BEBOPNET: DEEP NEURAL MODELS FOR PERSONALIZED JAZZ IMPROVISATIONS

Shunit Haviv Hakimi* **Nadav Bhonker*** **Ran El-Yaniv**
Computer Science Department
Technion – Israel Institute of Technology
shunithaviv@gmail.com, nadavbh@gmail.com, rani@cs.technion.ac.il

ABSTRACT

A major bottleneck in the evaluation of music generation is that music appreciation is a highly subjective matter. When considering an average appreciation as an evaluation metric, user studies can be helpful. The challenge of generating personalized content, however, has been examined only rarely in the literature. In this paper, we address generation of personalized music and propose a novel pipeline for music generation that learns and optimizes user-specific musical taste. We focus on the task of symbol-based, monophonic, harmony-constrained jazz improvisations. Our personalization pipeline begins with BebopNet, a music language model trained on a corpus of jazz improvisations by Bebop giants. BebopNet is able to generate improvisations based on any given chord progression [1]. We then assemble a personalized dataset, labeled by a specific user, and train a user-specific metric that reflects this user's unique musical taste. Finally, we employ a personalized variant of beam-search with BebopNet to optimize the generated jazz improvisations for that user. We present an extensive empirical study in which we apply this pipeline to extract individual models as implicitly defined by several human listeners. Our approach enables an objective examination of subjective personalized models whose performance is quantifiable. The results indicate that it is possible to model and optimize personal jazz preferences and offer a foundation for future research in personalized generation of art. We also briefly discuss opportunities, challenges, and questions that arise from our work, including issues related to creativity.

1. INTRODUCTION

Since the dawn of computers, researchers and artists have been interested in utilizing them for producing different forms of art, and notably for composing music [1]. The explosive growth of deep learning models over the past several years has expanded the possibilities for musical generation, leading to a line of work that pushed forward the state-of-the-art [2–6]. Another recent trend is the development and offerings of consumer services such as Spotify, Deezer and Pandora, aiming to provide personalized streams of existing music content. Perhaps the crowning achievement of such personalized services would be for the content itself to be generated explicitly to match each individual user's taste. In this work we focus on the task of generating user personalized, monophonic, symbolic jazz improvisations. To the best of our knowledge, this is the first work that aims at generating personalized jazz solos using deep learning techniques.

The common approach for generating music with neural networks is generally the same as for language modeling. Given a context of existing symbols (e.g., characters, words, music notes), the network is trained to predict the next symbol. Thus, once the network learns the distribution of sequences from the training set, it can generate novel sequences by sampling from the network output and feeding the result back into itself. The products of such models are sometimes evaluated through user studies (crowd-sourcing). Such studies assess the quality of generated music by asking users their opinion, and computing the mean opinion score (MOS). While these methods may measure the overall quality of the generated music, they tend to average-out evaluators' personal preferences. Another, more quantitative but rigid approach for evaluation of generated music is to compute a metric based on musical theory principles. While such metrics can, in principle, be defined for classical music, they are less suitable for jazz improvisation, which does not adhere to such strict rules.

To generate personalized jazz improvisations, we propose a framework consisting of the following elements: (a) BebopNet: jazz model learning; (b) user preference elicitation; (c) user preference metric learning; and (d) optimized music generation via planning.

As many jazz teachers would recommend, the key to attaining great improvisation skills is by studying and emulating great musicians. Following this advice, we train BebopNet, a harmony-conditioned jazz model that composes entire solos. We use a training dataset of hundreds of professionally transcribed jazz improvisations performed by saxophone giants such as Charlie Parker, Phil Woods and

[1] Supplementary material and numerous MP3 demonstrations of jazz improvisations of jazz standards and pop songs generated by BebopNet are provided in https://shunithaviv.github.io/bebopnet.

Attribution: Shunit Haviv Hakimi, Nadav Bhonker, and Ran El-Yaniv, "BebopNet: Deep Neural Models for Personalized Jazz Improvisations", in *Proc. of the 21st Int. Society for Music Information Retrieval Conf.,* Montréal, Canada, 2020.

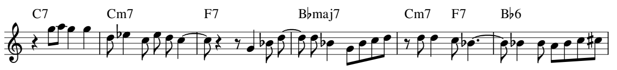


Figure 1. A short excerpt generated by BebopNet.

Cannonball Adderley (see details in Section 4.1.1). In this dataset, each solo is a monophonic note sequence given in symbolic form (MusicXML) accompanied by a synchronized harmony sequence. After training, BebopNet is capable of generating high fidelity improvisation phrases (this is a subjective impression of the authors). Figure 1 presents a short excerpt generated by BebopNet.

