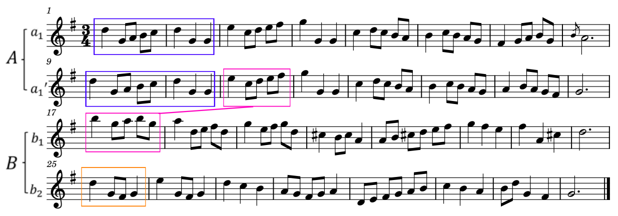
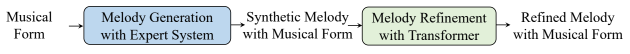
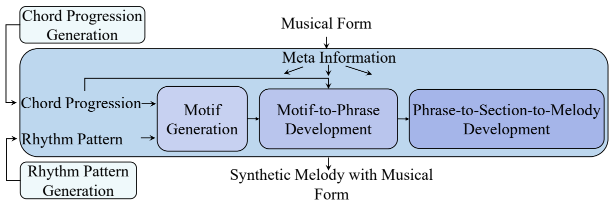


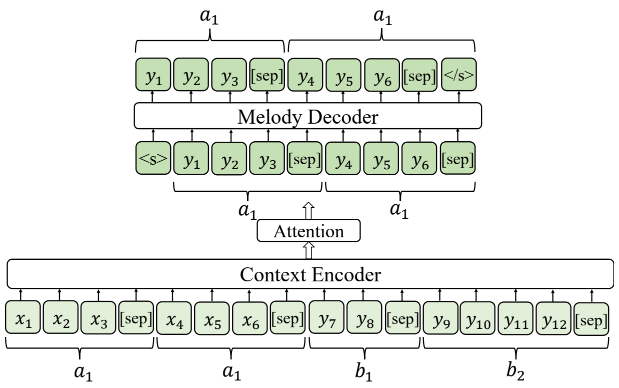
Figure 1. The score of the melody part from "*Minuet in G Major*" by the composer Christian Petzold. This is an example of a melody with musical form as $A(a_1, a_1)B(b_1, b_2)$. We use capital letters (e.g., A, B) to label sections, while using lowercase (e.g., a_1, b_1, b_2) for representing phrases. Different phrases in the same section are labeled with different numbers.


Musical Form
Melody Generation with Expert System
Synthetic Melody with Musical Form
Melody Refinement with Transformer
Refined Melody with Musical Form

(a) Pipeline of MeloForm.


Chord Progression Generation
Musical Form
Meta Information
Chord Progression
Rhythm Pattern
Motif Generation
Motif-to-Phrase Development
Phrase-to-Section-to-Melody Development
Rhythm Pattern Generation
Synthetic Melody with Musical Form

(b) Detailed process of the expert system.


a_1
a_1
y_1 y_2 y_3 [sep] y_4 y_5 y_6 [sep] </s>
Melody Decoder
<s> y_1 y_2 y_3 [sep] y_4 y_5 y_6 [sep]
a_1
a_1
Attention
Context Encoder
x_1 x_2 x_3 [sep] x_4 x_5 x_6 [sep] y_7 y_8 [sep] y_9 y_{10} y_{11} y_{12} [sep]
a_1
a_1
b_1
b_2

(c) Architecture of melody refinement neural networks. The input contains four phrases from a melody with musical form as $A(a_1, a_1)B(b_1, b_2)$. In each phrase, x represents the conditioning information (i.e., rhythm, chord and cadence) for each note, while y represents the melody information (i.e., rhythm and pitch) for each note. [sep] indicates the boundaries between phrases, while <s> and </s> indicates the beginning and end of phrases.

Figure 2. Architecture of MeloForm.

3.1 Melody Generation with Expert System

The designed expert system with purely handcraft rules is inspired by music theory [1], and is shown in Figure 2(b). Given the musical form with hierarchical structure of sections and phrases, the expert system firstly generates the motifs based on chord progression and rhythm patterns, then develops the generated motifs to phrases. In the phrase-to-section-to-melody development module, we arrange the phrases in sections and sections in melody with repetitions and variations according to the given musical form to generate the synthetic melody with musical form. For example in Figure 1, given the musical form $A(a_1, a_1)B(b_1, b_2)$, a 2-bar motif in the blue box is developed into 8-bar phrase a_1. Section A is formed by placing repeated phrase a_1 sequentially with some variations. The similar process is implemented to form section B, in which different phrases b_1 and b_2 are developed from different motifs. Then section A and section B are placed sequentially for getting the final composition.

