ACCOMONTAGE: ACCOMPANIMENT ARRANGEMENT VIA PHRASE SELECTION AND STYLE TRANSFER

Jingwei Zhao
Music X Lab, NYU Shanghai
`jz4807@nyu.edu`

Gus Xia
Music X Lab, NYU Shanghai
`gxia@nyu.edu`

ABSTRACT

Accompaniment arrangement is a difficult music generation task involving intertwined constraints of melody, harmony, texture, and music structure. Existing models are not yet able to capture all these constraints effectively, especially for long-term music generation. To address this problem, we propose *AccoMontage*, an accompaniment arrangement system for *whole pieces* of music through unifying phrase selection and neural style transfer.[1] We focus on generating piano accompaniments for folk/pop songs based on a lead sheet (i.e., melody with chord progression). Specifically, AccoMontage first retrieves phrase montages from a database while recombining them structurally using dynamic programming. Second, chords of the retrieved phrases are manipulated to match the lead sheet via style transfer. Lastly, the system offers controls over the generation process. In contrast to pure learning-based approaches, AccoMontage introduces a novel hybrid pathway, in which rule-based optimization and deep learning are both leveraged to complement each other for high-quality generation. Experiments show that our model generates well-structured accompaniment with delicate texture, significantly outperforming the baselines.

1. INTRODUCTION

Accompaniment arrangement refers to the task of reconceptualizing a piece by composing the accompaniment part given a lead sheet (a lead melody with a chord progression). When designing the texture and voicing of the accompaniment, arrangers are simultaneously dealing with the *constraints* from the original melody, chord progression, and other structural information. This constrained composition process is often modeled as a *conditioned generation problem* in music automation.

Despite recent promising advances in deep music generative models [1–7], existing methods cannot yet generate musical accompaniment while capturing the aforementioned constraints effectively. Specifically, most al-

gorithms fall short in preserving the fine granularity and structure of accompaniment in the long run. Also, it is difficult to explicitly control the generation process. We argue that these limits are mainly due to the current *generation from scratch* approach. In composition practice, however, arrangers often resort to existing pieces as accompaniment references. For example, a piano accompanist can improvise through off-the-shelf textures while transferring them into proper chords, which is essentially re-harmonizing a reference piece to fit a query lead sheet. In this way, the coherence and structure of the accompaniment are preserved from the reference pieces, and musicians also have control over what reference to choose.

To this end, we contribute *AccoMontage*, a *generalized template-based* approach to 1) given a lead sheet as the query, search for proper accompaniment phrases as the reference; 2) re-harmonize the selected reference via style transfer to accompany the query. We model the search stage as an optimization problem on the graph, where nodes represent candidate phrases in the dataset and edges represent inter-phrase transitions. Node scores are defined in a rule-based manner to reveal query-reference fitness, while edge scores are learned by contrastive learning to reveal smoothness of phrase transitions. As for the re-harmonization stage, we adopt the chord-texture disentanglement and transfer method in [8, 9].

The current system focuses on arranging piano accompaniments for a full-length folk or pop song. Experimental results show that the generated accompaniments not only harmonize well with the melody but also contain more intra-phrase coherence and inter-phrase structure compared to the baselines. In brief, our contributions are:

- **A generalized template-based approach**: A novel hybrid approach to generative models, where searching and deep learning are both leveraged to complement each other and enhance the overall generation quality. This strategy is also useful in other domains.

- **The AccoMontage system**: A system capable of generating long-term and structured accompaniments for full-length songs. The arranged accompaniments have state-of-the-art quality and are significantly better than existing pure learning-based and template-based baselines.

- **Controllable music generation**: Users can control the generation process by pre-filtering of two texture features: rhythm density and voice number.

---

[1] Codes and demos at https://github.com/zhaojw1998/AccoMontage.

2. RELATED WORK

We review three topics related to symbolic accompaniment arrangement: conditional music generation, template-based arrangement, and music style transfer.

2.1 Conditional Music Generation

Conditional music generation takes various forms, such as generating chords conditioned on the melody [10,11], generating melody on the underlying chords [6,7], and generating melody from metadata and descriptions [12]. In particular, accompaniment arrangement refers to generating accompaniment conditioned on the lead sheet, and this topic has recently drawn much research attention. We even see tailored datasets for piano arrangement tasks [13].

