# A variational autoencoder for music generation controlled by tonal tension

Rui Guo[1], Ivor Simpson[1], Thor Magnusson[1], Chris Kiefer[1], and Dorien Herremans[2]

[1] University of Sussex
[2] Singapore University of Technology and Design

**Abstract.** Many of the music generation systems based on neural networks are fully autonomous and do not offer control over the generation process. In this research, we present a controllable music generation system in terms of tonal tension. We incorporate two tonal tension measures based on the Spiral Array Tension theory into a variational autoencoder model. This allows us to control the direction of the tonal tension throughout the generated piece, as well as the overall level of tonal tension. Given a seed musical fragment, stemming from either the user input or from directly sampling from the latent space, the model can generate variations of this original seed fragment with altered tonal tension. This altered music still resembles the seed music rhythmically, but the pitch of the notes are changed to match the desired tonal tension as conditioned by the user.

**Keywords:** music generation, generative model, variational autoencoder, tonal tension

## 1 Introduction

Automatic music generation systems date back centuries. For instance, one famous example dates back to the 18th century, when people played the musical dice game to generate new music using a different (probabilistic) combination of musical bars (Herremans, Chuan, & Chew, 2017). Deep learning has caused a steep increase in popularity in this ancient field (Huang et al., 2019). However, giving the user high level control over the music that is being generated is still to be explored. When we allow for controllability, the resulting system can be used for narrative purposes such as film and game music. In this paper, we offer such a controllable system, with a focus on an important aspect of music: tension.

## 2 Related Work

A lot of work has been done on music generation systems in the last few decades. We refer the reader to Briot (2020); Herremans, Chuan, & Chew (2017) for a more complete overview. When we are able to control certain aspects of the generated music, we open up opportunities for co-creation between artist and

machine. When considering high level controls for music generation, one particular feature of interest is tonal tension, which is strongly related to emotion (Costa & Nese, 2020; Meyer, 1956). Controlling the tension gives us a way to control part of the affect in generated music. Tension in music often endows a feeling of cohesion, e.g. one might see two music phrases with the first one exhibiting a rise in tension followed by a phrase with a release in tension.

When it comes to generating music conditioned by tension, the existing research is limited. In a system by M. Farbood et al. (2007), a user-inputted harmony line guides the generated chord progressions. In another system, MorpheuS (Herremans & Chew, 2019) uses a variable neighborhood search optimization algorithm to generate music with specified tonal tension shapes. MorpheuS morphs the pitches of an existing input piece so that the resulting piece matches a given tonal tension shape, while preserving the musical structure (i.e. repeated patterns). Williams et al. (2017) proposed a neural network to adjust five musical features so as to generate music with the widest spread of stimuli in the valence-arousal space. Other related research uses long-short term memory (LSTM) networks to generate a tension profile first and uses this to condition the music generation (Verstraelen, 2019). The author reports the generated tension profile does not work as well as the template tension input. In Tan & Herremans (2020), Music FaderNets are used to change the amount of arousal in short, generated, musical fragments. In the current research, we aim to further improve the state-of-the-art by proposing to integrate two tonal tension measures into a variational autoencoder model to change the tonal tension of a seed fragment, while keeping the seed music rhythm mostly unchanged.

## 3 Method

### 3.1 Tension Measures

Musical tension may come from a variety of sources such as tempo, rhythm and timbre (M. M. Farbood, 2006; M. M. Farbood & Price, 2017; Herremans, Yang, et al., 2017). In this work, we focus on tonal tension, measured with a model based on the Spiral Array theory (Chew, 2002, 2014). The Spiral Array is a three-dimensional geometrical model which represents the tonal space. It consists of three spirals, one that represents the position of all pitches, one for chords, and one for keys. Within this geometrical space, a closer tonal distance results in a closer geometrical distance.

Herremans & Chew (2016) developed a model for tonal tension based on the Spiral Array. This model captures three aspects of tonal tension: cloud diameter, tensile strain, and cloud momentum. In this paper, we focus on the first two.

The **cloud diameter** captures how "tonally close" the pitches are in the tonal space, i.e. tension in terms of dissonance. To calculate this, we split our musical piece into windows (or clouds of notes). For each window, the cloud diameter represents the largest distance between notes of the cloud. A moving average window of one quarter note is used to smooth the diameter curve and prevent abrupt changes from one 16th note to another. The same applies for the tensile strain defined below.