Considering that different people have different musical tastes, our goal in this paper is to go beyond straightforward generation by this model and optimize the generation toward personalized preferences. For this purpose, we determine a user's preference by measuring the level of their satisfaction throughout the solos using a digital variant of continuous response interface (CRDI) [7]. This is accomplished by playing, for the user, computer-generated solos (from the jazz model) and recording their good/bad feedback in real time throughout each solo. Once we have gathered sufficient data about the user's preferences, consisting of two aligned sequences (for the solos and feedback), we train a user preference metric in the form of a recurrent regression model to predict this user's preferences. A key feature of our technique is that the resulting model can be evaluated *objectively* using hold-out user preference sequences (along with their corresponding solos). A big hurdle in accomplishing this step is that the signal elicited from the user is inevitably extremely noisy. To reduce this noise, we apply selective prediction techniques [8, 9] to distill cleaner predictions from the user's preference model. Thus, we allow this model to abstain whenever it is not sufficiently confident. The fact that it is possible to extract a human continuous response preference signal on musical phrases and use it to train (and test) a model with non-trivial predictive capabilities is interesting in itself (and new, to the best of our knowledge).

Equipped with a personalized user preference metric (via the trained model), in the last stage we employ a variant of beam-search [10], to generate optimized jazz solos from BebopNet. For each user, we apply the last three stages of this process where the preference elicitation stage takes several hours of tagging per user. We applied the proposed pipeline on four users, all of whom are amateur jazz musicians. We present numerical analysis of the results showing that a personalized metric can be trained and then used to optimize solo generation.

To summarize, our contributions include: (1) a useful monophonic neural model for general jazz improvisation within any desired harmonic context; (2) a viable methodology for eliciting and learning high resolution human preferences for music; (3) a personalized optimization process of jazz solo generation; and (4) an objective evaluation method for subjective content and plagiarism analysis for the generated improvisations.

2. RELATED WORK

Many different techniques for algorithmic musical composition have been used over the years. For example, some are grammar-based [11], rule-based [1, 12], use Markov chains [13–15], evolutionary methods [16, 17] or neural networks [18–20]. For a comprehensive summary of this broad area, we refer the reader to [21]. Here we confine the discussion to closely related works that mainly concern jazz improvisation using deep learning techniques over symbolic data. In this narrower context, most works follow a generation by prediction paradigm, whereby a model trained to predict the next symbol is used to greedily generate sequences. The first work on blues improvisation [22] straightforwardly applied long short-term memory (LSTM) networks on a small training set. While their results may seem limited at a distance of nearly two decades [2], they were the first to demonstrate long-term structure captured by neural networks.

One approach to improving a naïve greedy generation from a jazz model is by using a mixture of experts. For example, Franklin et al. [23] trained an ensemble of neural networks were trained, one specialized for each melody, and then selected from among them at generation time using reinforcement learning (RL) utilizing a handcrafted reward function. Johnson et al. [24] generated improvisations by training a network consisting of two experts, each focusing on a different note representation. The experts were combined using the technique of product of experts [25] [3]. Other remotely related non-jazz works have attempted to produce context-dependent melodies [2, 3, 5, 26–30].

A common method for collecting continuous measurements from human subjects listening to music is the continuous response digital interface (CRDI), first reported by [7]. CRDI has been successful in measuring a variety of signals from humans such as emotional response [31], tone quality and intonation [32], beauty in a vocal performance [33], preference for music of other cultures [34] and appreciation of the aesthetics of jazz music [35]. Using CRDI, listeners are required to rate different elements of the music by adjusting a dial (which looks similar to a volume control dial present on amplifiers).

3. PROBLEM STATEMENT

We now state the problem in mathematical terms. We denote an input $x_t = (s_t, c_t)$ consisting of a note s_t and its context c_t. Each note $s_t \in \mathcal{S}$, in turn, consists of a pitch and a duration at index t and $\mathcal{S}$ represents a predefined set of pitch-duration combinations (i.e., notes). The context $c_t \in \mathcal{C}$ represents the chord that is played with note s_t, where $\mathcal{C}$ is the set of all possible chords. The context may contain additional information such as the offset of the note within a measure (see details in Section 4). Let $\mathcal{D}$ denote a training dataset consisting of M solos. Each

[2] Listen to their generated pieces at www.iro.umontreal.ca/~eckdoug/blues/index.html.

[3] Listen to the generated solos at www.cs.hmc.edu/~keller/jazz/improvisor/iccc2017/

solo is a sequence $X_\tau = x_1 \cdots x_\tau \in (\mathcal{S} \times \mathcal{C})^\tau$ of arbitrary length τ. In our work, these are the aforementioned jazz improvisations.