For accompaniment arrangement, existing models that show satisfied arrangement quality typically apply only to *short* clips. GAN and VAE-based models are used to maintain inter-track music dependency [14–16], but limit music generation within 4 to 8 bars. Another popular approach is to generate longer accompaniment in a seq2seq manner with attention [5,6,8], but can easily converge to repetitive textural patterns in the long run. On the other hand, models that arrange for complete songs typically rely on a library of fixed elementary textures and often fail to generalize [17–19]. This paper aims to unite both high-quality and long-term accompaniment generation in one system, where "long-term" refers to full songs (32 bars and more) with dependencies to intra-phrase melody and chord progression, and inter-phrase structure.

2.2 Template-based Accompaniment Arrangement

The use of existing compositions to generate music is not an entirely new idea. Existing template-based algorithms include learning-based unit selection [20, 21], rule-based matching [17, 18], and genetic algorithms [19]. For accompaniment arrangement, a common problem lies in the difficulty to find a perfectly matched reference especially when the templates contain rich textures with non-chordal tones. Some works choose to only use basic accompaniment patterns to avoid this issue [17–19]. In contrast, our study addresses this problem by applying the style transfer technique on a selected template to improve the fitness between the accompaniment and the lead sheet. We name our approach after *generalized* template matching.

2.3 Music Style transfer

Music style transfer [22] is becoming a popular approach for controllable music generation. Through music-representation disentanglement and manipulation, users can transfer various factors of a reference music piece, including pitch contour, rhythm pattern, chord progression, polyphonic texture, etc [1,8]. Our approach can be seen as an extension of music style transfer in which the "reference search" step is also automated.

3. METHODOLOGY

The AccoMontage system uses a generalized template-based approach for piano accompaniment arrangement. The input to the system is a lead sheet of a complete folk/pop song with phrase labels, which we call a *query*. The search space of the system is a MIDI dataset of piano-arranged pop songs. In general, we can derive the chord progression and phrase labels of each song in the dataset by MIR algorithms. In our case, the chords are extracted by [13] and the phrases are labeled manually [23]. We refer to each phrase of associated accompaniment, melody, and chords as a *reference*. For the rest of this section, we first introduce the feature representation of the AccoMontage system in Section 3.1, and then describe the main pipeline algorithms in Section 3.2 and 3.3. Finally, we show how to further control the arrangement process in Section 3.4.

3.1 Feature Representation

Given a lead sheet as the query, we represent it as a sequence of ternary tuples:

$$q = \{(q_i^{\mathrm{mel}}, q_i^{\mathrm{chord}}, q_i^{\mathrm{label}})\}_{i=1}^n, \qquad (1)$$

where q_i^{mel}, the melody feature of query phrase i, is a sequence of 130-D one-hot vectors with 128 MIDI pitches plus a hold and a rest state [24]; q_i^{chord}, the chord feature aligned with q_i^{mel}, is a sequence of 12-D chromagram vectors [1, 2]; q_i^{label} is a phrase label string denoting within-song repetition and length in bar, such as A8, B8, etc. [23]. n is the number of phrases in lead sheet q.

We represent the accompaniment reference space as a collection of tuples:

$$r = \{(r_i^{\mathrm{mel}}, r_i^{\mathrm{chord}}, r_i^{\mathrm{acc}})\}_{i=1}^N, \qquad (2)$$

where r_i^{mel}, and r_i^{chord} are the melody and the chord feature of the i-th reference phrase, represented in the same format as in the query phrases; r_i^{acc} is the accompaniment feature, which is a 128-D piano-roll representation the same as [8]. N is the volume of the reference space.

3.2 Phrase Selection

Assuming there are n phrases in the query lead sheet, we aim to find a reference sequence:

$$\mathbf{x} = [x_1, x_2, \cdots, x_n], \qquad (3)$$

where we match reference x_i to the i-th phrase q_i in our query; $x_i \in r$ and has the same length as q_i.