We also calculate the **tensile strain** for each cloud of notes. This is defined by the distance between the geometric gravity point of all the pitches in the cloud (i.e. the center of effect). and the geometric position of the key of the piece.

### 3.2 Training data

A total of 7,289 MIDI files were selected from the LMD-matched dataset (Raffel, 2016) with the tag "pop" as calculated by (DuBreuil, 2020). The Midi-Miner tool (Guo et al., 2019) was used to extract both the melody and bass tracks from the MIDI files, and subsequently calculate two tonal tension measures described in the previous section. After that step, a total of 3,457 files are valid (i.e. contain both melody and bass track) to use as training data. The melody and bass track are both monophonic after preprocessing by Midi-Miner. A bass track is included to make the harmony progression clearer and the tension more perceptible. Each song was divided into several four-bar long fragments with both a melody and bass track, resulting in 44,900 four-bar long fragments. All of the songs were first transposed to C major or A minor (depending on their mode) and the key position $key_{pos}$ of all the songs is set as C major to calculate the tensile strain measure. It should be noted that the A minor $key_{pos}$ can also be used if the key of the song is given to the model, which is not the case here.

The input for the variational autoencoder (VAE) model is a $64 \times 89$ piano roll, whereby 89 is the feature dimension and 64 is the time dimension. One time step accounts for a 16th note, and 64 time steps equal four bars length in 4/4 meter. The 89 dimensions are comprised of four feature sets: melody pitches, melody onsets, bass pitches and bass onsets. The melody's pitches are represented as a 74-dim one-hot vector in the MIDI pitch range of $[24, 96]$. Pitches outside of that range will be omitted so as to focus on the most frequently occurring pitches. The last dimension of melody pitch vector marks a rest note. The melody's rhythm is also represented by a one-hot vector in which 1 represents a new note at that time step and 0 no new note. The bass' pitch is represented as a 13-dim one-hot vector which maps to 12 pitch classes plus a rest note (the last dimension). The bass' rhythm representation is similar to the melody's rhythm representation.

### 3.3 Model Details

The model is shown in Figure 1. A VAE (Kingma & Welling, 2013) encoder parameterised by $\phi$ is used to map the input piano roll $x$ to a latent space $z$. The decoder parameterised by $\theta$ maps $z$ to 6 separate outputs $[y_1..y_6]$. $[y1..y4]$ is the reconstruction of the input $x$ which includes four feature sets. The original VAE loss is the reconstruction loss minus the KL divergence of the unit Gaussian prior $p(z)$ and the posterior distribution $q_\phi(z|x)$. $\beta$ is used to change the balance of reconstruction loss and the Kullback–Leibler divergence loss.

$$\begin{aligned} \mathcal{L}_{vae} &= L_{rec}(\theta, \phi, x) - \beta D_{KL}(q_\phi(z|x)||p(z)) \\ &= \mathbb{E}_{q_\phi(z|x)}(\log p_\theta(x|z)) - \beta D_{KL}(q_\phi(z|x)||p(z)) \end{aligned} \tag{1}$$

In addition to the original VAE loss, the tension loss based on the predicted tension $y_5$ and $y_6$ are added, which are the tensile strain output and cloud diameter output respectively. $tensile\_strain(x)$ and $diameter(x)$ are tension measures calculated from the input $x$ by the spiral array theory. MSE is the mean square error loss. The tension loss is defined by:

$$\mathcal{L}_{tension} = MSE(tensile\_strain(x), y_5) + MSE(diameter(x), y_6) \tag{2}$$

The total loss for the model is hence given as:

$$\mathcal{L}_{total} = \mathcal{L}_{vae} + \mathcal{L}_{tension} \tag{3}$$

A two layer gated recurrent unit (GRU)(Cho et al., 2014) with 256 nodes is used in both the encoder and decoder. In the decoder, the latent variable is repeated for 64 timesteps to feed into the two GRU layers then followed by two dense layers for each of the six outputs. $\beta$ is set to 0.006 with KL annealing (Bowman et al., 2015) of an increase of 5e-7 for each batch until 0.006. The training, validation and test dataset split ratio is 0.8, 0.1, 0.1 respectively. The Adam optimiser is used with a start learning rate of 0.001. The model's test set loss is used for early stopping to mitigate overfitting.

