Pitch Class and Octave-Based Pitch Embedding Training Strategies for Symbolic Music Generation

Yuqiang Li[1] Shengchen Li[1] George Fazekas[2]

[1] Xi'an-Jiaotong Liverpool University
[2] Queen Mary University of London
yuqiang.li19@student.xjtlu.edu.cn

Abstract. This paper presents two strategies to prevent the pitch embeddings from being too close to the dataset characteristics so as to improve the pitch and pitch class distributions of generation. The first strategy is to switch the pitch representation from the MIDI number representation to an alternative representation that encodes a pitch into pitch class and octave, which forces musically similar pitches to share part of the embedding vectors. The second strategy freezes the pitch embeddings during training according to the proposed metrics that evaluate the quality of pitch embedding space, maintaining the robustness of the embedding obtained in the first strategy. The experiments show that, when both strategies are applied on the training in an auto-regressive melody generation task, the generated samples exhibit slightly improved pitch distribution but noticeably improved pitch class distribution, indicating the effectiveness of both strategies.

Keywords: Symbolic Music Generation, Word Embedding, Domain Knowledge

1 Introduction

The selection of an appropriate input music representation has been one of the key challenges in designing neural sequence models for symbolic generation, as multiple types of musical features must be serialized into sequences. Early MIDI event-like input representation (e.g. [26, 22]), suffered from the issues of being long and redundant to be handled by neural sequence models, and being implicit for models to reconstruct basic musical features (e.g. duration and metrical structures) [12]. Since then, solutions have been proposed to overcome these two problems, including applying constraints to the input representation using musical domain knowledge.

The REMI representation [13] uses the domain knowledge to recommend explicitly encoded durational and metrical features instead of MIDI-like note-on/note-off events, for a transformer to better capture durational and structural features on a sequential representation. The Compound Word representation (CPW) [11] improved the length limit and generation quality by shortening the input sequence length, based on the domain knowledge that tokens of the same type of musical features should be placed and treated similarly in the input. The recent Music Fundamental Embedding (MFE) [10] avoids a type of generation failure by treating pitch, duration and metric position features as numeric features to ensure consistency of the implied relative musical features in the embedding space. We notice that the general approach here is to apply domain knowledge constraints on the model input such that the explicitness can benefit the model in capturing specific features related to the domain knowledge.

Comparatively, pitch feature domain knowledge constraints are less researched in symbolic music generation, with most models using the simple MIDI number encoding for input. Also, the current generation systems still struggle with capturing slightly complicated pitch and harmony features. For instance, the generated music usually lack a clear key center, without clear harmonic tension and releases. However, in discriminative tasks, such as chord estimation, music style clustering and automatic harmonic analysis [17, 6, 31], the pitch class feature is used more often than MIDI pitch number, indicating its effectiveness in capturing pitch-based features. Therefore, we consider the concept of pitch class important in the generation task as well.

It is therefore hypothesized that using pitch class and octave for the pitch feature would improve the learned pitch representation and the generation pitch and pitch class distribution by preserve more pitch proximity in the embedding space. First, an auxiliary metric SLD is proposed for the evaluation of pitch embedding space. The hypothesis is then evaluated through two experiments. Experiment 1 tests whether pitch class and octave can improve the pitch distribution compared to MIDI number encoding. Experiment 2 is based on the results of experiment 1, testing if freezing the pitch embeddings according to the SLD metric maintains high pitch performance during training.

In experiment 1, a Transformer-XL model is trained for melody generation under the two different pitch encoding methods multiple times with different pitch-unrelated hyper-parameters. Results show that melodies sampled from the group of models using class-octave encoding have better pitch and pitch class distributions compared to the MIDI-number encoding group. Also, the evaluation of SLD metric on the corresponding metric space is consistent with the pitch and pitch class performance in generation.

Although the class-octave pitch encoding outperforms the other, it exhibit a behavior of deterioration over epochs which is more obvious than the MIDI number encoding. Correspondingly, the SLD metrics of most class-octave models are observed to have reached an local minima when the the model at the best pitch performance. Therefore, in experiment 2, the best model of Experiment 1 is trained multiple times but the pitch embeddings are frozen at different epochs, respectively. The results reveal that the models whose pitch embeddings are frozen near the local minima of the SLC metric has better performance over longer training.