3.1.1 Motif Generation

Motif is the smallest structural unit for identifying music theme, so we create an initial melody in a few bars (i.e., usually 1-2 bars) to construct the motif. Before the generation process, we should define the meta information (i.e., scale, pitch range, tempo and meter) for guiding the whole generation procedure. Considering note sequences are composed of pitches and rhythm patterns, we come up with rule-based algorithms for chord progression generation to guide pitch selection and for rhythm pattern generation to create rhythm patterns for the note sequence in motif. Specifically, the initial pitch of notes is selected from tones in corresponding chords. Then we add some embellishing tones [3] to decorate the motif melody. Besides, the interval between adjacent note pitches is constrained to be not greater than seven semitones to ensure pitch consistency. The detailed algorithms of chord progression generation and rhythm pattern generation can be referred in Supplementary Materials Section 1 and 2.

3.1.2 Motif-to-Phrase Development

A motif is a unit for identifying the music theme, but it is not a complete music expression. Thus, we need to develop it into a phrase, which is the smallest structural unit for expressing complete music ideas. In order to develop motif into phrase, we introduce three basic categories of development strategies: sequence [4], transformation and ending. These basic development strategies can be combined with each other to create compound development strategies. For example, in Figure 1, the 2-bar motif in the blue box is developed by sequence (i.e., 3-4 bars), transformation (i.e., 5-6 bars) and ending (i.e., 7-8 bars) to form 8-bar phrase a_1. The detailed description of each development strategy can be referred in Supplementary Materials Sections 3.

There may exist multiple different phrases in each section, each of which has its own motif. Too much different motifs in each section may distract listeners from the main ideas. Thus, we present another way to generate a new motif similar to already generated one for a new phrase. Specifically, we can either directly copy the already generated motif or borrow its rhythm patterns to create a new motif, which can help build some relationships between phrases in the same section. For example, in Figure 1, the rhythm patterns from the motif in the orange box of phrase

[3] `http://openmusictheory.com/embellishingTones.html`

[4] `https://en.wikipedia.org/wiki/Sequence_(music)`

b_2 is borrowed from the motif in the pink box of phrase b_1 with a few variations.

3.1.3 Phrase-to-Section-to-Melody Development

After generating all phrases, we need to arrange them with repetitions and variations to form sections, which are higher-level structural units. And sections are then ordered sequentially to finish the composition. As shown in Figure 1, given musical form as $A(a_1, a_1)B(b_1, b_2)$, repeated phrases a_1 with variations are placed in order to form section A. Phrase b_1 and b_2 are placed in order to form section B. At last, section A and section B are arranged sequentially to get the final composition.

Directly generating new motifs when building different sections is not always musically expressive, so we introduce another method for establishing not only similar but also contrastive motifs. To establish the similarity, we can borrow one of the random fragment from phrases in other sections. To form a contrast, we can adjust the rhythm patterns and pitch selection in desired section to change the tension. For example, in Figure 1, the rhythm patterns of the motif in the pink box from phrase b_1 is borrowed from that of the third bar from phrase a_1, which is not the exact motif for phrase a_1. And to form a contrast, the pitch is selected from higher range for more tension.

3.2 Melody Refinement with Transformer

Melodies generated by the expert system are guaranteed to have precise musical form, but are lack of musical richness since they are generated by handcraft rules. Thus, in this section, we introduce an encoder-attention-decoder Transformer that takes the synthetic melodies as input and outputs refined melodies with better musical richness. However, the refinement is challenging since it needs to keep the musical form unchanged while improving the musical richness. Due to the lack of controllability of neural networks, directly refining the whole melody without any constraints is easy to lose musical form information. To address this challenge, we describe several design principles in our Transformer model for refinement:

- *Refining melody phrases by phrases.* Since musical phrase is the smallest structural unit for a complete musical expression, it is better to refine the melody phrase by phrase in an iterative process, instead of refining the whole melody in a single time that fails to maintain the musical form from the expert system. However, only refining one phrase at a time is hard to model the repetitive patterns from similar phrases. Therefore, we should refine similar phrases at each iteration step.

- *Refining phrases by conditioning on rhythm and harmony.* Directly generating the refined musical phrases from scratch may lose bottom-level music structure determined by rhythm patterns and harmony. Thus, we should provide explicit condition and control with rhythm and harmony to guide the refine process.

- *Refining phrases by considering their differentiations among different sections.* When composers need to distinguish different sections (e.g., verse and chorus) in melodies, they usually change pitch distribution or rhythms patterns for building up contrasted tension. Accordingly, to maintain such contrast between sections, we need to consider these differentiations in the refinement process.

Based on these design principles, our model architecture is shown in Figure 2(c). We introduce the implementations corresponding to each design principle as follows.

3.2.1 Iterative Phrase Refinement

According to the first design principle, we refine the similar phrases in each iterative step. To train the Transformer model with iterative refinement capability, we first mask similar phrases of the melody in the training data and take the masked melody as the input of the encoder, and generate original phrases corresponding to the masked positions with the decoder, like the masked sequence to sequence (MASS) task in [39]. After model training, the Transformer model is used to refine the similar phrases in synthetic melodies from expert systems in an iterative way. Specifically, if we want to refine the melody with musical form as $A(a_1, a_1)B(b_1, b_2)$, we can refine phrases a_1 for the first iteration and replace the two original phrases with the refined versions, and then refine the phrase b_2 based on the refined melody. This iteration step is finished until all phrases are refined.

