# dMELODIES: A MUSIC DATASET FOR DISENTANGLEMENT LEARNING

**Ashis Pati**     **Siddharth Gururani**     **Alexander Lerch**

Center for Music Technology, Georgia Institute of Technology, USA

`ashis.pati@gatech.edu, siddgururani@gatech.edu, alexander.lerch@gatech.edu`

## ABSTRACT

Representation learning focused on disentangling the underlying factors of variation in given data has become an important area of research in machine learning. However, most of the studies in this area have relied on datasets from the computer vision domain and thus, have not been readily extended to music. In this paper, we present a new symbolic music dataset that will help researchers working on disentanglement problems demonstrate the efficacy of their algorithms on diverse domains. This will also provide a means for evaluating algorithms specifically designed for music. To this end, we create a dataset comprising of 2-bar monophonic melodies where each melody is the result of a unique combination of nine latent factors that span ordinal, categorical, and binary types. The dataset is large enough ($\approx 1.3$ million data points) to train and test deep networks for disentanglement learning. In addition, we present benchmarking experiments using popular unsupervised disentanglement algorithms on this dataset and compare the results with those obtained on an image-based dataset.

## 1. INTRODUCTION

Representation learning deals with extracting the underlying factors of variation in a given observation [1]. Learning compact and *disentangled* representations (see Figure 1 for an illustration) from given data, where important factors of variation are clearly separated, is considered useful for generative modeling and for improving performance on downstream tasks (such as speech recognition, speech synthesis, vision and language generation [2–4]). Disentangled representations allow a greater degree of interpretability and controllability, especially for content generation, be it language, speech, or music. In the context of Music Information Retrieval (MIR) and generative music models, learning some form of disentangled representation has been the central idea for a wide variety of tasks such as genre transfer [5], rhythm transfer [6, 7], timbre synthesis [8], instrument rearrangement [9], manipulating musical attributes [10, 11], and learning music similarity [12].

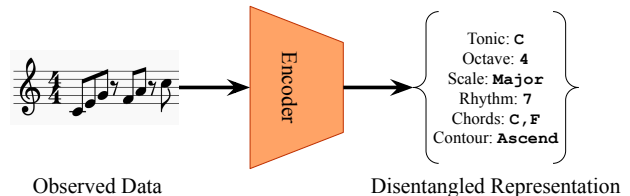Consequently, there exists a large body of research in

**Figure 1**: Disentanglement example where a high dimensional observed data is disentangled into a low dimensional representation comprising of semantically meaningful factors of variation.

the machine learning community focused on developing algorithms for learning disentangled representations. These span unsupervised [13–16], semi-supervised [17–19] and supervised [10, 20–22] methods. However, a vast majority of these algorithms are designed, developed, tested, and evaluated using data from the image or computer vision domain. The availability of standard image-based datasets such as dSprites [23], 3D-Shapes [24], and 3D-Chairs [25] among others has fostered disentanglement studies in vision. Additionally, having well-defined factors of variation (for instance, size and orientation in dSprites [23], pitch and elevation in Cars3D [26]) has allowed systematic studies and easy comparison of different algorithms. However, this restricted focus on a single domain raises concerns about the generalization of these methods [27] and prevents easy adoption into other domains such as music.

Research on disentanglement learning in music has often been application-oriented with researchers using their own problem-specific datasets. The factors of variation have also been chosen accordingly. To the best of our knowledge, there is no standard dataset for disentanglement learning in music. This has prevented systematic research on understanding disentanglement in the context of music.

In this paper, we introduce *dMelodies*, a new dataset of monophonic melodies, specifically intended for disentanglement studies. The dataset is created algorithmically and is based on a simple and yet diverse set of independent latent factors spanning ordinal, categorical and binary attributes. The full dataset contains $\approx 1.3$ million data points which matches the scale of image datasets and should be sufficient to train deep networks. We consider this dataset as the primary contribution of this paper. In addition, we also conduct benchmarking experiments using three popular unsupervised methods for disentanglement learning and present a comparison of the results with the dSprites dataset [23]. Our experiments show that disentanglement

learning methods do not directly translate between the image and music domains and having a music-focused dataset will be extremely useful to ascertain the generalizability of such methods. The dataset is available online [1] along with the code to reproduce our benchmarking experiments. [2]

## 2. MOTIVATION

In representation learning, given an observation $\mathbf{x}$, the task is to learn a representation $r(\mathbf{x})$ which "makes it easier to extract useful information when building classifiers or other predictors" [1]. The fundamental assumption is that any high-dimensional observation $\mathbf{x} \in \mathcal{X}$ (where $\mathcal{X}$ is the dataspace) can be decomposed into a semantically meaningful low dimensional latent variable $\mathbf{z} \in \mathcal{Z}$ (where $\mathcal{Z}$ is referred to as the latent space). Given a large number of observations in $\mathcal{X}$, the task of disentanglement learning is to estimate this low dimensional latent space $\mathcal{Z}$ by separating out the distinct factors of variation [1]. An ideal disentanglement method ensures that changes to a single underlying factor of variation in the data changes only a single factor in its representation [27]. From a generative modeling perspective, it is also important to learn the mapping from $\mathcal{Z}$ to $\mathcal{X}$ to enable better control over the generative process.