We define a context-dependent jazz model f_θ (Eq. 1), as the estimator of the probability of a note s_t given the sequence of previous inputs X_{t-1} and the current context c_t, where θ are the parameters of the model. This is similar to a human jazz improviser who is informed of the chord over which his next note will be played.

$$f_\theta(X_{t-1}, c_t) = Pr(s_t | X_{t-1}, c_t) \qquad (1)$$

For any solo X_τ, we also consider an associated sequence of annotation $Y_\tau = y_1 \cdots y_\tau \in \mathcal{Y}^\tau$. An annotation $y_t \in \mathcal{Y}$ represents the quality of the solo up to point t by some metric. In our case, y_t may be a measure of preference as indicated by a user or a score measuring harmonic compliance. Let $\widetilde{\mathcal{D}}$ denote a training dataset consisting of N solos. Each solo X_τ of arbitrary length τ is labeled with a sequence Y_τ. Given $\widetilde{\mathcal{D}}$, we define a metric g_ϕ (Eq. 2) to predict y_τ given a sequence of inputs X_τ. g_ϕ is the user-preference model and ϕ are the learned parameters.

$$\hat{y}_\tau = g_\phi(X_\tau) \qquad (2)$$

We denote by ψ a function that is used to sample notes from f_θ to generate solos. In our case, this will be our beam-search variant. The objective here is to train viable models, f_θ and g_ϕ, and then to use ψ to sample solos from f_θ while maximizing g_ϕ.

4. METHODS

In this section we describe the methods used and implementation details of our personalized generation pipeline.

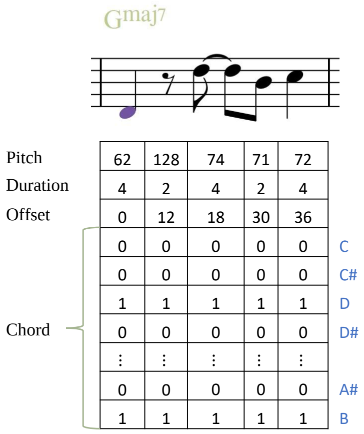
4.1 BebopNet: Jazz Model Learning

In the first step of our pipeline, we use supervised learning to train BebopNet, a context-dependent jazz model f_θ from a given corpus of transcribed jazz solos.

4.1.1 *Dataset and music representation*

Our corpus $\mathcal{D}$ consists of 284 professionally transcribed solos of (mostly) Bebop saxophone players of the early 20th century. These are Charlie Parker, Sonny Stitt, Cannonball Adderley, Dexter Gordon, Sonny Rollins, Stan Getz, Phil Woods and Gene Ammons. We consider only solos that are in 4/4 metre and include chords in their transcription. The solos are provided in musicXML format. As opposed to MIDI, this format allows the inclusion of chord symbols[4]. We represent notes using a representation method inspired by sheet music (see Figure 2).

Pitch The pitch is encoded as a one-hot vector of size 129. Indices 0—127 match the pitch range of the MIDI standard.[5] Index 128 corresponds to the *rest* symbol.



Pitch	62	128	74	71	72	
Duration	4	2	4	2	4	
Offset	0	12	18	30	36	
	0	0	0	0	0	C
	0	0	0	0	0	C#
	1	1	1	1	1	D
Chord	0	0	0	0	0	D#
	⋮	⋮	⋮	⋮	⋮	
	0	0	0	0	0	A#
	1	1	1	1	1	B

Figure 2. An example of a measure in music notation and its vector representation. Integers are converted to one-hot representations.

Duration The duration of each note is encoded using a one-hot vector consisting of all the existing durations in the dataset. Durations smaller than $1/24$ are removed.

Offset The offset of the note lies within the measure and is quantized to 48 “ticks” per (four-beat) measure. This corresponds to a duration of 1/12 of a beat. This is similar to the learned positional-encoding used in translation [37].

Chord The chord is represented by a four-hot vector of size 12, representing the 12 possible pitch classes to appear in a chord. As common in jazz music, unless otherwise noted, we assume that chords are played using their 7th form. Thus, the chord pitches are usually the 1st, 3rd, 5th, and 7th degrees of the root of the chord. This chord representation allows the flexibility of representing rare chords such as sixth, diminished and augmented chords.

4.1.2 *Network Architecture*

BebopNet, as many language models, can be implemented using different architectures such as recurrent neural networks (RNNs), convolutional networks (CNNs) [5, 26, 38] or attention-based models [39]. BebopNet contains a three-layer LSTM network [40]. Recent promising results with attention based models enabled us to improve BebopNet by replacing the LSTM with Transformer-XL [41]. The architecture of the network used to estimate f_θ is illustrated in Figure 3. The network’s input x_t includes the note s_t (pitch and duration) and context c_t (offset and chord). The pitch, duration and offset are each represented by learned embedding layers. The chord is encoded by using the embedding of the pitches comprising it. While notes at different octaves have different embeddings, the chord pitch embeddings are always taken from the octave in which most notes in the dataset reside. This embedded vector is passed to the LSTM network. The LSTM output is then passed to two heads. Each head consists of two fully-connected layers with a sigmoid activation in-between. The output of the first layer is the same size as the embedding of the pitch (or duration), and the second output size is the number of possible pitches (or durations).

[4] The solos were purchased from SaxSolos.com [36]; we are thus unable to publish them. Nevertheless, in the supplementary material we provide a complete list of solos used for training, which are available from the above vendor.