The outcomes of the two experiments show the effectiveness of the pitch class and octave constraints on the pitch representation, which informed the development of two practical pitch training strategies presented in this paper.

The rest of the paper would begin by a brief review of the previous methodologies in Section 2, followed by proposed methods in Section 3. The experiment and results are discussed in 4 and 5.

2 Related Work

2.1 Pitch representations

The one-hot representation is a widely used pitch representation in the literature [1, 14]. It does not assume any pitch structure or proximity, as all the one-hot pitch vectors are equidistant. However, Mozer [24] argued that equidistant one-hot pitch vectors are problematic for music generation. Mozer proposed a novel pitch representation called PHCCCF based on the spiral model by [25] and psychoacoustic experiments in 1979 [15, 16]. In PHCCCF, pitch vectors are closer in euclidean distance if they are closer as perceived by ears. While Mozer's results has been able to learn some structure of diatonic scales [1], the psychoacoustic experiments were limited to isolated pitches without musical context, making the pitch representation less generalizable to music generation, where musical context is vital. In this work, we use the concept of pitch class and octave (both having been used in PHCCCF) but stick to the embedding representation learned through back propagation rather than static representation. To the best of our knowledge, PiRhDy [19] is the only recent music generation work that employed pitch class and octave, but the authors did not provide a comparison with the MIDI number encoding. Therefore, our work should be the first to compare these two different pitch encodings.

Alternative pitch encodings with domain knowledge have also been used in the symbolic music domain, but less frequently used in symbolic music generation. The tonnetz representation, proposed by Euler [7] in 1739, arranges pitch classes along major third, minor third, and perfect fifth dimensions. It has been successfully used for both feature extraction [3] and generative modelling in [20], but lacks smooth presentation of voice leading (namely the semitone or major second movements). Pitch classes are also effectively adopted in some discriminative tasks, e.g. chord classification[17] and style clustering [6, 31], but pitch-class-only representations ignore octave information needed for precise pitch description in generation tasks. This work, as a result, combines the pitch class and the octave feature for comparison with the MIDI-number encoding.

2.2 Word Embedding Training Strategies

Word embedding suffers from the representation degeneration problem [8], i.e. the embedding vector distribution is gradually distorted into a narrow cone shape, increasing the similarity of the word vectors with decreasing performance. [30] explained that rare token embeddings are pushed by their gradients away from the non-rare tokens, causing degeneration. Our observations, likewise, show that the pitch embedding space is biased towards the imbalanced pitch and pitch class distributions in the dataset. To prevent degeneration, [30] proposed a gradient gating strategy that freezes the rare tokens at early training, inspiring our strategy two.

Regarding poor numeracy performance of word embedding in language models [27], Gorishniy et al [9] demonstrated the advantages of using piecewise linear encoding (PLE) and sinusoidal activation functions (PAF) for numerical feature embedding. The FME [10], adopted an similar embedding scheme to embed pitch, duration and position features, ensuring the consistency of relative musical features such as intervals and durations in the embedding space. Instead of enhancing the pitch feature numeracy, this work studies the robustness brought by periodicity of pitch class and octave.

3 Methods

3.1 Pitch Encodings

In the commonly used music representations, (e.g. the MIDI event representation and the REMI representation), a pitch is encoded as a single token, indexed by the MIDI number, which we refer to as the MIDI number encoding. Being represented by one-hot vectors before embedding, the pitch vectors contain no domain knowledge information about frequency or pitch height as the dimensions are isotropic. This encoding is the baseline encoding.

The **class-octave** encoding is an alternative pitch encoding, which is less used in generation models [19] but more common in discriminative tasks as part of the input features [17, 6, 31, 18, 2]. It encodes a pitch to its pitch class (0 to 11) and the pitch octave number (0 to 9, if considering the highest valid MIDI pitch). If this encoding is used in a sequential music representation, a pitch is represented by two separate tokens in the sequence: the pitch class token ($p \mod 12$) followed by an octave token $\left\lfloor \frac{p}{12} \right\rfloor$. For instance, the pitch 60 (C4) is encoded into token `p0` and `o5`, corresponding to two different embedding vectors, respectively.