3.2.2 Condition with Rhythm and Harmony

The phrases are developed from motif by sequence strategy that results in similar rhythm patterns within some bars. Besides, harmony plays an important role to control pitch distribution in phrase development. Thus, to maintain these music structures in refinement, we condition the model with rhythm patterns, chord progression and cadence to encourage the model to only refine the pitch of the phrase. Specifically, instead of replacing the masked phrases with masked symbols in MASS [39], we replace them with the corresponding rhythm, chord and cadence symbols, shown as x in Figure 2(c).

3.2.3 Differentiation between Sections

We differentiate the phrases in different sections into aspects: 1) Controlling rhythm patterns in the expert system. Rhythm patterns can be adjusted directly in expert system by increasing/decreasing note density, prolong/shrink note length, etc. It is common that larger note density or faster tempo can bring about more intensity. 2) Controlling the pitch distribution in the Transformer model. Although expert systems can control the pitch distribution by selecting pitch from different ranges, it is hard for neural networks to control in the same way. Thus, we insert an "AVG-PITCH" token at the beginning of the condition from each target phrase to represent the average pitch of this phrase, and another "SPAN" token to indicate the pitch span (i.e., the difference between the maximum pitch with minimum pitch). Higher average pitch and more pitch span can build up much more tension.

		Structure↑	Thematic↑	Richness↑	Overall↑
Dataset	LMD [40]	3.18 (±0.64)	3.07 (±0.58)	3.14 (±0.40)	3.00 (±0.57)
	POP909 [41]	4.06 (±0.49)	3.83 (±0.54)	4.00 (±0.61)	4.11 (±0.46)
Method	Music Transformer [3]	3.00 (±0.76)	3.21 (±0.74)	2.11 (±0.46)	2.32 (±0.54)
	MELONS [8]	3.18 (±0.64)	3.07 (±0.58)	2.64 (±0.64)	2.89 (±0.52)
	POP909_lm	2.61 (±0.62)	2.93 (±0.60)	2.96 (±0.45)	2.96 (±0.49)
	MeloForm	**3.68** (±**0.35**)	**3.57** (±**0.36**)	**3.43** (±**0.43**)	**3.61** (±**0.34**)

Table 1. Subjective evaluation results, with mean opinion scores and 95% confidence interval for each metric.

4. EXPERIMENTAL RESULTS

In this section, we first describe the dataset and system configurations. Next, we show the main results compared with baseline systems. Then we implement some method analysis to further validate the effectiveness of each design. Finally we implement some extensions to demonstrate the scalability of this system. Music samples generated by MeloForm are available via this link [5] . Our code and Supplementary materials are available via this link [6] .

4.1 Experiment Setup

Dataset. We utilize the LMD-matched MIDI dataset [40] in the training stage of the Transformer based neural networks for melody refinement. We select melodies in 4/4 time signature, and normalize them to the tonality of "C major" or "A minor". At last, we obtain 30,218 MIDI samples. This dataset contains 471,058 phrases, in which there exists 100,948 distinct set of similar phrases based on our phrase boundary detection and similarity calculation, of which the detailed algorithm can be found in Supplementary Materials Section 4.

System Configurations. The encoder-attention-decoder Transformer has 4 encoder and decoder layers, both of which contains 4 attention heads. The hidden size is set as 256. We use Adam optimizer [42] with the learning rate is 0.0005. The dropout rate is set as 0.2 during training. There contains maximum of 4,096 tokens in each batch. The dataset is randomly splitted into training/valid/test set with the ratio of 0.8, 0.1, and 0.1, respectively. We use nucleus sampling [43] with the p set as 0.9.

Subjective Evaluation Metrics. For subjective listening, we invite 10 participants to evaluate 30 samples. There are five randomly selected samples from MeloForm and baselines described in Section 4.2. For each listener, they are randomly ordered to avoid perceptual bias and habituation effect. The rating is based on five-point scale. We define the following four metrics: 1) Structure: Does this melody have complete structure with repetitions and variations? 2) Thematic: Can you figure out the musical theme? 3) Richness: Is the melody similarly melodious as human compositions? 4) Overall: Is the overall quality of the melody the same as human compositions?