### 2.1 Lack of diversity in disentanglement learning

Most state-of-the-art methods for unsupervised disentanglement learning are based on the Variational Auto-Encoder (VAE) [28] framework. The key idea behind these methods is that factorizing the latent representation to have an aggregated posterior should lead to better disentanglement [27]. This is achieved using different means, e.g., imposing constraints on the information capacity of the latent space [13, 29, 30], maximizing the mutual information between a subset of the latent code and the observations [31], and maximizing the independence between the latent variables [14, 15]. However, unsupervised methods for disentanglement learning are sensitive to inductive biases (such network architectures, hyperparameters, and random seeds) and consequently there is a need to properly evaluate such methods by using datasets from diverse domains [27].

Apart from unsupervised methods for disentanglement learning, there has also been some research on semi-supervised [18, 19] and supervised [20, 21, 32, 33] learning techniques to manipulate specific attributes in the context of generative models. In these paradigms, a labeled loss is used in addition to the unsupervised loss. Available labels can be utilized in various ways. They can help with disentangling known factors (e.g., digit class in MNIST) from latent factors (e.g., handwriting style) [34], or supervising specific latent dimensions to map to specific attributes [10]. However, most of these approaches are evaluated using image domain datasets.

Tremendous interest from the machine learning community has led to the creation of benchmarking datasets (albeit image-based) specifically targeted towards disentanglement learning such as dSprites [23], 3D-Shapes [24], 3D-chairs [25], MPI3D [35], most of which are artificially generated and have simple factors of variation. While one can argue that artificial datasets do not reflect real-world scenarios, the relative simplicity of these datasets is often desirable since they enable rapid prototyping.

### 2.2 Lack of consistency in music-based studies

Representation learning has also been explored in the field of MIR. Much like images, learning better representations has been shown to work well for MIR tasks such as composer classification [36, 37], music tagging [38], and audio-to-score alignment [39]. The idea of disentanglement has been particularly gaining traction in the context of interactive music generation models [5, 6, 11, 33]. Disentangling semantically meaningful factors can significantly improve the usefulness of music generation tools. Many researchers have independently tried to tackle the problem of disentanglement in the context of symbolic music by using different musically meaningful attributes such as genre [5], note density [10], rhythm [6], and timbre [8]. However, these methods and techniques have all been evaluated using different datasets which makes a direct comparison impossible. Part of the reason behind this lack of consistency is the difference in the problems that these methods were looking to address. However, the availability of a common dataset allowing researchers to easily compare algorithms and test their hypotheses will surely aid systematic research.

## 3. dMELODIES DATASET

The primary objective of this work is to create a simple dataset for music disentanglement that can alleviate some of the shortcomings mentioned in Section 2: first, researchers interested in disentanglement will have access to more diverse data to evaluate their methods, and second, research on music disentanglement will have the means for conducting systematic, comparable evaluation. This section describes the design choices and the methodology used for creating the proposed *dMelodies* dataset.

While core MIR tasks such as music transcription, or tagging focus more on analysis of audio signals, research on generative models for music has focused more on the symbolic domain. Considering most of the interest in disentanglement learning stems from research on generative models, we decided to create this dataset using symbolic music representations.

### 3.1 Design Principles

To enable objective evaluation of disentanglement algorithms, one needs to either know the ground-truth values of the underlying factors of variation for each data point, or be able to synthesize the data points based on the attribute values. The dSprites dataset [23], for instance, consists of single images of different 2-dimensional shapes with simple attributes specifying the position, scale and orientation of these shapes against a black background. The design of our

---

[1] https://github.com/ashispati/dmelodies_dataset
[2] https://github.com/ashispati/dmelodies_benchmarking

dataset is loosely based on the dSprites dataset. The following principles were used to finalize other design choices:

(a) The dataset should have a simple construction with homogenous data points and intuitive factors of variation. It should allow for easy differentiation between data points and have clearly distinguishable latent factors.

(b) The factors of variation should be independent, i.e., changing any one factor should not cause changes to other factors. While this is not always true for real-world data, it enables consistent objective evaluation.

(c) There should be a clear one-to-one mapping between the latent factors and the individual data points. In other words, each unique combination of the factors should result in a unique data point.

(d) The factors of variation should be diverse. In addition, it would be ideal to have the factors span different types such as discrete, ordinal, categorical and binary.