What is unique to about the class-octave encoding is its robustness to the slight pitch shifts, which manifests the proximity in listening experience before and after the shift. The pitch class-octave encoding is experimented to be compared with the baseline encoding because it has a much smaller vocabulary size (12 + *the number of octaves to be encoded*), which reduces the chances of over-parameterization. The pitch class-octave encoding also explicitly provides the constraints on the translational invariance for octaves ($\delta = 12$), i.e. all pitches that are octaves apart from each other share the same pitch class vector. Hence, it is expected to result in pitch embeddings that outperforms that of the MIDI number encoding.

3.2 Freezing Pitch Embedding in Early Training

The decreasing trend of the pitch performance over epochs suggests the possibility of deterioration of the pitch embeddings. As proposed in [30], freezing the rare token embeddings at early stage can alleviate the performance decline by preventing the embedding degeneration problem. In the music generation task of interest, most datasets have imbalanced pitch and pitch class distributions. Likewise, if the pitch embeddings are frozen at the optimal state, the resulting pitch performance is expected to be better. Hence, freezing the pitch embeddings at different epochs of training is investigated.

3.3 Metrics

This study employs two kinds of evaluative metrics to examine whether the proposed strategies effectively alleviate the pitch performance issue caused by imbalanced pitch (and pitch class) distribution in the dataset. The first kind evaluates the pitch embedding space itself and the other kind focuses on the generation quality, particularly about pitch.

Embedding Space Evaluation Metrics In order to obtain consistent embedding representations for intervals, (i.e. relative pitch features), the pitch vectors in the embedding space must follow certain constraints about intervals. According to FME [10], all the interval vectors $\{\mathbf{p}_{i+\delta} - \mathbf{p}_{j+\delta} | \delta \in \mathbb{Z}\}$ that represent the same pitch distance $|i - j|$ must have the same magnitude. As is not satisfied in most existing generation systems, this constraint is too strict. Therefore, we propose SLD, a metric that loosely measures the violation of such constraints. The Standard deviation of L2 Distances of pitch vectors[3] in the embedding space is defined as follows:

$$\mathrm{SLD}(\mathbf{P}) := \sum_{\delta=1}^{\delta_{\max}} \left[\underset{i=1..n-\delta}{\mathrm{Std}} \left(|\mathbf{p}_{i+\delta} - \mathbf{p}_i| \right) \right] . \tag{1}$$

This metric penalizes the differences in magnitudes for all pitch vectors whose difference vector represents the interval of δ semitone. $\delta_{\max}$ is empirically set to 24 here for two octaves, since intervals larger that are likely to have more different auditory experiences depending on the actual pitch height [23]. A better pitch embedding space is expected to have a lower SLD.

Generation Quality Evaluation Metrics (for Pitch) Admittedly, it is not practical to conduct a subjective listening test when the many models are experimented, also because the differences in the generated pitch distributions can be subtle to human audiences. Hence, objective metrics are adopted to evaluate the pitch performance in the generated samples. That is, the entropy of pitch class distribution H(PC), and for pitch H(P), as used in [28, 5]. These two metrics can accurately capture the lack of pitch diversity, or the repetition of very limited pitches when the H(P) is lower than that of the dataset, while H(PC) is an octave-agnostic version of H(P). The H(P) and H(PC) distributions of the test dataset are first approximated by Gaussian Kernel Density Estimation (KDE), and then compared to the KDEs of generation distributions. The overlapping area (OA) [29] between the fake and the true is used to score the generation quality, with the higher OA being the better.