Given the query's phrase structure, the reference space forms a graph of layered structures shown as Figure 1. Each layer consists of equal-length reference phrases and consecutive layers are fully connected to each other. Each node in graph describes the fitness between x_i and q_i, and each edge evaluates the transition from x_i to x_{i+1}. A complete selection of reference phrases corresponds to a path that traverses through all layers. To evaluate a path, We design a *fitness model* and a *transition model* as follows.

3.2.1 Phrase Fitness Model

We rely on the phrase fitness model to evaluate if a reference accompaniment phrase matches a query phrase. Formally, we define the fitness model $f(x_i, q_i)$ as follows:

$$
\begin{aligned}
f(x_i, q_i) = {} & \alpha \mathrm{sim}(x_i^{\mathrm{rhy}}, q_i^{\mathrm{rhy}}) \\
& + \beta \mathrm{sim}(\mathrm{T}(x_i^{\mathrm{chord}}), \mathrm{T}(q_i^{\mathrm{chord}})),
\end{aligned} \tag{4}
$$

where $\mathrm{sim}(\cdot, \cdot)$ measures the similarity between two inputs. In our work, we use the cosine similarity. $\mathrm{T}(\cdot)$ is the Tonal Interval Vector (TIV) operator that maps a chromagram to a 12-D tonal interval space whose geometric properties concur with harmonic relationships of the tonal system [25]. x_i^{rhy} and q_i^{rhy} are both rhythm features, which condense the original 130-D melody feature to 3-D that denotes an onset of any pitch, a hold state, and rest [1]. x_i^{chord} and q_i^{chord} are chord features (chromagram) defined in Section 3.1 and we further augment the reference space by transposing phrases to all 12 keys. While computing the similarity, we consider the rhythm feature and TIV as 2-D matrices each with channel number 3 and 12. We calculate the cosine similarity of both features by feeding in their channel-flattened vectors.

Note that in Eq (4), we compare only the rhythm and chord features for query-reference matching. The underlying assumption is that *if lead sheet A is similar to another lead sheet B in rhythm and chord progression, then B's accompaniment will be very likely to fit A as well.*

3.2.2 Transition Model

We exploit the transition model to reveal the inter-phrase transition and structural constraints. Formally, we define the transition score between two reference accompaniment phrases $t(x_i, x_{i+1})$ as follows:

$$
\begin{aligned}
t(x_i, x_{i+1}) = {} & \mathrm{sim}(W_1 x_i^{\mathrm{txt}}, W_2 x_{i+1}^{\mathrm{txt}}) \\
& + \mathrm{form}(x_i, x_{i+1}).
\end{aligned} \tag{5}
$$

The first term in Eq (5) aims to reveal the transition naturalness of the polyphonic *texture* between two adjacent phrases. Instead of using rule-based heuristics to process texture information, we resort to neural representation learning and contrastive learning. Formally, let x_i^{txt} denote the feature vector that represents the accompaniment texture of x_i^{acc}. It is computed by:

$$
x_i^{\mathrm{txt}} = \mathrm{Enc}_\theta^{\mathrm{txt}}(x_i^{\mathrm{acc}}), \tag{6}
$$

where the design of $\mathrm{Enc}_\theta^{\mathrm{txt}}(\cdot)$ is adopted from the chord-texture representation disentanglement model in [8]. This texture encoder regards piano-roll inputs as images and uses a CNN to compute a rough "sketch" of polyphonic texture that is not sensitive to mild chord variations.

To reveal whether two adjacent textures $(x_i^{\mathrm{txt}}, x_{i+1}^{\mathrm{txt}})$ follow a natural transition, we use a contrastive loss $\mathcal{L}$ to simultaneously train the weight matrix W in Eq (5) and fine-tune $\mathrm{Enc}_\theta^{\mathrm{txt}}(\cdot)$ (with parameter θ) in Eq (6):

$$
\mathcal{L}(W, \theta) = 1 - \frac{\exp(\mathrm{sim}(W_1 x_i^{\mathrm{txt}}, W_2 x_{i+1}^{\mathrm{txt}}))}{\sum_{x \in S} \exp(\mathrm{sim}(W_1 x_i^{\mathrm{txt}}, W_2 x_k^{\mathrm{txt}}))}, \tag{7}
$$

Figure 1. Phrase selection on the graph. Based on the lead sheet with an AABB phrase structure, the search space forms a graph with four consecutive layers. Graph nodes are assigned with similarity scores, and edges with transition scores. The form term is part of the transition score.

where x_i and x_{i+1} are supposed to be consecutive pairs. S is a collection of k samples which contains x_{i+1} and other $k-1$ randomly selected phrases from reference space r. Following [20], we choose $k = 5$.