(e) Finally, the different combinations of factors should result in a dataset large enough to train deep neural networks. Based on size of the different image-based datasets [23,40], we would require a dataset of the order of at least a few hundred thousand data points.

## 3.2 Dataset Construction

Considering the design principles outlined above, we decided to focus on monophonic pitch sequences. While there are other options such as polyphonic or multi-instrumental music, the choice of monophonic melodies was to ensure simplicity. Monophonic melodies are a simple form of music uniquely defined by the pitch and duration of their note sequences. The pitches are typically based on the key or scale in which the melody is being played and the rhythm is defined by the onset positions of the notes.

Since the set of all possible monophonic melodies is very large and heterogeneous, the following additional constraints were imposed on the melody in order to enforce homogeneity and satisfy the other design principles:

(a) Each melody is based on a scale selected from a finite set of allowed scales. This choice of scale also serves as one of the factors of variation. The melody will also be uniquely defined by the pitch class of the tonic (root pitch) and the octave number.

(b) In order to constrain the space of all possible pitch patterns within a scale, we restrict each melody to be an arpeggio over the standard I-IV-V-I cadence chord pattern. Consequently, each melody consists of 12 notes (3 notes for each of the 4 chords).

(c) In order to vary the pitch patterns, the direction of arpeggiation of each chord, i.e. up or down, is used as a latent factor. This choice adds a few binary factors of variation to the dataset.

(d) The melodies are fixed to 2-bar sequences with 8th note as the minimum note duration. This makes the dataset uniform in terms of sequence lengths of the data points and also helps reduce the complexity of the sequences. 2-bar sequences have been used in other music generation studies as well [10, 41]. We use a tokenized data representation such that each melody is

| Factor | # Options | Notes |
|--------|-----------|-------|
| *Tonic* | 12 | C, C#, D, through B |
| *Octave* | 3 | Octave 4, 5 and 6 |
| *Scale* | 3 | major, harmonic minor, and blues |
| *Rhythm Bar 1* | 28 | $\binom{8}{6}$, based on onset locations of 6 notes |
| *Rhythm Bar 2* | 28 | $\binom{8}{6}$, based on onset locations of 6 notes |
| *Arp Chord 1* | 2 | up/down, for Chord 1 |
| *Arp Chord 2* | 2 | up/down, for Chord 2 |
| *Arp Chord 3* | 2 | up/down, for Chord 3 |
| *Arp Chord 4* | 2 | up/down, for Chord 4 |

**Table 1**: Table showing the different factors of variation for the dMelodies dataset. Since all factors of variation are independent, the total dataset contains 1,354,752 unique melodies.

a sequence of length 16.

(e) If we consider the space of all possible unique rhythms, the number of options will explode to $\binom{16}{12}$ which will be significantly larger than other factors of variation. Hence, we choose to break the latent factor for rhythm into 2 independent factors: rhythm for bar 1 and bar 2.

(f) The rhythm of a melody is based on the metrical onset position of the notes [42]. Consequently, rhythm is dependent on the number of notes. In order to keep rhythm independent from other factors, we constrain each bar to have 6 notes (play 2 chords) thereby obtaining $\binom{8}{6}$ options for each bar.

Based on the above design choices, the dMelodies dataset consists of 2-bar monophonic melodies with 9 factors of variations listed in Table 1. The factors of variation were chosen to satisfy the design principles listed in Section 3.1. For instance, while melodic transformations such as repetition, inversion, retrograde would have made more musical sense, they did not allow creation of a large-enough dataset with independent factors of variation. The resulting dataset thus contains simple melodies which do not adequately reflect real-world musical data. A side-effect of this choice of factors is that some of them (such as arpeggiation direction and rhythm) affect only a specific part of the data. Since each unique combination of these factors results in a unique data point we get 1,354,752 unique melodies. Figure 2 shows one such melody from the dataset and its corresponding latent factors. The dataset is generated using the *music21* [43] python package.

## 4. BENCHMARKING EXPERIMENTS

In this section, we present benchmarking experiments to demonstrate the performance of some of the existing unsupervised disentanglement algorithms on the proposed dMelodies dataset and contrast the results with those obtained on the image-based dSprites dataset.
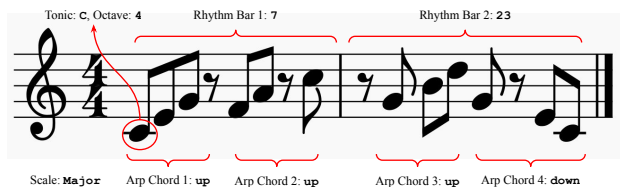
**Figure 2**: Example of a sample melody from the dMelodies dataset. Also shown are the values of the different latent factors. For rhythm latent factors, the shown value corresponds to the index from the rhythm dictionary.