4 Experiment Setup

4.1 Dataset

A cleaned version of the Wikifonia dataset[4] is used. Specifically, we only keep the songs with constant 4/4 time signatures. The training set (90%) contains 3,861 songs, and 429 songs for the test set. Note that quite a number of songs have modulations (key changes), so we do not do any kind of transposition for dataset balance as it will not completely balance the distribution. The imbalanced pitch class distribution is plotted in Figure 1. As can be seen, The frequent pitches come from the C major scales, the rest being rare in both subsets. The pitch class entropy H(PC) of the train set and dataset are

[3] The pitch vectors must be z-score transformed before SLD calculation, so as to eliminate the influence of the scaling along different dimensions

[4] http://www.wikifonia.org

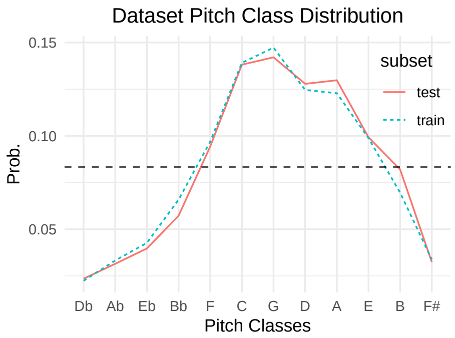


Fig. 1: Pitch Class Distributions of the Training Set and the Test Set

3.370 and 3.376 bits, respectively. Hence, the H(PC) of generated melodies should also be close to this dataset average value.

4.2 Data Representation

The input music representation resembles the REMI representation [13] because of the usage of duration, bar and position tokens. However, the features *chord*, *tempo* and *velocity* that are defined in REMI are ignored. In this work, the vocabulary set is formed by *pitch*, *octave* (if used), *duration*, *bar* and *position* tokens[5]. We also vary the beat resolution settings, allowing for the identification of consistent patterns in the model performance and a more robust analysis of the results.

4.3 Model and Training Specifications

A 4-layer transformer-XL network (proposed by [4] as used in [13, 28]) is employed to generates melodies in a next-token-prediction manner. The parameter size of the network is also cut down to 4M from the original, 12-layer model of 150M parameters in order to reduce the risks of over-fitting on such a small symbolic music dataset.

The experimented models in this work share most of the training hyper-parameters, including the cross-entropy loss, 0.9 to 0.1 train-test split, the optimizer AdamW [21], the learning rate 8e-4, batch size 32 and the number of epochs. Since the Transformer-XL architecture does not have a limit on the sequence length, the training sequence length is set to 1,024 tokens chunked into 8 segments of 128 tokens. The model is saved at the end of each epoch. Top-k sampling (at $k = 5$) and softmax temperature $\tau = 1.0$ is used for inference. For each model, 128 melodies are (unconditionally) sampled to evaluate the generation quality. However, only 512 tokens are sampled for each melody since longer sequences seem to be repetitive at the end.

[5] Miscellaneous tokens include a REST for silence that comes before duration, and PAD that pads the sequence

5 Results and Discussions

5.1 Experiment 1 - Comparison of Pitch Encodings

In this experiment, models are trained in pairs for token-by-token melody generation, teacher-forced. The two models in each pair share the common dataset, model architecture and only differ in the pitch encoding of the data representation: one uses the MIDI number encoding and the other uses the class-octave encoding. 24 pairs are set in order to compare the performance of two pitch encodings in different hyper-parameter configurations (e.g. the beat resolution).

Generation Result Metrics 128 melodies are sampled from each model for evaluation. The distributions of pitch entropy (H(P)) and pitch class entropy (H(PC)) are calculated for all the samples for each models. The overlapping area between the generation distribution KDE and the test dataset KDE are obtained to represent the performance of a model on a specific metric. Higher values are better.

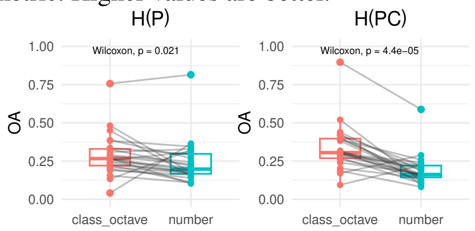
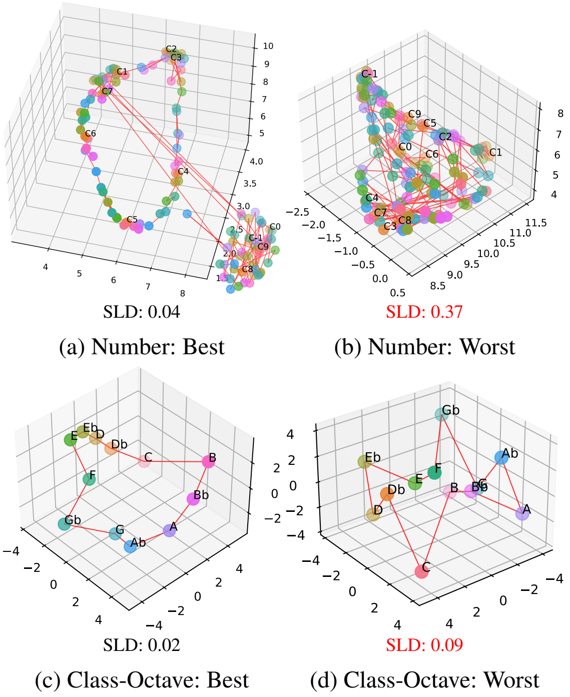