[5] The notes appearing in the corpus all belong to a much smaller range; however, the MIDI range standard was maintained for simplicity.

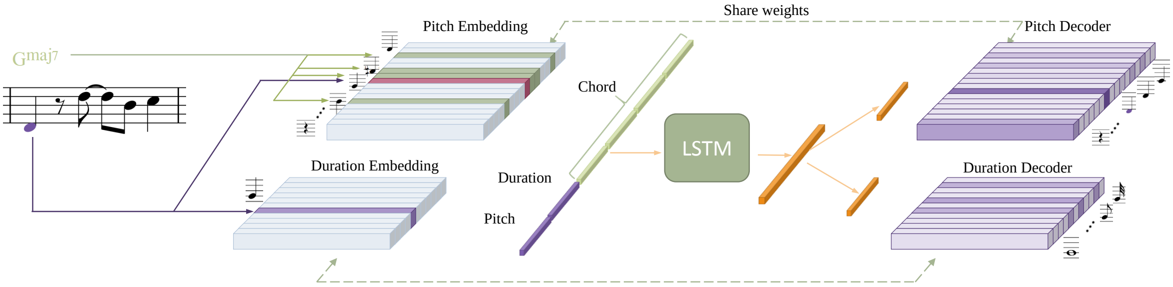


Figure 3. The BebopNet architecture for the next note prediction. Each note is represented by concatenating the embeddings of the pitch (red bar), the duration (purple bar) and the four pitches comprising the current chord (green bars). The output of the LSTM is passed to two heads (orange bars), one the size of the pitch embedding (top) and the other the size of the duration embedding (bottom).

Following [42, 43], we tie the weights of the final fully-connected layers to those of the embedding. Finally, the outputs of the two heads pass through a softmax layer and are trained to minimize the negative log-likelihood of the corpus. To enrich our dataset while encouraging harmonic context dependence, we augment our dataset by transposing to all 12 keys.

4.2 User Preference Elicitation

Using BebopNet, we created a dataset to be labeled by users, consisting of 124 improvisations. These solos were divided into three groups of roughly the same size: solos from the original corpus, solos generated by BebopNet over jazz standards present in the training set, and generated solos over jazz standards not present in the training set. The length of each solo is two choruses, or twice the length of the melody. For each standard, we generated a backing track in MP3 format that includes a rhythm section and a harmonic instrument to play along the improvisation using Band-in-a-Box [44]. This dataset amounts to approximately five hours of played music.

We created a system inspired by CRDI that is entirely digital, replacing the analog dial with strokes of a keyboard moving a digital dial. A figure of our dial is presented in the supplementary material. While the original CRDI had a range of 255 values, our initial experiments found that quantizing the values to five levels was easier for users. We recorded the location of the dial at every time step and aligned it to the note being played at the same moment.

4.3 User Preference Metric Learning

In the user preference metric learning stage we again use supervised learning to train a metric function g_ϕ. This function should predict user preference scores for any solo, given its harmonic context. During training, for each sequence X_τ we estimate y_τ, corresponding to the label the user provided for the last note in the sequence. We choose the last label of the sequence, rather than the mode or mean, because of delayed feedback. During the user elicitation step, we noticed that when a user decides to change the position of the dial, it is because he has just heard a sequence of notes that he considers to be more (or less) pleasing than those he heard previously. Thus, the label indicates the preference of the past sequence. The labels are linearly scaled down to the range $[-1, 1]$. Since the data in $\widetilde{\mathcal{D}}$ is small and unbalanced, we use stratified sampling over solos to divide the dataset into training and validation sets. We then use bagging to create an ensemble of five models for the final estimate.

4.3.1 *Network Architecture*

We estimate the function g_ϕ using transfer learning from BebopNet. The user preference model consists of the same layers as BebopNet without the final fully-connected layers. Next, we apply scaled dot-product attention [45] over τ time steps followed by fully-connected and tanh layers. The transferred layers are initialized using the weights θ of BebopNet. Furthermore, the weights of the embedding layers are frozen during training.

4.3.2 *Selective Prediction*

To elevate the accuracy of g_ϕ, we utilize selective prediction whereby we ignore predictions whose confidence is too low. We use the prediction magnitude as a proxy for confidence. Given confidence threshold parameters, $\beta_1 < 0, \beta_2 > 0$, we define $g'_{\phi,\beta_1,\beta_2}(X_t^i)$ in Eq. 3.

$$g'_{\phi,\beta_1,\beta_2}(X_t^i) = \begin{cases} 0 & \text{if } \beta_1 < g_\phi(X_t^i) < \beta_2 \\ g_\phi(X_t^i) & \text{else} \end{cases} \quad (3)$$

The parameters β_1 and β_2 change our coverage rate and are determined by minimizing error (risk) on the risk-coverage plot along a predefined coverage contour. More details are given in Section 5.2.