### 4.1 Experimental Setup

We consider 3 different disentanglement learning methods: $\beta$-VAE [13], Annealed-VAE [29], and FactorVAE [15]. All these methods are based on different regularization terms applied to the VAE loss function.

#### 4.1.1 Data Representation

We use a tokenized data representation [44] with the 8th-note as the smallest note duration. Each 8th note position is encoded with a token corresponding to the note name which starts on that position. A special continuation symbol ('__') is used which denotes that the previous note is held. A special token is used for rest.

#### 4.1.2 Model Architectures

Two different VAE architectures are chosen to conduct these experiments. The first architecture (dMelodies-CNN) is based on Convolutional Neural Networks (CNNs) and is similar to those used for several image-based VAEs, except that we use 1-D convolutions. The second architecture (dMelodies-RNN) is based on a hierarchical recurrent model [41, 45]. Details of the model architectures are provided in the supplementary material.

#### 4.1.3 Hyperparameters

Each learning method has its own regularizing hyperparameter. For $\beta$-VAE, we use three different values of $\beta \in \{0.2, 1.0, 4.0\}$. This choice is loosely based on the notion of normalized-$\beta$ [13]. In addition, we force the KL-regularization only when the KL-divergence exceeds a fixed threshold $\tau = 50$ [41, 46]. For Annealed-VAE, we fix $\gamma = 1.0$ and use three different values of capacity, $C \in \{25.0, 50.0, 75.0\}$. For FactorVAE, we use the Annealed-VAE loss function with a fixed capacity ($C = 50$), and choose three different values for $\gamma \in \{1, 10, 50\}$.

#### 4.1.4 Training Specifications

For each of the above methods, model, and hyperparameter combination, we train 3 models with different random seeds. To ensure consistency across training, all models are trained with a batch-size of 512 for 100 epochs. The ADAM optimizer [47] is used with a fixed learning rate of $1e-4$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-8$. For $\beta$-VAE and Annealed-VAE, we use 10 warm-up epochs where $\beta = 0.0$. After warm-up, the regularization hyperparameter ($\beta$ for

$\beta$-VAE and $C$ for Annealed-VAE) is annealed exponentially from 0.0 to their target values over 100000 iterations. For FactorVAE, we stick to the original implementation and do not anneal any of the parameters in the loss function. The VAE optimizer is the same as mentioned earlier. The Factor-VAE discriminator is optimized using ADAM with a fixed learning rate of $1e-4$, $\beta_1 = 0.8$, $\beta_2 = 0.9$, and $\epsilon = 1e-8$. We found that utilizing the original hyperparameters [15] for this optimizer led to unstable training on dMelodies.

For comparison with dSprites, we present the results for all the three methods using a CNN-based VAE architecture. The set of hyperparameters and other training configurations were kept the same for the dSprites dataset, except for the FactorVAE where we use the originally proposed loss function and discriminator optimizer hyperparameters, as the model does not converge otherwise.

#### 4.1.5 Disentanglement Metrics

The following objective metrics for measuring disentanglement are used: (a) *Mutual Information Gap (MIG)* [14], which measures the difference of mutual information between a given latent factor and the top two dimensions of the latent space which share maximum mutual information with the factor, (b) *Modularity* [48], which measures if each dimension of the latent space depends on only one latent factor, and (c) *Separated Attribute Predictability (SAP)* [16], which measures the difference in the prediction error of the two most predictive dimensions of the latent space for a given factor. For each metric, the mean across all latent factors is used for aggregation. For consistency, standard implementations of the different metrics are used [27].

### 4.2 Experimental Results

#### 4.2.1 Disentanglement

In this experiment, we present the comparative disentanglement performance of the different methods on dMelodies. The result for each method is aggregated across the different hyperparameters and random seeds. Figure 3 shows the results for all three disentanglement metrics. We group the trained models based on the architecture. The results for the dSprites dataset are also shown for comparison.

First, we compare the performance of different methods on dMelodies. Annealed-VAE shows better performance for MIG and SAP. These metrics indicate the ability of a method to ensure that each factor of variation is mapped to a single latent dimension. The performance in terms of Modularity is similar across the different methods. High Modularity indicates that each dimension of the latent space maps to only a single factor of variation. For dSprites, FactorVAE seems to be best method overall across metrics. However, the high variance in the results shows that choice of random seeds and hyperparameters is probably more important than the disentanglement method itself. This is in line with observations in previous studies [27].