Fig. 2: Paired Box Plots of Model Performance Scores on Two Objective Metrics, Grouped by Pitch Encoding Used.

In Figure 2, each dot represents the generation metric distribution performance measured by OA of a model, grouped by the pitch encoding that the model uses. Line-connected dots are pairs of models that only differ in non-pitch hyper-parameters. The class-octave group on average outperforms the number group because of higher average performance. Paired Wilcoxcon tests on show that, such mean differences are significant ($p = 0.021$ for pitch, and $p = 4.4 \times 10^{-5}$ for pitch class).

Note that there is a considerable gap between the two models with the highest OA H(PC) , where the class-octave has model learned about 81% of the true H(PC) distribution while the number pitch model only learned around 60%. This suggests that the best performance on pitch class is dominated by class-octave encoding. However, the best performance of the class-octave group on pitch is slightly inferior to the number encoding, which is not surprising since the number-encoding pitch vectors have more parameters directly fitted to pitch distributions more accurately.

Embedding Space Metrics The best model and the worst model judged by OA H(PC) of each group are picked out, with their embedding space visualized in Figure 3. PCA is used to reduce the dimensionality from 32 two the 3 primary components with the largest variances for visualization purpose. However, the two number pitch visualizations are obtained from Uniform Manifold Approximation and Projection (UMAP)



(a) Number: Best

(b) Number: Worst

(c) Class-Octave: Best

(d) Class-Octave: Worst

Fig. 3: The pitch / pitch class vectors are plotted as points, colored by pitch classes. Red segments represent semitone relationship. Clear proximity between semitone pitch vectors is shown in embedding spaces with low SLD values, but less clear when SLD is high. The low SLDs are consistent with the actual best generation performance on both pitch and pitch class.

since the large number of pitch vectors are crowded in the PCA results and can be better clustered in the UMAP results. Note that when calculating SLD for class-octave embedding spaces, we first take the sum of octaves to all the pitch classes to restore the 128 pitches[6].

Overall, the visual differences between the best cases and the worst cases in Figure 3 suggest that pitch embedding space quality greatly contributes to the model performance on pitch performances. The two best cases demonstrate the success of modelling pitch distributions in early finished instances (because of other hyper-parameters e.g. beat resolution, that affects the model's learning ability before over-fitting happens), while the two worse cases show how the embedding space deteriorates over epochs.

[6] Summation is just one way to approximate the vector representation of the pitch feature, under the assumption that the embedding vectors are semantic and they follow the analogy property of word embedding. However, this can lead to different expected ranges of SLDs from that of the number pitch vectors, because the vector differences cancel out octave vectors if the two pitches are from the same octave. After all, this approximation error does not change the overall trend of SLD, which is of interest, since the error is only on the formula.

By comparing Figure 3a with 3c, and 3b with 3d, the class-octave encoding shows strong robustness and the embedding spaces suffer much less from the rare-token degeneration problem. That is, in MIDI number encoding, the lowest and the highest pitches are always rare tokens, regardless of the data augmentation methods such as random transposition. As a result, the rare pitch tokens are pushed into a cluster during the optimization (as demonstrated in [30]), resulting in worse pitch performance.

In contrast, for the class-octave encoding, the rare pitches are represented by only a few octave tokens (e.g. `o0` to `o3`, `o8` to `o9`), and their pitch classes are no different from that of the non-rare pitches because they share the pitch classes. The degeneration problem can still be seen on the visualization (3d), i.e. $\{D\flat, E\flat, F\sharp, A\flat, B\flat\}$ these rare pitch classes in this dataset (see Figure 1), are extruding out away from the non-rare pitch classes, causing worse pitch performance.

