*   *length*   *slant*   *thickness*   *width*

Fig. 16 Manipulating attributes for ten different digits from the Morpho-MNIST dataset images using AR-VAE. The interpolations for each attribute are generated by changing the latent code of the corresponding regularized dimension for the original digits shown on the extreme left. AR-VAE is able to manipulate the different attributes and is able to retain the digit identity in most cases

rhythmic complexity

pitch range

note density

contour

Bach Chorales   Folk Music

Fig. 17 Musical score corresponding to the AR-VAE generated interpolations from Fig. 11. While the attribute values of the generated measures are controlled effectively, the musical coherence is often lost (particularly in the case of Bach Chorales)