Second, we observe no significant impact of model architecture on the disentanglement performance. For both the CNN and the hierarchical RNN-based VAE, the performance of all the different methods on dMelodies is
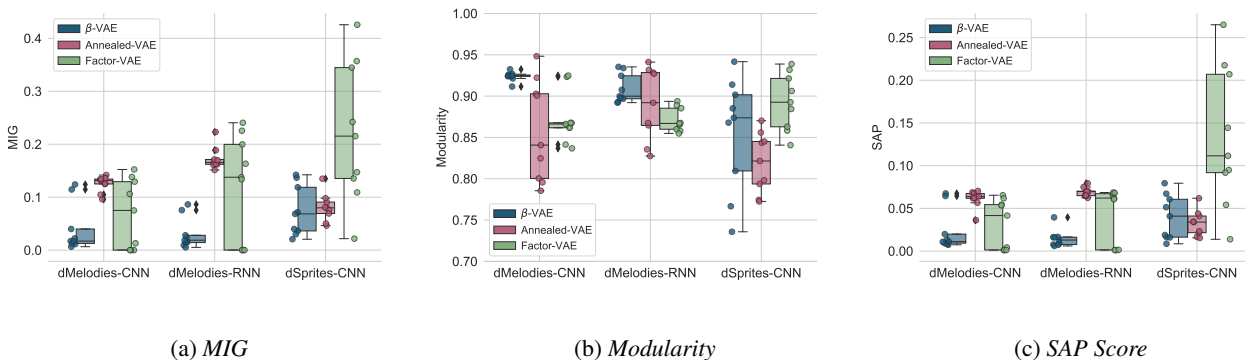
(a) *MIG*       (b) *Modularity*       (c) *SAP Score*

**Figure 3**: Overall disentanglement performance (higher is better) of different methods on the dMelodies and dSprites datasets. Individual points denote results for different hyperparameter and random seed combinations. Please refer to supplementary material Sec.2.1 for the best hyperparameter settings.
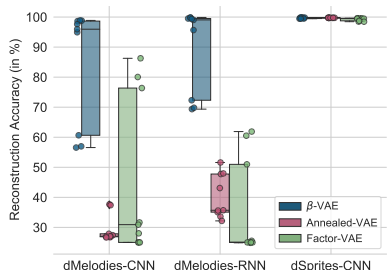


**Figure 4**: Overall reconstruction accuracies (higher is better) of the different methods on the dMelodies and dSprites datasets. Individual points denote results for different hyperparameter and random seed combinations.

comparable. This might be due to the relatively short sequence lengths used in dMelodies which do not fully utilize the capabilities of the hierarchical-RNN architecture (which has been shown to work well in learning long-term dependencies [41]). On the positive side, this indicates that the dMelodies dataset might be agnostic to the VAE-architecture.

Finally, we compare differences in the performance between the two datasets. In terms of MIG and SAP, the performance for dSprites is slightly better (especially for Factor-VAE), while for Modularity, performance across both datasets is comparable. However, once again, the differences are not significant. Looking at the disentanglement metrics alone, one might be tempted to conclude that the different methods are domain invariant. However, as the next experiments will show, there are significant differences.

### 4.2.2 Reconstruction Fidelity

From a generative modeling standpoint, it is important that along with better disentanglement performance we also retain good reconstruction fidelity. This is measured using the reconstruction accuracy shown in Figure 4. It is clear that all three methods fail to achieve a consistently good reconstruction accuracy on dMelodies. $\beta$-VAE gets an accuracy $\geq 90\%$ for some hyperparameter values (more on this in

Section 4.2.3). However, both Annealed-VAE and Factor-VAE struggle to cross a median-accuracy of $40\%$ (which would be unusable from a generative modeling perspective). The performance of the hierarchical RNN-based VAE is slightly better than the CNN-based architecture. In comparison, for dSprites, all three methods are able to consistently achieve better reconstruction accuracies.

### 4.2.3 Sensitivity to Hyperparameters

The previous experiments presented aggregated results over the different hyperparameter values for each method. Next, we take a closer look at the individual impact of those hyperparameters, i.e., the effect of changing the hyperparameters on the disentanglement performance (MIG) and the reconstruction accuracy. Figure 5 shows this in the form of scatter plots. The ideal models should lie on the top right corner of the plots (with high values of both reconstruction accuracy and MIG).

Models trained on dMelodies are very sensitive to hyperparameter adjustments. This is especially true for reconstruction accuracy. For instance, increasing $\beta$ for the $\beta$-VAE model improves MIG but severely reduces reconstruction performance. For Annealed-VAE and Factor-VAE there is a wider spread in the scatter plots. For Annealed-VAE, having a high capacity $C$ seems to marginally improve reconstruction (especially for the recurrent VAE). For FactorVAE, increasing $\gamma$ leads to a drop in both disentanglement and reconstruction.

Contrast this with the scatter plots for dSprites. For all three methods, the hyperparameters seem to only significantly affect the disentanglement performance. For instance, increasing $\beta$ and $\gamma$ (for $\beta$-VAE and FactorVAE, respectively) result in clear improvement in MIG. More importantly, however, there is no adverse impact on the reconstruction accuracy.