To conclude, the class-octave encoding is an underrated pitch encoding in the symbolic domain, outperforming the zero-domain-knowledge number encoding. It displays stronger robustness and interpretablity. In addition, the results show that a low SLD is a necessary condition of models being able to precisely capturing pitch and pitch class distributions.

5.2 Experiment 2 - Freezing Pitch Embedding Space at Different Stages of Training

This experiment validates the existence of the optimal state of the pitch embeddings by freezing the pitch embedding vectors at different epochs of training and tracking the their states (SLD and pitch performance).

The best set of non-pitch hyper-parameters[7] used in experiment 1 was adopted. Specifically, both the number encoding models and the class-octave model achieved lowest test set loss around epoch 5, which ended way earlier than other models who were trained for around 30 – 40 epochs, suggesting that further training the models is prone to decreasing performance.

However, as previously discussed, the SLD of the number encoding model did not decrease (or slightly decreased but rose very quickly at the beginning), which is a general problem regardless of most hyper-parameter settings. Conversely, the SLD of the best class-octave model decreased in the first 5 epochs and started to increase, reaching the best OA H(PC) at epoch 5, too. Hence, this experiment is dedicated to **class-octave** encoding where the SLD can decrease more noticeably at the beginning of training.

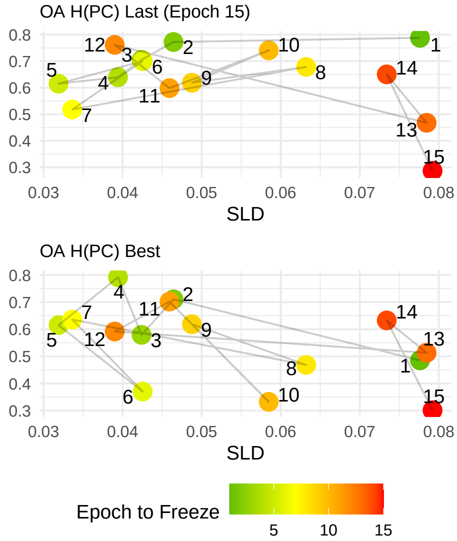
15 model instances were separately trained for 15 epochs from scratch (for better reproducibility), except that every time the pitch vectors are frozen 1 epoch later by zeroing out the gradients of pitch vectors. For each model, pitch embedding SLD was evaluated at each epoch until frozen. Note that the embedding vectors would still slightly change after frozen because of the existence of the projection layer between embedding output and the transformer input, which was not frozen as it is shared by all word vectors (including non-pitch vectors). In actual results, there was a very slight increase in the SLD for models but they did not change the ranking of different SLDs.

[7] Hyper-parameters: A beat resolution of 8 subdivisions per quarter note, a position grid similar to REMI [13] but each bar now has 8 × 4 = 32 positions instead of 16 used by the authors.

Metric Results Each of the trained models was evaluated at two states: **Best**, referring to the epoch of lowest test NLL loss; **Last**, at the end of epoch 15. The embedding metric SLD and pitch performance metric overlapping area OA H(PC) are listed in Table 4a. Arrows near the metrics indicated whether the maximum or the minimum is desired.

Freezing Epoch	SLD Best ↓	H(PC) Last ↑	H(PC) Best ↑	Best Epoch
1	0.078	**0.788**	0.485	11
2	0.046	0.772	0.710	7
3	0.042	0.705	0.579	13
4	**0.039**	0.639	**0.792**	4
5	0.032	0.615	0.615	14
6	0.043	0.700	0.370	11
7	0.034	0.518	0.636	11
8	0.063	0.678	0.469	11
9	0.049	0.619	0.619	14
10	0.058	0.741	0.332	4
11	0.046	0.598	0.702	9
12	**0.039**	0.761	0.592	12
13	0.078	0.468	0.513	9
14	0.073	0.650	0.633	12
15	0.079	0.286	0.301	4

(a) The Embedding SLDs and Generation OA H(PC)s of 15 Models. SLDs are measured at model reaching lowest test error, not necessarily before or after the freezing moment.