4.4 Optimized Music Generation

To optimize generations from f_θ, we apply a variant of beam-search, ψ, whose objective scores are obtained from non-rejected predictions of g_ϕ. Pseudocode of the ψ procedure is presented in the supplementary material. We denote by $V_b = [X_t^1, X_t^2, ..., X_t^b]$ a running batch (beam) of size (beam-width) b containing the most promising candidate sequences found so far by the algorithm. The sequences are all initialized with the starting input sequence. In our

Name	Adderley	Gordon	Getz	Parker	Rollins	Stitt	Woods	Ammons	**BebopNet (Heard)**	**BebopNet (Unheard)**
Chord	0.50	0.54	0.53	0.52	0.52	0.53	0.50	0.54	**0.53**	**0.52**
Scale	0.78	0.83	0.81	0.80	0.81	0.83	0.78	0.83	**0.82**	**0.81**

Table 1. Harmonic coherence: The average chord and scale matches computed for artists in the dataset and for BebopNet. A higher number indicates a high coherency level. BebopNet is measured separately for harmonic progressions heard and not heard in the training dataset.

case, this is the melody of the jazz standard. At every time step t, we produce a probability distribution of the next note of every sequence in V_b by passing the b sequences through the network $f_\theta(X_t^i, c_{t+1}^i)$. As opposed to typical applications of beam-search, rather than choosing the most probable notes from $Pr(s_{t+1}|X_t^i, c_{t+1}^i)$, we independently and randomly sample them. We then calculate the score of the extended candidates using the preference metric, g_ϕ.

Every δ steps, we perform a beam update process. We choose the highest scoring k sequences calculated by g_ϕ. Then we duplicate these sequences b/k times to maintain a full beam of b sequences. Choosing different values of δ allows us to control a horizon parameter, which facilitates longer term predictions when extending candidate sequences in the beam. The use of larger horizons may lead to sub-optimal optimization but increases variability.

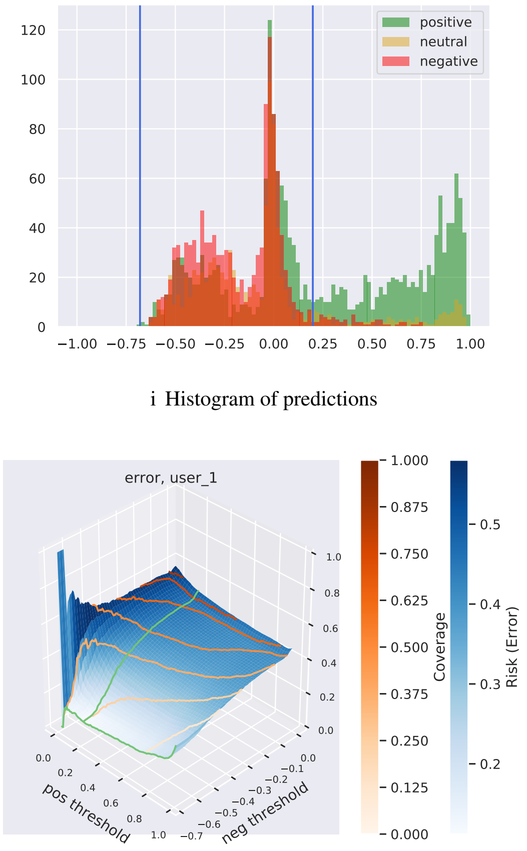
5. EXPERIMENTS

We start the experimental process by training BebopNet as described in Section 4. After training, we use BebopNet to generate multiple solos over different jazz standards [6] . To verify that BebopNet can generalize to harmonic progressions of different musical genres, we also generate improvisations over pop songs (see supplementary material).

This section has two sub-sections. First, we evaluate BebopNet in terms of harmonic coherence (5.1). Next, we present an analysis of our personalization process (5.2). All experiments were performed on desktop computers with a single Titan X GPU. Hyperparameters are provided in the supplementary material.

5.1 Harmonic Coherence

We begin by evaluating the extent to which BebopNet was able to capture the context of chords, which we term *harmonic coherence*. We define two harmonic coherence metrics using either scale match or chord match. These metrics are defined as the percent of time within a measure where notes match pitches of the scale or the chord being played, respectively. We rely on a standard definition of matching scales to chords using the chord-scale system [46]. While most notes in a solo should be harmonically coherent, some non-coherent notes are often incorporated. Common examples of their uses are chromatic lines, approach notes and enclosures [47]. Therefore, as we do not expect a perfect harmonic match according to pure music rules, we take as a baseline the average matching statistics of these quantities for each jazz artist in our dataset. The harmonic coherence statistics of BebopNet are computed over the dataset used for the preference metric learning (generated by BebopNet), which also includes chord progressions not heard during the jazz modeling stage. The baselines and results are reported in Table 1. It is evident that our model exhibits harmonic coherence in the 'ballpark' of the jazz artists even on chord progressions not previously heard.



i Histogram of predictions

ii Risk-coverage plot

Figure 4. 4i Predictions of the preference model on sequences from a validation set. Green: sequences labeled with a positive score ($y_\tau > 0$); yellow: neutral ($y_\tau = 0$); red: negative ($y_\tau < 0$). The blue vertical lines indicate thresholds β_1, β_2 used for selective prediction. 4ii Risk-coverage plot for the predictions of the preference model. β_1, β_2 (green lines) are defined to be the thresholds that yield a minimum error on the contour of 25% coverage.