### 4.2.4 Factor-wise Disentanglement

We also looked at how the individual factors of variation are disentangled. We consider the $\beta$-VAE model for this since it has the highest reconstruction accuracy. Figure 6

(a) $\beta$-VAE: Varying $\beta$       (b) Annealed-VAE: Varying $C$       (c) Factor-VAE: Varying $\gamma$
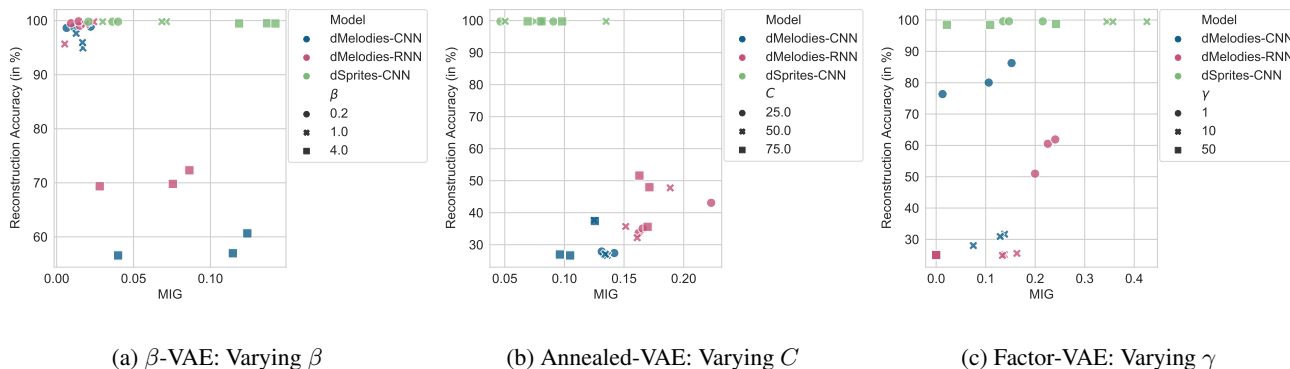
**Figure 5**: Effect of the hyperparameters on the different disentanglement methods. Overall, for improving disentanglement on dMelodies results in severe drop in reconstruction accuracy. The dSprites dataset does not suffer from this drawback.
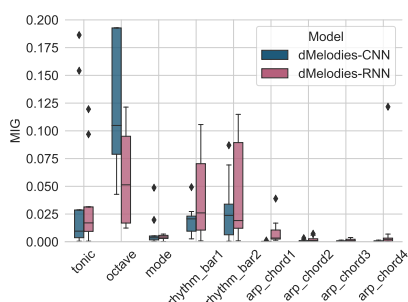


**Figure 6**: Factor-wise MIG for the $\beta$-VAE method.

shows the factor-wise *MIG* for both the CNN and RNN-based models. Factors corresponding to octave and rhythm are disentangled better. This is consistent with some recent research on disentangling rhythm [6, 7]. In contrast, the factors corresponding to the arpeggiation direction perform the worst. This might be due to their binary type. Similar analysis for the dSprites dataset reveals better disentanglement for the scale and position based factors. Additional results are provided in the supplementary material.

## 5. DISCUSSION

As mentioned in Section 2, disentanglement techniques have been shown to be sensitive to the choice of hyper-parameters and random seeds [27]. The results obtained in our benchmarking experiments in the previous section using dMelodies seem to ascertain this even further. We find that methods which work well for image-based datasets do not extend directly to the music domain. When moving between domains, not only do we have to tune hyperparameters separately, but the model behavior may vary significantly when hyperparameters are changed. For instance, reconstruction fidelity is hardly effected by hyperparameter choice in the case of dSprites while for dMelodies it varies significantly. While sensitivity to hyperparameters is expected in neural networks, this is also one of the main reasons for evaluating methods on more than one dataset, preferably from multiple domains.

Some aspects of the dataset design, especially the na-

ture of the factors of variation, might have affected our experimental results. While the factors of variation in dSprites are continuous (except the shape attribute), those for dMelodies span different data-types (categorical, ordinal and binary). This might make other types of models (such as VQ-VAEs [49]) more suitable. Another consideration is that some factors of variation (such as the arpeggiation direction and rhythm) effect only a part of the data. However, the effect of this on the disentanglement performance needs further investigation since we get good performance for rhythm but poor performance for arpeggiation direction.

Unsupervised methods for disentanglement learning have their own limitations and some degree of supervision might actually be essential [27]. It is still unclear if it is possible to develop general domain-invariant disentanglement methods. Consequently, supervised and semi-supervised methods have been garnering more attention [10, 11, 19, 34]. The dMelodies dataset can also be used to explore such methods for music-based tasks. There has been some work recently in disentangling musical attributes such as rhythm and melodic contours which are considered important from an interactive music generation perspective [6,11,50]. Apart from the designed latent factors of variation, other low-level musical attributes such as rhythmic complexity and contours can also be computationally extracted using this dataset to meet task-specific requirements.