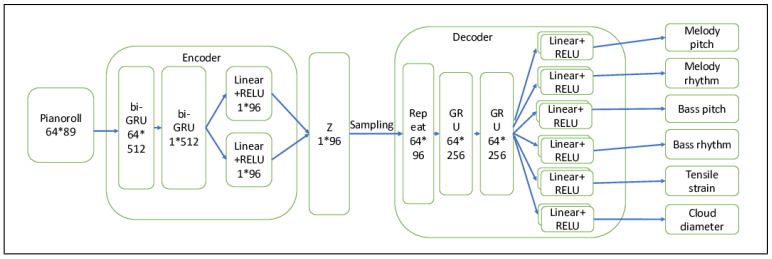


**Fig. 1.** Proposed model architecture. The input is a 64*89 piano roll with melody and bass tracks, and the output includes the reconstruction of the piano roll and the tension measures.

## 4 Experiment and evaluation

In this section, we first compare the loss of the proposed model with that of models that do not output tension measures. Then we explore the latent space and identify four latent vectors which allow us to control the tension direction or overall tension level of the generated music. The effect of those tension vectors is validated and finally, possible applications of those latent vectors are discussed.

### 4.1 Model loss comparison

We compare the proposed model with several variants. The baseline model simply reconstructs the input piano roll. Other variants of the model jointly learn to generate tension values by incorporating this information during the model

training. Each model was run five times and the result for the best run was used as shown in Table 1. Please note that additional loss terms were added for the cases in which tension was present. The KL loss is weighted by $\beta = 0.006$ in the total loss. When learning to predict tensile strain, the model bass pitch loss decreased. When predicting both the tension measures, the rhythm loss does not change much compared to the baseline model.

| Model | Total loss | Melody pitch | Melody rhythm | Bass pitch | Bass rhythm | Tensile | Diameter | KL |
|---|---|---|---|---|---|---|---|---|
| Baseline | 2.025 | 0.5335 | 0.1723 | 0.4120 | 0.1463 | NA | NA | 128 |
| Added ts | 2.0816 | 0.5457 | 0.1737 | 0.4080 | 0.1533 | 0.049 | NA | 126 |
| Added cd | 2.1942 | 0.5304 | 0.2998 | 0.41 | 0.1551 | NA | 0.1244 | 129 |
| Proposed | 2.2334 | 0.5405 | 0.1967 | 0.3994 | 0.1520 | 0.0454 | 0.1197 | 129 |

**Table 1.** Model loss with different conditional outputs. The baseline model only reconstructs the input piano roll, the other models generate not only the reconstructed input, but also additional features such as tensile strain (ts) and cloud diameter (cd). The proposed model learns to generate two tension measures along with reconstructing the input piano roll.

### 4.2 Identifying tension feature vectors

We are interested in using the latent space to manipulate the tension of the generated music. To understand how certain aspects of tonal tension are captured by the latent variables in our model, we have identified two types of latent feature vectors that we can use to change the music tension learned in the model to create specific manipulations of the generated music tension. One type allows us to control the evolution of the tension direction from the beginning of the 4 bar fragment to the end. The other controls the overall tension level throughout the 4 bars. We first assign tension class labels to our music fragments (upwards/downwards tensile strain/cloud diameter and high/low tensile strain/cloud diameter). To assign the first pair of labels, the correlation between the tension throughout the fragment and a straight line that goes through (0,0) to (1,1) with slope of 1 is calculated. For the second pair of labels, the 2-norm of difference of the tension value and a threshold value is calculated. Different thresholds for the correlation and 2-norm are used to select around 1,000 samples for each of the class labels. We adopt an approach similar to (Hou et al., 2017) to identify the feature vectors. Four latent feature vectors are identified: *tensile_strain_direction*, *tensile_strain_level*, *cloud_diameter_direction*, and *cloud_diameter_level*.

### 4.3 Validation of tension feature vectors

In this subsection, we evaluate the effectiveness of the four feature vectors identified in the previous section and how they interact with each other.

**Influence of each tension feature vector** We random sampled 10,000 $z$ from the latent space to conduct the following experiments. First we add the four

different tension feature vectors multiplied by a scaling factor to the original sampled $z$ and then decode from this new latent vector. We expect that: a) the tension of the generated music should change according to the definition of the chosen tension feature; b) the generated music should keep the original rhythm and not much changed compared with the original music as tonal tension is not influenced by rhythm.