For the form term $\mathrm{form}(x_i, x_{i+1})$, we introduce this term to bias a more well-structured transition. Concretely, if query phrases q_i and q_{i+1} share the same phrase label, we would prefer to also retrieve equal-labeled references, i.e., accompaniments with recapitulated melody themes. To maximize such likelihood, we define the form term:

$$
\mathrm{form}(x_i, x_{i+1}) = \mathbb{1}_{\{q_i^{\mathrm{label}} = q_{i+1}^{\mathrm{label}}\}} \cdot \mathbb{1}_{\{x_i^{\mathrm{mel}} \approx x_{i+1}^{\mathrm{mel}}\}}, \tag{8}
$$

where we define $x_i^{\mathrm{mel}} \approx x_{i+1}^{\mathrm{mel}}$ if and only if their step-wise cosine similarity is greater than 0.99.

3.2.3 Model Inference

The reference space forms a layered graph with consecutive layers fully connected to each other. In Figure 1, we leverage the transition model to assign weights of edges and the fitness model to assign weights of nodes. Thus, the phrase selection is formulated as:

$$
\mathbf{x}^* = \operatorname*{argmax}_{x_1, x_2, \cdots, x_n} \delta \sum_{i=1}^n f(x_i, q_i) + \gamma \sum_{i=1}^{n-1} t(x_i, x_{i+1}), \tag{9}
$$

where $f(\cdot)$ and $t(\cdot)$ are as defined in Eq (4) and Eq (5), and δ and γ are hyper-parameters.

We optimize Eq (9) by dynamic programming to retrieve the Viterbi path $\mathbf{x}^*$ as the optimal solution [26]. The time complexity is $\mathcal{O}(nN^2)$, where n is the number of query phrases and N is the volume of the reference space.

In summary, the phrase selection algorithm enforces strong structural constraints (song-level form and phrase-level fitness) as well as weak harmonic constraints (chord term in Eq (4)) to the selection of accompaniment reference. We argue that this is a good compromise because strong harmonic constraints can potentially "bury" well-structured references due to unmatched chord when our

dataset is limited. To maintain a better harmonic fitness, we resort to music style transfer.

3.3 Style Transfer

The essence of style transfer is to transfer the chord sequence of a selected reference phrase while keeping its texture. To this end, we adopt the chord-texture disentanglement VAE framework by [8]. The VAE consists of a chord encoder Enc^{chd} and a texture encoder Enc^{txt}. Enc^{chd} takes in a two-bar chord progression under one-beat resolution and exploits a bi-directional GRU to approximate a latent chord representation z_{chd}. Enc^{txt} is introduced in Section 3.2.2 and it extracts a latent texture representation z_{txt}. The decoder Dec takes the concatenation of z_{chd} and z_{txt} and decodes the music segment using the same architecture invented in [9]. Sustaining texture input and varying chords, the whole model works like a conditional VAE which re-harmonizes texture based on the chord condition.

In our case, to re-harmonize the selecetd accompaniment x_i^{acc} to query lead sheet q_i, the style transfer works in a pipeline as follows:

$$
\begin{aligned}
z_{\text{chd}} &= \text{Enc}^{\text{chd}}(q_i^{\text{chord}}), \\
z_{\text{txt}} &= \text{Enc}^{\text{txt}}(x_i^{\text{acc}}), \\
x_i' &= \text{Dec}(z_{\text{chd}}, z_{\text{txt}}),
\end{aligned}
\tag{10}
$$

where x_i' is the re-harmonized result. The final accompaniment arrangement result is $\mathbf{x}^{*\prime} = [x_1^{*\prime}, x_2^{*\prime}, \cdots, x_n^{*\prime}]$.