## 6. CONCLUSION

This paper addresses the need for more diverse modes of data for studying disentangled representation learning by introducing a new music dataset for the task. The *dMelodies* dataset comprises of more than 1 million data points of 2-bar melodies. The dataset is constructed based on fixed rules that maintain independence between different factors of variation, thus enabling researchers to use it for studying disentanglement learning. Benchmarking experiments conducted using popular disentanglement learning methods show that existing methods do not achieve performance comparable to those obtained on an analogous image-based dataset. This showcases the need for further research on domain-invariant algorithms for disentanglement learning.

## 7. ACKNOWLEDGMENT

## 8. REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, 2013.

[2] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *Advances in Neural Information Processing Systems 30 (NeurIPS)*, Long Beach, California, USA, 2017.

[3] W. Hsu, Y. Zhang, R. J. Weiss, Y. Chung, Y. Wang, Y. Wu, and J. R. Glass, "Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019*, Brighton, United Kingdom, 2019.

[4] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding," in *Advances in Neural Information Processing Systems 31 (NeurIPS)*, Montréal, Canada, 2018.

[5] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, "MIDI-VAE: Modeling Dynamics and Instrumentation of Music with Applications to Style Transfer," in *Proc. of 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018.

[6] R. Yang, D. Wang, Z. Wang, T. Chen, J. Jiang, and G. Xia, "Deep music analogy via latent representation disentanglement," in *Proc. of 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019.

[7] J. Jiang, G. G. Xia, D. B. Carlton, C. N. Anderson, and R. H. Miyakawa, "Transformer vae: A hierarchical model for structure-aware and interpretable music representation learning," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 516–520.

[8] Y.-J. Luo, K. Agres, and D. Herremans, "Learning disentangled representations of timbre and pitch for musical instrument sounds using gaussian mixture variational autoencoders," in *Proc. of 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019.

[9] Y.-N. Hung, I.-T. Chiang, Y.-A. Chen, and Y.-H. Yang, "Musical composition style transfer via disentangled timbre representations," in *Proc. of 28th International Joint Conference on Artificial Intelligence (IJCAI)*, Macao, China, 2020.

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

[11] A. Pati and A. Lerch, "Latent space regularization for explicit control of musical attributes," in *Proc. of ICML Workshop on Machine Learning for Music Discovery Workshop (ML4MD), Extended Abstract*, Long Beach, California, USA, 2019.

[12] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, "Disentangled multidimensional metric learning for music similarity," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 6–10.

[13] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner, "$\beta$-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," in *Proc. of 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[14] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud, "Isolating Sources of Disentanglement in Variational Autoencoders," in *Advances in Neural Information Processing Systems 31 (NeurIPS)*, Montréal, Canada, 2018.

[15] H. Kim and A. Mnih, "Disentangling by Factorising," in *Proc. of 35th International Conference on Machine Learning (ICML)*, Stockholm, Sweeden, 2018.

[16] A. Kumar, P. Sattigeri, and A. Balakrishnan, "Variational Inference of Disentangled Latent Concepts from Unlabeled Observations," in *Proc. of 5th International Conference of Learning Representations (ICLR)*, Toulon, France, 2017.

[17] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems 27 (NeurIPS)*, Montréal, Canada, 2014.

[18] N. Siddharth, B. Paige, J.-W. van de Meent, A. Desmaison, N. D. Goodman, P. Kohli, F. Wood, and P. H. Torr, "Learning disentangled representations with semi-supervised deep generative models," in *Advances in Neural Information Processing Systems 30 (NeurIPS)*, Long Beach, California, USA, 2017.

[19] F. Locatello, M. Tschannen, S. Bauer, G. Rätsch, B. Schölkopf, and O. Bachem, "Disentangling factors of variations using few labels," in *Proc. of 8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2020.

[20] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato, "Fader Networks:Manipulating Images by Sliding Attributes," in *Advances in Neural Information Processing Systems 30 (NeurIPS)*, Long Beach, California, USA, 2017, pp. 5967–5976.

[21] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, "Deep Convolutional Inverse Graphics Network," in *Advances in Neural Information Processing Systems 28 (NeurIPS)*, Montréal, Canada, 2015, pp. 2539–2547.

[22] C. Donahue, Z. C. Lipton, A. Balsubramani, and J. McAuley, "Semantically Decomposing the Latent Spaces of Generative Adversarial Networks," in *Proc. of 6th International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 2018.

[23] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner, "dSprites: Disentanglement testing Sprites dataset," https://github.com/deepmind/dsprites-dataset, 2017, last accessed, 2nd April 2020.