Figure 2 shows the influence of adding different scaled tension direction feature vector to the latent space of each sample from the selected dataset. The tension upward ratio is calculated by the number of generated pieces with with upward tension divided by the total number of generated pieces. From the most left figures, we can see that the upward ratio of those tension measures increases with a larger scaled feature vector added to the original latent vector. These results confirm that adding this vector to the latent space has a significant influence on both the tensile strain and cloud diameter. In the middle graphs, we see that the change in melody rhythm by adding the *tensile_strain_direction* feature vector is small, while the *cloud_diameter_direction* change the melody rhythm is slightly higher. The bass rhythm F-score does not change much compared to the bass pitch accuracy and even less than the melody rhythm F-score.

From these results, we postulate that *tensile_strain_direction* changes the harmony progression and keeps the rhythm, while the *cloud_diameter_direction* adds/reduces notes with high tension to the melody resulting in a rhythm change. This was equally confirmed for the *tensile_strain_level* and *cloud_diameter_level* in Figure 5 in Appendix.

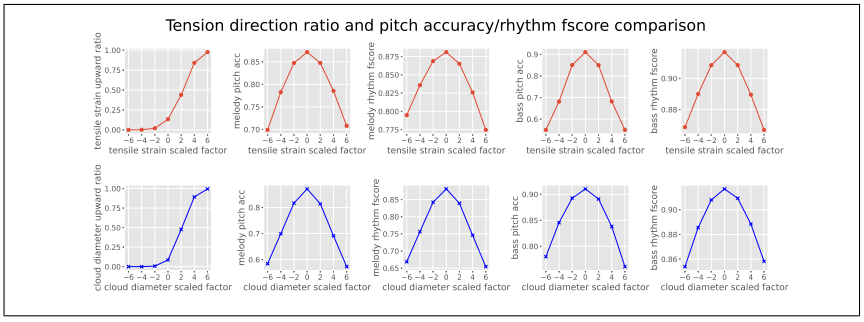


**Fig. 2.** Tension direction ratio and pitch accuracy/rhythm F-score comparison with scaled direction feature vectors. The pitch accuracy change is higher than the rhythm F-score change, and the melody rhythm is more affected by than the bass rhythm.

**Interaction among feature vectors** We also explored the interactions between different tension feature vectors so as to find out how much they can influence the tension output controlled by the other feature vector, as well as the function they play in the generation of music. In particular, the interaction between *tensile_strain_direction* and *cloud_diameter_direction* is explored in order to examine if applying the *tensile_strain_direction* vector can change the cloud diameter prediction and vice versa. In Figure 3, the *tensile_strain_direction* or

*cloud_diameter_direction* with varying scaling factors are added to the latent space $z$ and the tensile strain/cloud diameter upward ratio is shown. A change of the scaling factor for *tensile_strain_direction* does not affect the cloud diameter upward ratio a lot, however, the *cloud_diameter_direction* changes the tensile strain upward ratio more. This shows that applying a change in tensile strain does not necessarily change the cloud diameter of the output, and a change in the cloud diameter of the output is more likely to change the tensile strain, which can be explained by their definition (Section 3.1). A similar analysis of the tension level feature vector can be found in Figure 6 in Appendix.

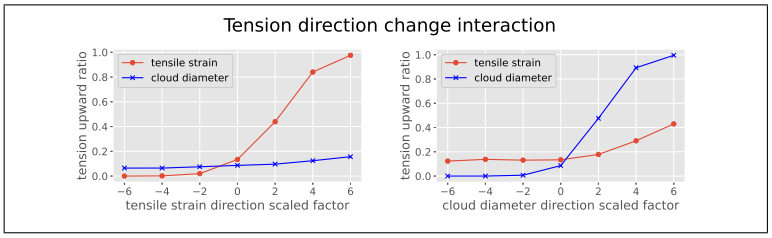


**Fig. 3.** Tension direction changes by varying the applied scaled *tensile_strain_direction* or *cloud_diameter_direction*. The upward ratio is calculated by the number of the output with tension upward shape divided by the total number of samples.

We also performed a number of experiments to compare the pitch distribution of the original music as well as the changed versions. In Figure 4, the pitch distribution of the original music and the music with added $6*$*tensile_strain_direction* is compared. The frequency of note C and G decreases and note A, E, D occurs in a much higher proportion in the music with tensile strain upward shape.