3.4 Controllability

In the phrase selection stage, we essentially traverse a route on the graph. Intuitively, we can control generation of the whole route by assigning the first node. In our case, we filter reference candidates for the first phrase based on textural properties. The current design has two filter criteria: *rhythm density* and *voice number*. we define three intervals *low*, *medium*, and *high* for both properties and mask the references that do not fall in the expected interval.

- Rhythm Density (RD): the ratio of time steps with note onsets to all time steps;

- Voice Number (VN): the average number of notes that are simultaneously played.

4. EXPERIMENT

4.1 Dataset

We collect our reference space from POP909 dataset [13] with the phrase segmentation created by [23]. POP909 contains piano arrangements of 909 popular songs created by professional musicians, which is a great source of delicate piano textures. Each song has a separated melody, chord, and accompaniment MIDI track. We only keep the pieces with $\frac{2}{4}$ and $\frac{4}{4}$ meters and quantize them at 16th notes (chords at 4th). This derives 857 songs segmented into 11032 phrases. As shown in Table 1, we have four-bar and

Table 1. Length Distribution of POP909 Phrase

bars	<4	4	5~7	8	>8
Phrases	1338	3591	855	3796	1402

eight-bar phrases in majority, which makes sense for popular songs. We also use POP909 to fine-tune our transition model, during which we randomly split the dataset (at song level) into training (95%) and validation (5%) sets.

At inference time, the query lead sheets come from the Nottingham Dataset [27], a collection of ~1000 British and American folk tunes. We also adopt $\frac{2}{4}$ and $\frac{4}{4}$ pieces quantized at 16th (chords at 4th). We label their phrase segmentation by hand, where four-bar and eight-bar phrases are also the most common ones.

4.2 Architecture Design

We develop our model based on the chord-texture disentanglement model proposed by [8], which comprises a texture encoder, a chord encoder, and a decoder. The texture encoder consists of a convolutional layer with kernel size 12×4 and stride 1×4 and a bi-directional GRU encoder [24]. The convolutional layer is followed by a ReLU activation [28] and max-pooling with kernel size 4×1 and stride 4×1. The chord encoder is a bi-directional GRU encoder. The decoder is consistent with PianoTree VAE [9], a hierarchical architecture for polyphonic representation learning. The architecture of $\text{Enc}^{\text{txt}}(\cdot)$ in the proposed transition model is the same as the texture encoder illustrated above. We directly take the chord-texture disentanglement model with pre-trained weights as our style transfer model. We fine-tune the transition model with W_1 and W_2 in Eq (7) as trainable parameters.

4.3 Training

Our model is trained with a mini-batch of 128 piano-roll pairs for 50 epochs using Adam optimizer [29] with a learning rate from 1e-4 exponentially decayed to 5e-6. Note that each piano-roll pair contains 2 consecutive piano-rolls and 4 randomly sampled ones. We first pre-train a chord-texture disentanglement model and initialize $\text{Enc}^{\text{txt}}(\cdot)$ using weights of the texture encoder in the pre-trained model. Then we update all the parameters of the proposed transition model using contrastive loss $\mathcal{L}$ in Eq (7). We set both α and β in Eq (4) to 0.5. During inference, we set δ and γ in Eq (9) to 0.3 and 0.7.

4.4 Generated Examples

To this end, we show two long-term accompaniment arrangement examples by the Accomontage system. The first one is illustrated in Figure 2, in which we show a whole piece (32-bar music) piano arrangement (the bottom two staves) base on the lead sheet (the top stave). We see that the generated accompaniment matches with the melody and has a natural flow on its texture. Moreover, it follows the `A8A8B8B8` structure of the melody.