[6] To appreciate the diversity of BebopNet, listen to seven solos generated for user-4 for the tune Recorda-Me in the supplementary material.

5.2 Analyzing Personalized Models

We applied the proposed pipeline to generate personalized models for each of the four users, all amateur jazz musicians. All users listened to the same training dataset of solos to create their personal metric (see Section 4). Each user provided continuous feedback for each solo using our CRDI variant. In this section, we describe our evaluation process for user-1. The evaluation results for the rest of the users are presented in the supplementary material.

We analyze the quality of our preference metric function g_ϕ by plotting a histogram of the network's predictions applied on a validation set. Consider Figure 4i. We can crudely divide the histogram into three areas: the right-hand side region corresponds to mostly positive sequences predicted with high accuracy; the center region corresponds to high confusion between positive and negative; and the left one, to mostly negative sequences predicted with some confusion. While the overall error of the preference model is high (0.4 MSE where the regression domain is [-1,1]), it is still useful since we are interested in its predictions in the positive (green) spectrum for the forthcoming optimization stage. While trading-off coverage, we increase prediction accuracy using selective prediction by allowing our classifier to abstain when it is not sufficiently confident. To this end, we ignore predictions whose magnitude is between two rejection thresholds (see Section 4.3.2). Based on preliminary observations, we fix the rejection thresholds to maintain 25% coverage over the validation set. In Figure 4ii we present a risk-coverage plot for user-1 (see definition in [8]). The risk surface is computed by moving two thresholds β_1 and β_2 across the histogram in Figure 4i, and at each point, for data not between the thresholds, we calculate the risk (error of classification to three categories: positive, neutral and negative) and the coverage (percent of data maintained).

We increase the diversity of generated samples by taking the score's sign rather than the exact score predicted by the preference model g_ϕ. Therefore, different positive samples are given equal score. For user-1, the average score predicted by g_ϕ for generated solos of BebopNet is **0.07**. As we introduce beam-search and increase the beam width, the performance increases up to an optimal point from which it decreases (see supplementary material). User-1's scores peaked at **0.8** with $b = 32, k = 8$. Anecdotally, there was one solo that user-1 felt was exceptionally good. For that solo, the model predicted the perfect score of 1. This indicates that the use of beam-search is indeed beneficial for optimizing the preference metric.

6. PLAGIARISM ANALYSIS

One major concern is the extent to which BebopNet plagiarizes. In our calculations, two sequences that are identical up to transposition are considered the same. To quantify plagiarism in a solo with respect to a set of source solos, we measure the percentage of n-grams in that solo that also appear in any other solo in the source. These statistics are also applied to any artist in our dataset to form a baseline for the typical amount of copying exhibited by humans.

Another plagiarism measurement we define is the largest common sub-sequence. For each solo, we consider the solos of other artists as the source set. Then, we average the results per artist. Also, for every artist, we compare every solo against the rest of his solos to measure self-plagiarism. For BebopNet, we quantify the plagiarism level with respect to the entire corpus. The average plagiarism level of BebopNet is 3.8. Interestingly, this value lies within the human plagiarism range found in the dataset. This indicates that BebopNet can be accused of plagiarism as much as some of the famous jazz giants. We present the extended results in the supplementary material.

7. CONCLUDING REMARKS

We presented a novel pipeline for generating personalized harmony-constrained jazz improvisations by learning and optimizing a user-specific musical preference model. To distill the noisy human preference models, we used a selective prediction approach. We introduced an objective evaluation method for subjective content and numerically analysed our proposed pipeline on four users.

Our work raises many questions and directions for future research. While our generated solos are locally coherent and often interesting/pleasing, they lack the qualities of professional jazz related to general structure such as motif development and variations. Preliminary models we have trained on smaller datasets were substantially weak. Can a much larger dataset generate a significantly better model? To acquire such a large corpus it might be necessary to abandon the symbolic approach and rely on raw audio.

Our work emphasizes the need to develop effective methodologies and techniques to extract and distill noisy human feedback that will be required for developing many personalized applications. Our proposed method raises many questions. To what extent does our metric express the specifics of one's musical taste? Can we extract precise properties from this metric? Additionally, our technique relies on a sufficiently large labeled sample to be provided by each user, a substantial effort on the user's part. We anticipate that the problem of eliciting user feedback will be solved in a completely different manner, for example, by monitoring user satisfaction unobtrusively, e.g., using a camera, EEG, or even direct brain-computer connections.