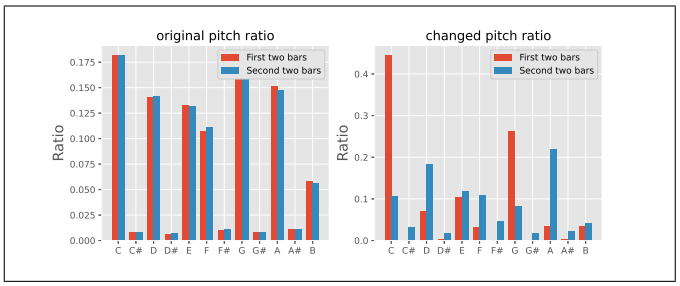


**Fig. 4.** The pitch distribution of the original music and music with upward tensile strain direction. The number of occurrences of both notes C and G drops and of notes A, E, D rises significantly in the second two bars of the tensile strain upward music compared to the original music.

In addition to the feature vector identified, our proposed method can change the tensile strain or cloud diameter into an arbitrary shape if a tension vector with a specific shape defined and identified. To validate this, a $\wedge$ shape tensile strain attribute vector was found. Adding this scaled vector to the latent space

allows us to generate music with a tensile strain shaped in the form of ╲╱ or ╱╲.

**Musical pieces** A demonstration of modifying the music tension by adding the *tensile_strain_direction* and *cloud_diameter_level* to the latent space is shown in Figure 7 in the Appendix. Although this model can only output 4 bar long music, it can be used to create different variations within a larger musical piece. For instance, we can generate the first 8 bars of music by adding the scaled *tensile_strain_direction* vector to the latent space of given input/sampled music, and then generate another 8 bars by first adding randomly scaled *cloud_diameter_level* to the first generated 8 bars' latent space vector. In this way, a variation of arbitrary length can be generated, which still sounds coherent to the other fragments as the same seed is used. The reader is invited to listen to some generated fragments and their tension figures at `https://ruiguo-bio.github.io/tension_vae.github.io` and explore our online colab notebook `https://github.com/ruiguo-bio/colab_tension_vae`.

## 5 Conclusion

We propose a generative VAE model to control the tonal tension in generated music. Through experiments, we have shown that our system can modify the tonal tension in generated music based on either a existing or a newly sampled seed fragment. We have successfully identified "tension feature vectors", which can achieve different transformations of the output music by adding the scaled tension feature vector to the latent space. In our experiments, we show the following: 1) By adding a scaled tension feature vector to the latest space variables of the seed music, we can generate music with increasing tension, both in terms of tensile strain as well as cloud diameter (depending which tension feature vector is used); 2) Using a similar method we can also increase or decrease the overall increase or decreased tension level 3) Additional tension feature vectors can be found that can realise different shapes of tension in the generated music. 4) The bass pitch loss becomes less by incorporating the tension strain feature into the model.

In the future work, we would like to explore the inclusion of more musical tension factors. Besides the two tension measures used in this research, musical tension is also related to rhythm, timbre and many other factors. Meaningful musical properties (Wang et al., 2020) can be fed into the system so as to further control the music generation. An interactive tool will be helpful to compose music by drawing tension. By using this tool to generate different variations to form a complete piece, we will examine how this affects the long-term structure and coherence of the generated music.

**Acknowledgements** This project was supported by the China Scholarship Council and MOE T2 grant no. MOE2018-T2-2-161.

# References


Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., & Bengio, S. (2015). Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Briot, J.-P. (2020). From artificial neural networks to deep learning for music generation - history, concepts and trends. *ArXiv*, *abs/2004.03586*.

Chew, E. (2002). The Spiral Array: An Algorithm for Determining Key Boundaries. In *Proceedings of the Second International Conference, ICMAI 2002* (pp. 18–31). Springer.

Chew, E. (2014). Mathematical and computational modeling of tonality. *AMC*, *10*(12), 141.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Costa, M., & Nese, M. (2020, 03). Perceived Tension, Movement, and Pleasantness in Harmonic Musical Intervals and Noises. *Music Perception*, *37*(4), 298-322.

DuBreuil, A. (2020). *Hands-on music generation with magenta: Explore the role of deep learning in music generation and assisted music composition.* Packt Publishing.