Figure 2. Accompaniment arrangement for *Castles in the Air* from Nottingham Dataset by AccoMontage. The 32-bar song has an `A8A8B8B8` phrase structure which is captured during accompaniment arrangement. Second melodies and texture variations are also introduced to manifest music flow. Here we highlight some texture re-harmonization of 7th chords.
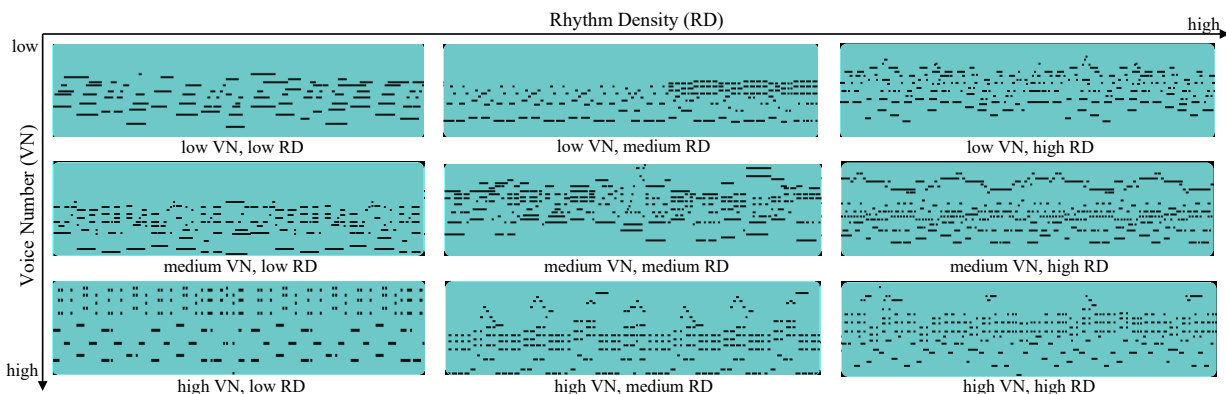

Figure 3. Pre-Filtering Control on Rhythm Density and Voice Number

The second example shows that our controls on rhythm density and voice number are quite successful. To better illustrate, we switch to a piano-roll representation in Figure 3, where 9 arranged accompaniments for the same lead sheet is shown in a 3 by 3 grid. The rhythm density control increases from left to right, while the voice number control increases from top to bottom. We can see that both controls have a significant influence on the generated results.

4.5 Evaluation

4.5.1 Baseline Models

The AccoMontage system is a generalized template-based model that leverages both rule-based optimization and deep learning to complement each other. To evaluate, we introduce a hard template-based and a pure learning-based baseline to compare with our model. Specifically, the base-

line model architectures are as follows:

Hard Template-Based (HTB): The hard template-based model also retrieves references from existing accompaniment, but directly applies them without any style transfer. It uses the same phrase selection architecture as our model while skipping the style transfer stage.

Pure Learning-Based (PLB): We adopt the accompaniment arrangement model in [8], a seq2seq framework combining Transformer [30] and chord-texture disentanglement. We consider [8] the current state-of-the-art algorithm for controllable accompaniment generation due to its tailored design of harmony and texture representations, sophisticated neural structure, and convincing demos. The input to the model is a lead sheet and its first four-bar accompaniment. The model composes the rest by predicting every four bars based on the current lead sheet and previous four-bar accompaniment.
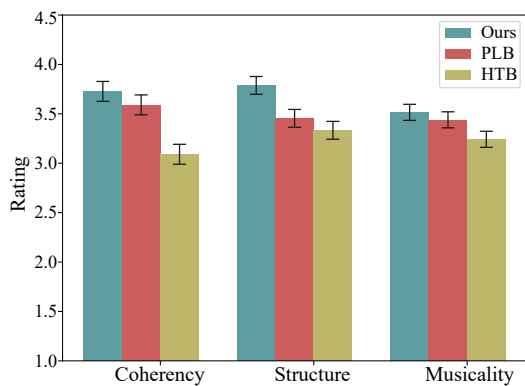
Figure 4. Subjective Evaluation Results.

Figure 5. Evaluation of Transition Model. The contrastive loss (CL) and differences of RD and VN are calculated for three types of phrase pairs. A consistent decreasing trend illustrates reliable discernment of smooth transition.