The challenge of evaluating neural networks that generate art remains a central issue in this research field. An ideal jazz solo should be creative, interesting and meaningful. Nevertheless, when evaluating jazz solos, there are no mathematical definitions for these properties—as yet. Previous works attempted to define and optimize creativity [48], but no one has yet delineated an explicit objective definition. Some of the main properties of creative performance are innovation and the generations of patterns that reside out-of-the-box— namely, the extrapolation of outlier patterns beyond the observed distribution. Present machine learning regimes, however, are mainly capable of handling interpolation tasks and not extrapolation. Is it at all possible to learn the patterns of outliers?

8. ACKNOWLEDGEMENTS

* Both authors contributed equally to this work.

9. REFERENCES

[1] L. A. Hiller Jr. and L. M. Isaacson, "Musical Composition with a High Speed Digital Computer," in *Audio Engineering Society Convention 9*. Audio Engineering Society, 1957.

[2] N. Jaques, S. Gu, R. E. Turner, and D. Eck, "Tuning Recurrent Neural Networks with Reinforcement Learning," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*, 2017. [Online]. Available: https://openreview.net/forum?id=Syyv2e-Kx

[3] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, C. Hawthorne, A. M. Dai, M. D. Hoffman, and D. Eck, "Music Transformer: Generating Music with Long-Term Structure," *arXiv preprint arXiv:1809.04281*, 2018.

[4] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset," *CoRR*, vol. abs/1810.12247, 2018. [Online]. Available: http://arxiv.org/abs/1810.12247

[5] K. Chen, W. Zhang, S. Dubnov, G. Xia, and W. Li, "The Effect of Explicit Structure Encoding of Deep Neural Networks for Symbolic Music Generation," in *2019 International Workshop on Multilayer Music Representation and Processing (MMRP)*. IEEE, 2019, pp. 77–84.

[6] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," *arXiv preprint arXiv:1904.10509*, 2019.

[7] C. R. Robinson, "Differentiated Modes of Choral Performance Evaluation Using Traditional Procedures and a Continuous Response Digital Interface device," Ph.D. dissertation, Florida State University, 1988.

[8] Y. Geifman and R. El-Yaniv, "Selective Classification for Deep Neural Networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 4878–4887.

[9] Y. Geifman and R. El-Yaniv, "SelectiveNet: A Deep Neural Network with an Integrated Reject Option," in *International Conference on Machine Learning (ICML)*, 2019.

[10] P. Norvig, *Paradigms of Artificial Intelligence Programming: Case Studies in Common LISP*. Morgan Kaufmann, 1992.

[11] J. Gillick, K. Tang, and R. M. Keller, "Learning Jazz Grammars," *Proceedings of the SMC*, pp. 125–130, 2009.

[12] M. Löthe, "Knowledge Based Automatic Composition and Variation of Melodies for Minuets in Early Classical Style," in *Annual Conference on Artificial Intelligence*. Springer, 1999, pp. 159–170.

[13] F. Pachet, "The Continuator: Musical Interaction with Style," *Journal of New Music Research*, vol. 32, no. 3, pp. 333–341, 2003.

[14] R. Wooller and A. R. Brown, "Investigating Morphing Algorithms for Generative Music," in *Third Iteration: Third International Conference on Generative Systems in the Electronic Arts, Melbourne, Australia*, 2005.

[15] J. Sakellariou, F. Tria, V. Loreto, and F. Pachet, "Maximum Entropy Models Capture Melodic Styles," *Scientific reports*, vol. 7, no. 1, p. 9172, 2017.

[16] P. Laine and M. Kuuskankare, "Genetic Algorithms in Musical Style Oriented Generation," in *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*. IEEE, 1994, pp. 858–862.

[17] G. Papadopoulos and G. Wiggins, "A Genetic Algorithm for the Generation of Jazz Melodies," *Proceedings of STEP*, vol. 98, 1998.

[18] P. Toiviainen, "Modeling the Target-Note Technique of Bebop-Style Jazz Improvisation: An Artificial Neural Network Approach," *Music Perception: An Interdisciplinary Journal*, vol. 12, no. 4, pp. 399–413, 1995.

[19] M. Nishijima and K. Watanabe, "Interactive Music Composer Based on Neural Networks," in *Proceedings of the 1992 International Computer Music Conference, ICMC 1992, San Jose, California, USA, October 14-18, 1992*, 1992. [Online]. Available: http://hdl.handle.net/2027/spo.bbp2372.1992.015

[20] J. Franklin, "Multi-Phase Learning for Jazz Improvisation and Interaction," in *In Proceedings of the Biennial Symposium on Arts and Technology*, 2001.

[21] J. D. Fernández and F. Vico, "AI Methods in Algorithmic Composition: A Comprehensive Survey," *Journal of Artificial Intelligence Research*, vol. 48, pp. 513–582, 2013.

[22] D. Eck and J. Schmidhuber, "Learning the Long-Term Structure of the Blues," in *International Conference on Artificial Neural Networks*. Springer, 2002, pp. 284–289.