Farbood, M., Kaufman, H., & Jennings, K. (2007). Composing with hyperscore: An intuitive interface for visualizing musical structure. In *International computer music conference.*

Farbood, M. M. (2006). *A quantitative, parametric model of musical tension* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Farbood, M. M., & Price, K. C. (2017). The contribution of timbre attributes to musical tension. *The Journal of the Acoustical Society of America*, *141*(1), 419–427.

Guo, R., Herremans, D., & Magnusson, T. (2019). Midi Miner – A Python library for tonal tension and track classification. *arXiv:1910.02049*.

Herremans, D., & Chew, E. (2016). Tension ribbons: Quantifying and visualising tonal tension. In *2nd international conference on technologies for music notation and representation (tenor).* Cambridge, UK.

Herremans, D., & Chew, E. (2019). MorpheuS: Generating structured music with constrained patterns and tension. *IEEE Trans. on Affective Computing*, *10*(4), 510-523.

Herremans, D., Chuan, C.-H., & Chew, E. (2017, September). A functional taxonomy of music generation systems. *ACM Comput. Surv.*, *50*(5).

Herremans, D., Yang, S., Chuan, C.-H., Barthet, M., & Chew, E. (2017). Immaemo: A multimodal interface for visualising score-and audio-synchronised emotion annotations. In *Proceedings of the 12th international audio mostly conference on augmented and participatory sound and music experiences* (pp. 1–8).

Hou, X., Shen, L., Sun, K., & Qiu, G. (2017). Deep feature consistent variational autoencoder. In *Proc. of IEEE winter conference on applications of computer vision (wacv)* (pp. 1133–1141).

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Simon, I., Hawthorne, C., Shazeer, N., ... Eck, D. (2019). Music transformer: Generating music with long-term structure. In *International conference of learning representations.*

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv: 1312.6114*.

Meyer, L. (1956). *Emotion and meaning in music.* University of Chicago Press: Chicago.

Raffel, C. (2016). *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching* (Unpublished doctoral dissertation). Columbia University.

Tan, H., & Herremans, D. (2020). Music fadernets: Controllable music generation based on high-level features via low-level feature modelling. In *Proceedings of the int. society of music information retrieval (ismir).*

Verstraelen, V. (2019). *Generating Music with Coherent Harmonic Tension* (Unpublished master's thesis). Ghent University.

Wang, Z., Zhang, S., & Chen, X. (2020). Exploring Inherent Properties of the Monophonic Melody of Songs. *arXiv:2003.09287*.

Williams, D., Kirke, A., Miranda, E., Daly, I., Hwang, F., Weaver, J., & Nasuto, S. (2017). Affective calibration of musical feature sets in an emotionally intelligent music composition system. *ACM Transactions on Applied Perception (TAP)*, *14*(3), 1–13.

# 6 Appendix

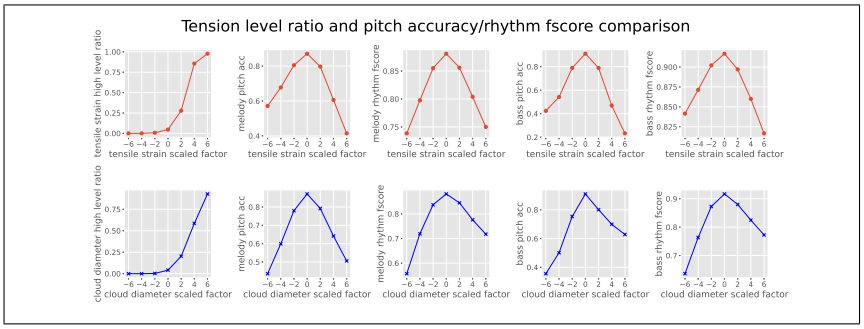


**Fig. 5.** Tension level and reconstruction accuracy change comparison by adding different scaled *tensile_strain_level* or *cloud_diameter_level* to the latent space of 10,000 random samples of the dataset. The high level ratio of tensile strain and diameter correlated with a larger scaling factor. The rhythm F-score is much less affected than the pitch accuracy vector, and *cloud_diameter_level* changes rhythm more than *tensile_strain_level*.