4.5.2 Subjective Evaluation

We conduct a survey to evaluate the musical quality of the arrangement performance of all models. In our survey, each subject listens to 1 to 3 songs randomly selected from a pool of 14. All 14 songs are randomly selected from the Nottingham Dataset, 12 of which have 32 bars and the other two 24 and 16 bars. Each song has three accompaniment versions generated by our and the baseline models. The subjects are required to rate all three accompaniment versions of one song based on three metrics: coherence, structure, and musicality. The rating is base on a 5-point scale from 1 (very poor) to 5 (excellent).

- **Coherence**: If the accompaniment matches the lead melody in harmony and texture;

- **Structure**:If the accompaniment flows dynamically with the structure of the melody;

- **Musicality**: Overall musicality of accompaniment.

A total of 72 subjects (37 females and 35 males) participated in our survey and we obtain 67 effective ratings for each metric. As in Figure 4, the heights of bars represent the means of the ratings and the error bars represent the MSEs computed via within-subject ANOVA [31]. We report a significantly better performance of our model than both baselines in coherence and structure ($p < 0.05$), and a marginally better performance in musicality ($p = 0.053$).

4.5.3 Objective Evaluation

In the phrase selection stage, we leverage a self-supervised contrastive loss (Eq (7)) to enforce a smooth textural transition among reference phrases. We expect a lower loss for true adjacent phrase pairs than in other situations. Meanwhile, true consecutive pairs should hold a similar texture pattern with smaller differences in general properties.

We investigate the contrastive loss (CL) and the difference of rhythm density (RD) and voice number (VN) among three types of phrase pairs from the validation set. Namely, *Random*, *Same Song*, and *Adjacent*. Between totally randomly pairing and strict adjacency, *Same Song* refers to randomly selecting two phrases (not necessarily adjacent) from one song. Results are shown in Figure 5.
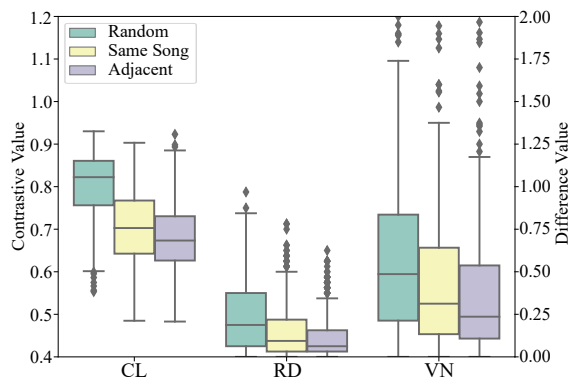
Table 2. Ranking Accuracy and Mean Rank

Metric	Phrase Acc.	Song Acc.	Rank@50
Value	0.2425	0.3769	5.8003

For contrastive loss and each property, we see a consistent decreasing trend from *Random* to *Same Song* and to *Adjacent*. Specifically, we see the upper quartile of *Adjacent* is remarkably lower than the lower quartile of *Random* for CL, which indicates a reliable textural discernment that ensures smooth phrase transitions. This is also proved by the metric of ranking accuracy and mean rank [20], where the selection rank of the true adjacent phrase out of $k - 1$ randomly selected phrases (Rank@k) is calculated. We follow [20] and adopt Rank@50, and the results are shown in Table 2. Phrase Acc. and Song Acc. each refers to the accuracy that the top-ranked phrase is *Adjacent* or belongs to the *Same Song*. The high rank of adjacent pairs illustrates our model's reliability to explore smooth transitions.

5. CONCLUSION

In conclusion, we contribute a generalized template-based algorithm for the accompaniment arrangement problem. The main novelty lies in the methodology that seamlessly combines deep generation and search-based generation. In specific, searching is used to optimize the high-level structure, while neural style transfer is in charge of local coherency and melody-accompaniment fitness. Such a top-down hybrid strategy is inspired by how human musicians arrange accompaniments in practice. We aim to bring a new perspective not only to music generation, but to long-term sequence generation in general. Experiments show that our AccoMontage system significantly outperforms pure learning-based and template-based methods, being capable of rendering well-structured and fine-grained accompaniment for full-length songs.

6. ACKNOWLEDGEMENT

7. REFERENCES

[1] R. Yang, D. Wang, Z. Wang, T. Chen, J. Jiang, and G. Xia, "Deep music analogy via latent representation disentanglement," *arXiv preprint arXiv:1906.03626*, 2019.