(b) The plot traces the pair of both OA H(PC) and SLD over epochs of freezing. Higher positions stand for better pitch class performance while lefter positions for better embedding quality.

Fig. 4: Models with Pitch Vectors Frozen at Different Epochs

The 15 models display an interesting 3-phase training dynamics every 5 epochs.

- In phase 1, when frozen before epoch 5, the pitch embedding SLD decreased. The OA H(PC) of the resulting best models climbed up, reaching the maximal performance 0.79 at epoch 4. Models 1 to 4 at epoch 15 have OA H(PC) higher than 0.6, suggesting that the pitch performance is maintained in longer training.
- In phase 2, freezing happened between epoch 6 and 9, when the SLD was higher. Both *last* and *best* OA H(PC) slightly decreased, especially for the best models the OA H(PC) dropped below 0.4.
- In phrase 3, from epoch 10 onward, the pitch performance became much more unstable. The SLD for around epoch 13 to 15 quickly increases, with decreasing OA H(PC). Also notice that the "best epoch" numbers below the dashed line in Table 4a are all smaller than the freezing epoch, indicating over-fitting if pitch embeddings were frozen later than epoch 10. Conversely, if freezing happened before epoch 10, all except model 4 could last for longer training.

The results first suggest that it is effective to freeze pitch embeddings at low SLD level to retain pitch performance at higher levels for both the best and the last models. In addition, this strategy offers the benefit of being able to train a properly frozen embedding longer before the model is over-fitted.

6 Conclusion

This paper presents two strategies aiming at improving the pitch performance of a symbolic music generation model. Both involve incorporating domain knowledge that restricts the pitch representation in terms of feature encoding and feature representation, which effectively alleviate the problem of pitch performance deterioration. Strategy 1 introduces the concept of octave and pitch class, which preserves more pitch proximity than the MIDI number encoding while strategy 2 maintains the advantage of strategy 1 according to the proposed SLD, a loose version of translational invariance property. This study and also calls attention to the generation performance issues related to lack of prior knowledge when designing music generation models. In futural works, the authors plan to generalize such strategies for more advanced pitch features, such as intervals and harmony, or other non-pitch musical features with similar constraints.

References

1. Briot, J.P., Hadjeres, G., Pachet, F.D.: Deep Learning Techniques for Music Generation – A Survey. arXiv:1709.01620 [cs] (Aug 2019)
2. Chawin, D., Rom, U.B.: Sliding-Window Pitch-Class Histograms as a Means of Modeling Musical Form. Transactions of the International Society for Music Information Retrieval **4**(1), 223–235 (Dec 2021). https://doi.org/10.5334/tismir.83
3. Chuan, C.H., Herremans, D.: Modeling Temporal Tonal Relations in Polyphonic Music Through Deep Networks With a Novel Image-Based Representation. Proceedings of the AAAI Conference on Artificial Intelligence **32**(1) (Apr 2018). https://doi.org/10.1609/aaai.v32i1.11880
4. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context (Jun 2019). https://doi.org/10.48550/arXiv.1901.02860
5. Dong, H.W., Chen, K., McAuley, J., Berg-Kirkpatrick, T.: MusPy: A Toolkit for Symbolic Music Generation (Aug 2020)
6. Ens, J., Pasquier, P.: Quantifying Musical Style: Ranking Symbolic Music based on Similarity to a Style (Mar 2020)
7. Euler, L.: Tentamen novae theoriae musicae: ex certissimis harmoniae principiis dilucide expositae. Saint Petersburg Academy (1739)
8. Gao, J., He, D., Tan, X., Qin, T., Wang, L., Liu, T.: Representation Degeneration Problem in Training Natural Language Generation Models. In: International Conference on Learning Representations (Feb 2022)
9. Gorishniy, Y., Rubachev, I., Babenko, A.: On Embeddings for Numerical Features in Tabular Deep Learning (Mar 2022)
10. Guo, Z., Kang, J., Herremans, D.: A Domain-Knowledge-Inspired Music Embedding Space and a Novel Attention Mechanism for Symbolic Music Modeling (Dec 2022)
11. Hsiao, W.Y., Liu, J.Y., Yeh, Y.C., Yang, Y.H.: Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs (Jan 2021)
12. Huang, C.Z.A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A.M., Hoffman, M.D., Dinculescu, M., Eck, D.: Music Transformer. In: International Conference on Learning Representations (2019)