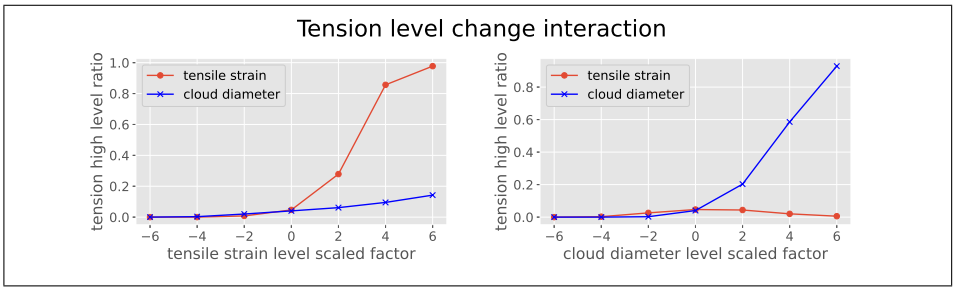


**Fig. 6.** Tension level changes by adding only one of the scaled *tensile_strain_level* or *cloud_diameter_level* to the latent space. The high level ratio is calculated by the number of the output with tension high level divided by the total number of samples. The scaled *cloud_diameter_level* changes the tensile strain level more than the the reverse, which can be explained by their definition.

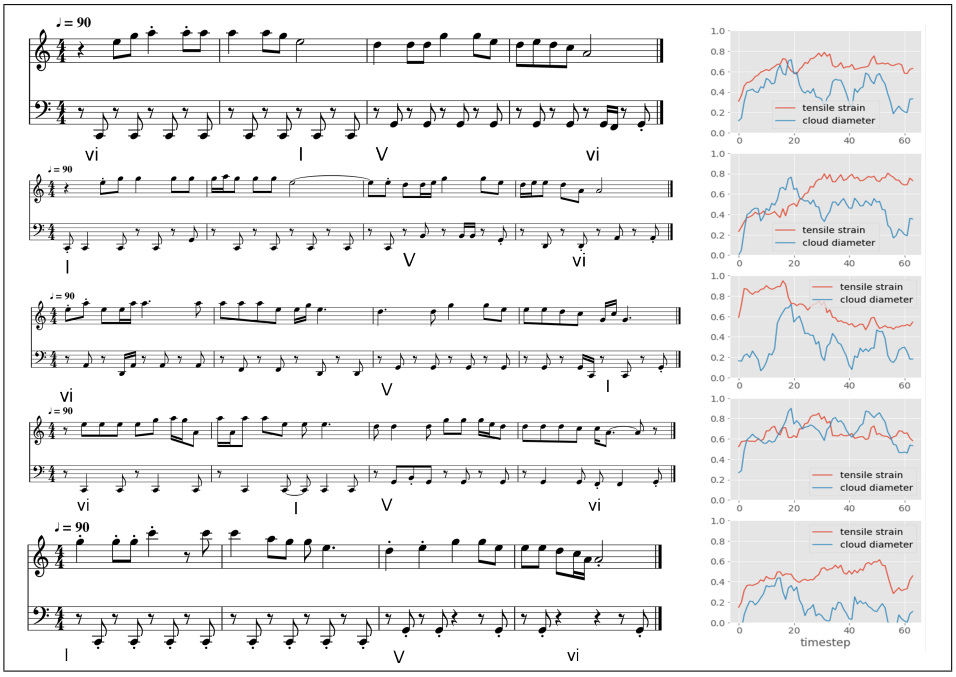


**Fig. 7.** Music variations generated by changing *tensile_strain_direction* and *cloud_diameter_level* and their tension predictions. The top staff is generated music by sampling the latent space, the second and third staff is the music changed by adding/subtracting $6 * tensile\_strain\_direction$ vector to the latent space of the original sample, and the fourth and fifth staff is music changed by adding/subtracting $3 * cloud\_diameter\_level$ vector to the latent space of the original sample. The corresponding tension for those five staff is right side of the figure. The music generated by adding $6 * tensile\_strain\_direction$ to the latent space changed its beginning harmony to C major and the harmony in the last two bar changes to A minor, and reverse is for the negative scaled $-6 * tensile\_strain\_direction$ music. The cloud diameter level up version has more notes increasing the tonal tension of the whole piece, and the diameter level down version has less tension by reducing the notes with higher tension. This shows the change of *cloud_diameter_level* add/reduce the density of notes to increase/decrease the cloud diameter, resulting in the change of rhythm. This phenomenon is the same when we change the *cloud_diameter_direction* before. Although the cloud diameter definition does not contain rhythm information explicitly, more or less notes with higher tension will change rhythm as result.