[24] C. Burgess and K. Hyunjik, "3d-shapes Dataset," https://github.com/deepmind/3d-shapes, Feb. 2020, last accessed, 2nd April 2020.

[25] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic, "Seeing 3D Chairs: Exemplar Part-based 2D-3D Alignment using a Large Dataset of CAD Models," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, Ohio, USA, 2014, pp. 3762–3769.

[26] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee, "Deep Visual Analogy-Making," in *Advances in Neural Information Processing Systems 28 (NeurIPS)*, Montréal, Canada, 2015, pp. 1252–1260.

[27] F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations," in *Proc. of 36th International Conference on Machine Learning (ICML)*, Long Beach, California, USA, 2019.

[28] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *Proc. of 2nd International Conference on Learning Representations (ICLR)*, Banff, Canada, 2014.

[29] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in $\beta$-VAE," in *NIPS Workshop on Learning Disentangled Representations*, Long Beach, California, USA, 2017.

[30] P. Rubenstein, B. Scholkopf, and I. Tolstikhin, "Learning Disentangled Representations with Wasserstein Auto-Encoders," in *Proc. of 6th International Conference on Learning Representations (ICLR), Workshop Track*, Vancouver, Canada, 2018.

[31] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 29 (NeurIPS)*, Barcelona, Spain, 2016, pp. 2172–2180.

[32] M. Connor and C. Rozell, "Representing closed transformation paths in encoded network latent space," in *Proc. of 34th AAAI Conference on Artificial Intelligence*, New York, USA, 2020.

[33] J. Engel, M. Hoffman, and A. Roberts, "Latent Constraints: Learning to Generate Conditionally from Unconditional Generative Models," in *Proc. of 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[34] D. Bouchacourt, R. Tomioka, and S. Nowozin, "Multi-Level Variational Autoencoder: Learning Disentangled Representations From Grouped Observations," in *Proc. of 32nd AAAI Conference on Artificial Intelligence*, New Orleans, USA, 2018.

[35] M. W. Gondal, M. Wüthrich, Ð. Miladinović, F. Locatello, M. Breidt, V. Volchkov, J. Akpo, O. Bachem, B. Schölkopf, and S. Bauer, "On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset," in *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019, pp. 15 740–15 751.

[36] M. Bretan and L. Heck, "Learning semantic similarity in music via self-supervision," in *Proc. of 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019.

[37] S. Gururani, A. Lerch, and M. Bretan, "A comparison of music input domains for self-supervised feature learning," in *Proc. of ICML Workshop on Machine Learning for Music Discovery Workshop (ML4MD), Extended Abstract*, Long Beach, California, USA, 2019.

[38] K. Choi, G. Fazekas, M. B. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proc. of 18th International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 141–149.

[39] S. Lattner, M. Dörfler, and A. Arzt, "Learning complex basis functions for invariant representations of audio," in *Proc. of 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019.

[40] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild," in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 3730–3738.

[41] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music," in *Proc. of 35th*

*International Conference on Machine Learning (ICML)*, Stockholm, Sweeden, 2018.

[42] G. Toussaint, "A Mathematical Analysis of African, Brazilian and Cuban Clave Rhythms," in *Proc. of BRIDGES: Mathematical Connections in Art, Music and Science*, 2002, pp. 157–168.

[43] M. S. Cuthbert and C. Ariza, "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," in *Proc. of 11th International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, The Netherlands, 2010.

[44] G. Hadjeres, F. Pachet, and F. Nielsen, "DeepBach: A steerable model for Bach chorales generation," in *Proc. of 34th International Conference on Machine Learning (ICML)*, Sydney, Australia, 2017, pp. 1362–1371.

[45] A. Pati, A. Lerch, and G. Hadjeres, "Learning to Traverse Latent Spaces for Musical Score Inpainting," in *Proc. of 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019.

[46] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved Variational Inference with Inverse Autoregressive Flow," in *Advances in Neural Information Processing Systems 29 (NeurIPS)*, Barcelona, Spain, 2016, pp. 4743–4751.

[47] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. of 3rd International Conference on Learning Representations (ICLR)*, San Diego, USA, 2015.

[48] K. Ridgeway and M. C. Mozer, "Learning Deep Disentangled Embeddings With the F-Statistic Loss," in *Advances in Neural Information Processing Systems 31 (NeurIPS)*, Montréal, Canada, 2018, pp. 185–194.

[49] A. van den Oord, O. Vinyals, and k. kavukcuoglu, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems 30 (NeurIPS)*, Long Beach, California, USA, 2017, pp. 6306–6315.

[50] T. Akama, "Controlling Symbolic Music Generation Based On Concept Learning From Domain Knowledge," in *Proc. of 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019.