[23] J. A. Franklin, "Jazz Melody Generation Using Recurrent Networks and Reinforcement Learning," *International Journal on Artificial Intelligence Tools*, vol. 15, no. 04, pp. 623–650, 2006.

[24] D. D. Johnson, R. M. Keller, and N. Weintraut, "Learning to Create Jazz Melodies Using a Product of Experts," in *Proceedings of the Eighth International Conference on Computational Creativity (ICCC'17), Atlanta, GA, 19*, 2017, p. 151.

[25] G. E. Hinton, “Training Products of Experts by Minimizing Contrastive Divergence,” *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[26] L. Yang, S. Chou, and Y. Yang, “MidiNet: A Convolutional Generative Adversarial Network for Symbolic-Domain Music Generation,” in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, 2017, pp. 324–331. [Online]. Available: https://ismir2017.smcnus.org/wp-content/uploads/2017/10/226_Paper.pdf

[27] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset,” in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=r1lYRjC9F7

[28] G. Hadjeres, F. Pachet, and F. Nielsen, “DeepBach: a Steerable Model for Bach Chorales Generation,” in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017, pp. 1362–1371. [Online]. Available: http://proceedings.mlr.press/v70/hadjeres17a.html

[29] H. H. Mao, T. Shin, and G. W. Cottrell, “DeepJ: Style-Specific Music Generation,” in *12th IEEE International Conference on Semantic Computing, ICSC 2018, Laguna Hills, CA, USA, January 31 - February 2, 2018*, 2018, pp. 377–382. [Online]. Available: https://doi.org/10.1109/ICSC.2018.00077

[30] G. Hadjeres and F. Nielsen, “Interactive Music Generation with Positional Constraints using Anticipation-RNNs,” *CoRR*, vol. abs/1709.06404, 2017. [Online]. Available: http://arxiv.org/abs/1709.06404

[31] E. Schubert, “Continuous Measurement of Self-Report Emotional Response to Music,” *Music and Emotion: Theory and Research*, pp. 394–414, 2001.

[32] C. K. Madsen and J. M. Geringer, “Comparison of Good Versus Bad Tone Quality/Intonation of Vocal and String Performances: Issues Concerning Measurement and Reliability of the Continuous Response Digital Interface,” *Bulletin of the Council for Research in Music education*, pp. 86–92, 1999.

[33] E. Himonides, “Mapping a Beautiful Voice: The Continuous Response Measurement Apparatus (CReMA),” *Journal of Music, Technology & Education*, vol. 4, no. 1, pp. 5–25, 2011.

[34] R. V. Brittin, “Listeners’ Preference for Music of Other Cultures: Comparing Response Modes,” *Journal of Research in Music Education*, vol. 44, no. 4, pp. 328–340, 1996.

[35] J. C. Coggiola, “The Effect of Conceptual Advancement in Jazz Music Selections and Jazz Experience on Musicians’ Aesthetic Response,” *Journal of Research in Music Education*, vol. 52, no. 1, pp. 29–42, 2004.

[36] “Sax solos,” https://saxsolos.com/, accessed: 2019-05-16.

[37] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional Sequence to Sequence Learning,” in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017, pp. 1243–1252. [Online]. Available: http://proceedings.mlr.press/v70/gehring17a.html

[38] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “WaveNet: A Generative Model for Raw Audio,” in *The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016*, 2016, p. 125. [Online]. Available: http://www.isca-speech.org/archive/SSW_2016/abstracts/ssw9_DS-4_van_den_Oord.html

[39] R. Al-Rfou, D. Choe, N. Constant, M. Guo, and L. Jones, “Character-Level Language Modeling with Deeper Self-Attention,” *CoRR*, vol. abs/1808.04444, 2018. [Online]. Available: http://arxiv.org/abs/1808.04444

[40] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[41] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, “Transformer-xl: Attentive language models beyond a fixed-length context,” *arXiv preprint arXiv:1901.02860*, 2019.

[42] H. Inan, K. Khosravi, and R. Socher, “Tying Word Vectors and Word Classifiers: A Loss Framework for Language Modeling,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. [Online]. Available: https://openreview.net/forum?id=r1aPbsFle

[43] O. Press and L. Wolf, “Using the Output Embedding to Improve Language Models,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, 2017, pp. 157–163. [Online]. Available: https://aclanthology.info/papers/E17-2025/e17-2025

[44] PG Music Inc., “Band-in-a-box.” [Online]. Available: https://www.pgmusic.com/

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is All you Need,” in *Advances in*

Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, 2017, pp. 6000–6010. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need

[46] M. Cooke, D. Horn, and J. Cross, *The Cambridge Companion to Jazz.* Cambridge University Press, 2002.

[47] J. Cocker, "Elements of the Jazz Language for the Developing Improviser," *Miami: CPP Belwin*, 1991.

[48] J. Schmidhuber, "Formal theory of creativity, fun, and intrinsic motivation (1990–2010)," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pp. 230–247, 2010.