13. Huang, Y.S., Yang, Y.H.: Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 1180–1188. Association for Computing Machinery, NY, USA (Oct 2020)
14. Ji, S., Luo, J., Yang, X.: A Comprehensive Survey on Deep Music Generation: Multi-level Representations, Algorithms, Evaluations, and Future Directions. arXiv:2011.06801 [cs, eess] (Nov 2020)
15. Krumhansl, C.L.: The Psychological Representation of Musical Pitch in a Tonal Context. Cognitive Psychology **11**(3), 346–374 (Jul 1979). https://doi.org/10.1016/0010-0285(79)90016-1
16. Krumhansl, C.L., Kessler, E.J.: Tracing the Dynamic Changes in Perceived Tonal Organization in a Spatial Representation of Musical Keys. Psychological Review **89**(4), 334–368 (1982)
17. Laden, B., Keefe, D.H.: The Representation of Pitch in a Neural Net Model of Chord Classification. Computer Music Journal **13**(4), 12–26 (1989). https://doi.org/10.2307/3679550
18. Lazzari, N., Poltronieri, A., Presutti, V.: Pitchclass2vec: Symbolic Music Structure Segmentation with Chord Embeddings. Workshop on Artificial Intelligence and Creativity p. 17 (Nov 2022)
19. Liang, H., Lei, W., Chan, P.Y., Yang, Z., Sun, M., Chua, T.S.: PiRhDy: Learning Pitch-, Rhythm-, and Dynamics-aware Embeddings for Symbolic Music. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 574–582 (Oct 2020). https://doi.org/10.1145/3394171.3414032
20. Lieck, R., Moss, F.C., Rohrmeier, M.: The Tonal Diffusion Model. Transactions of the International Society for Music Information Retrieval **3**(1), 153–164 (Oct 2020). https://doi.org/10.5334/tismir.46
21. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization (Jan 2019). https://doi.org/10.48550/arXiv.1711.05101
22. Meade, N., Barreyre, N., Lowe, S.C., Oore, S.: Exploring Conditioning for Generative Music Systems with Human-Interpretable Controls. arXiv:1907.04352 [cs, eess] (Aug 2019)
23. Moore, B.C.J.: An Introduction to the Psychology of Hearing. BRILL (2012)
24. Mozer, M.C.: Connectionist Music Composition Based On Melodic, Stylistic and psychophysical Constraints. Computer Science Technical Reports (476) (May 1990)
25. Shepard, R.N.: Geometrical approximations to the structure of musical pitch. Psychological Review **89**, 305–333 (1982). https://doi.org/10.1037/0033-295X.89.4.305
26. Simon, I., Oore, S.: Performance rNN: Generating music with expressive timing and dynamics. https://magenta.tensorflow.org/performance-rnn (2017)
27. Wallace, E., Wang, Y., Li, S., Singh, S., Gardner, M.: Do NLP Models Know Numbers? Probing Numeracy in Embeddings (Sep 2019)
28. Wu, S.L., Yang, Y.H.: The Jazz Transformer on the Front Line: Exploring the Shortcomings of AI-composed Music through Quantitative Measures (Aug 2020)
29. Yang, L.C., Chou, S.Y., Yang, Y.H.: MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation. arXiv:1703.10847 [cs] (Jul 2017)
30. Yu, S., Song, J., Kim, H., Lee, S.m., Ryu, W.J., Yoon, S.: Rare Tokens Degenerate All Tokens: Improving Neural Text Generation via Adaptive Gradient Gating for Rare Token Embeddings (Jun 2022). https://doi.org/10.48550/arXiv.2109.03127
31. Yust, J., Lee, J., Pinsky, E.: A Clustering-Based Approach to Automatic Harmonic Analysis: An Exploratory Study of Harmony and Form in Mozart's Piano Sonatas. Transactions of the International Society for Music Information Retrieval **5**(1), 113–128 (Oct 2022). https://doi.org/10.5334/tismir.114