[2] K. Chen, G. Xia, and S. Dubnov, "Continuous melody generation via disentangled short-term representations and structural conditions," in *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, 2020, pp. 128–135.

[3] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," *arXiv preprint arXiv:1809.04281*, 2018.

[4] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[5] Y. Ren, J. He, X. Tan, T. Qin, Z. Zhao, and T.-Y. Liu, "Popmag: Pop music accompaniment generation," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1198–1206.

[6] H. Zhu, Q. Liu, N. J. Yuan, C. Qin, J. Li, K. Zhang, G. Zhou, F. Wei, Y. Xu, and E. Chen, "Xiaoice band: A melody and arrangement generation framework for pop music," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2837–2846.

[7] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "Midinet: A convolutional generative adversarial network for symbolic-domain music generation," *arXiv preprint arXiv:1703.10847*, 2017.

[8] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," *arXiv preprint arXiv:2008.07122*, 2020.

[9] Z. Wang, Y. Zhang, Y. Zhang, J. Jiang, R. Yang, J. Zhao, and G. Xia, "Pianotree vae: Structured representation learning for polyphonic music," *arXiv preprint arXiv:2008.07118*, 2020.

[10] I. Simon, D. Morris, and S. Basu, "Mysong: automatic accompaniment generation for vocal melodies," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2008, pp. 725–734.

[11] H. Lim, S. Rhyu, and K. Lee, "Chord generation from symbolic melody using blstm networks," *arXiv preprint arXiv:1712.01011*, 2017.

[12] Y. Zhang, Z. Wang, D. Wang, and G. Xia, "Butter: A representation learning framework for bi-directional music-sentence retrieval and generation," in *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*, 2020, pp. 54–58.

[13] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, X. Gu, and G. Xia, "Pop909: A pop-song dataset for music arrangement generation," *arXiv preprint arXiv:2008.07142*, 2020.

[14] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[15] H.-M. Liu and Y.-H. Yang, "Lead sheet generation and arrangement by conditional generative adversarial network," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 722–727.

[16] B. Jia, J. Lv, Y. Pu, and X. Yang, "Impromptu accompaniment of pop music using coupled latent variable model with binary regularizer," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–6.

[17] P.-C. Chen, K.-S. Lin, and H. H. Chen, "Automatic accompaniment generation to evoke specific emotion," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2013, pp. 1–6.

[18] Y.-C. Wu and H. H. Chen, "Emotion-flow guided music accompaniment generation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 574–578.

[19] C.-H. Liu and C.-K. Ting, "Polyphonic accompaniment using genetic algorithm with music theory," in *2012 IEEE Congress on Evolutionary Computation*. IEEE, 2012, pp. 1–7.

[20] M. Bretan, G. Weinberg, and L. Heck, "A unit selection methodology for music generation using deep neural networks," *arXiv preprint arXiv:1612.03789*, 2016.

[21] G. Xia, "Expressive collaborative music performance via machine learning," Jul 2018. [Online]. Available: https://kilthub.cmu.edu/articles/thesis/Expressive_Collaborative_Music_Performance_via_Machine_Learning/6716609/1

[22] S. Dai, Z. Zhang, and G. G. Xia, "Music style transfer: A position paper," *arXiv preprint arXiv:1803.06841*, 2018.

[23] S. Dai, H. Zhang, and R. B. Dannenberg, "Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music," *arXiv preprint arXiv:2010.07518*, 2020.

[24] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4364–4373.

[25] G. Bernardes, D. Cocharro, M. Caetano, C. Guedes, and M. E. Davies, "A multi-level tonal interval space for modelling pitch relatedness and musical consonance," *Journal of New Music Research*, vol. 45, no. 4, pp. 281–294, 2016.

[26] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[27] E. Foxley, "Nottingham database," [EB/OL], https://ifdo.ca/~seymour/nottingham/nottingham.html Accessed May 25, 2021.

[28] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[31] H. Scheffe, *The analysis of variance*. John Wiley & Sons, 1999, vol. 72.