[5] `https://ai-muzic.github.io/meloform/`

[6] `https://github.com/microsoft/muzic/tree/main/meloform`

4.2 Main Results

We compare MeloForm with the following baselines: 1) Music Transformer [3], an end-to-end neural network solution which introduces the relative attention to learn long-term repetitive patterns; 2) MELONS [8], a Linear Transformer and structure graph based framework for generating melody with bar-level structure; 3) POP909_lm, a language model implemented by ourself for leveraging the phrase labels in POP909 [41] to generate melodies given musical form, whose details can be found in Supplementary section 6. These baselines share the same system configurations with MeloForm for fair comparison.

Table 1 shows the subjective evaluation results in the mean opinion scores (MOS) and 95% confidence interval for these four metrics over all experimental groups. Comparing with baseline systems, MeloForm outperforms all of them in thematicness, structureness, richness and overall quality. We calculate the controlling accuracy of phrase labels for POP909_lm to verify if same phrase labels results in similar melodies. The high accuracy of 100% demonstrates it is capable of generating corresponding repetitive melodies for same phrase labels. It is worth noticing that although POP909_lm can realize the repetitions of phrases, it still fails to provide good listening experience comparing with MeloForm. This demonstrate simply repeating phrases is still deficient. Besides, the less 95% confidence interval of MeloForm provides more confidence for its better performance. We also compare MeloForm with human composed data from the LMD-matched MIDI dataset and POP909. MeloForm shows better performance compared with LMD-matched MIDI dataset, since the samples from LMD-matched MIDI dataset are collected from various kinds of publicly-available sources on the internet, which is hard to guarantee the production quality. In contrast, POP909 is constructed by hiring professional musicians for creating the samples and reviewing the whole generation process, which ensures the quality to be the same as realistic compositions. The comparing results with POP909 reveal that there still exist gaps between melodies generated by MeloForm with the human composed ones.

4.3 Method Analysis

Controllability Study. We calculate the controlling accuracy of musical form by calculating the similarity score between generated melodies from same phrases. The accu-

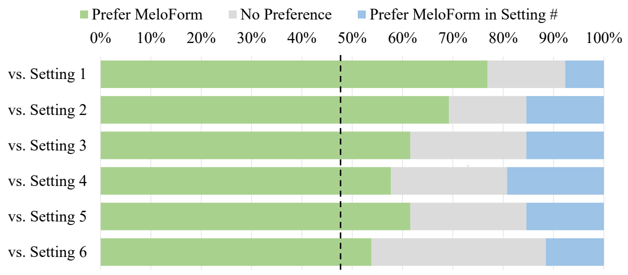


Figure 3. Preference distribution for ablation study, which compares the melodies generated by MeloForm with that from modified system (i.e., MeloForm in Setting #).

racy of 97.79% indicates MeloForm can generate melodies with precise musical form control. We also calculate the accuracy to verify if the pitch distribution is well controlled by average pitch and pitch span. The accuracy of 92.5% in average pitch controlling and 99.71% in pitch span controlling demonstrate its controllability to differentiate sections in pitch dimension.

Ablation Study. We conduct preference tests to verify the contribution of the components in MeloForm with the following settings: 1) Setting 1: w/o development strategies from expert systems. Notes are randomly arranged in expert systems without any development strategies. 2) Setting 2: w/o expert systems. Melodies are first generated for each phrase by neural networks, and then got copied directly with the given musical form for repetition in the same phrase. 3) Setting 3: w/o neural networks. We remove melody refinement neural networks. 4) Setting 4: w/o fine-grained rhythm pattern condition. The condition of rhythm patterns changed from fine-grained level to coarse-grained level. 5) Setting 5: w/o refinement strategy. We directly copy the generate melodies from each phrase to the following same phrases. 6) Setting 6: w/o section differentiation. The “AVGPITCH” and “SPAN” tokens are removed from the beginning of each phrase. The participants are required to compare the samples from MeloForm and these settings and give their preference score.

The preference distribution for the above settings is shown in Figure 3. We derived some observations in the following: 1) More preferences over setting 1 and 2 demonstrate the effectiveness of the whole expert system and the development strategies for building up phrases. 2) More preferences over setting 3, 4, 5, and 6 validate the efficacy of the whole neural refinement model, the rhythm pattern condition, the iterative refinement strategy, and the section differentiation method.

4.4 Extensions to More Musical Forms

In this section, we illustrate the principles to generate melodies with four different musical forms (i.e., Verse and Chorus, Rondo, Variational and Sonata form) based on MeloForm. The examples from these extensions are shown in the demo page.

Verse and Chorus Form is widely used in popular music. It contains two contrasting sections: verse section (i.e., A) and chorus section (i.e., B), which can be arranged in various kinds of ways, such as $AABAAB$, ABA, or $AABB$. Users can follow the method in Section 3.1 to build the verse and chorus sections, and leverage the section differentiation in Section 3.2.3 to change the tension. For example, Chorus tends to have more intensive emotional expression. To achieve this, we can control the average pitch to higher level or increase the pitch span for this section.

Rondo Form is a musical form where the refrained section alternates with contrasting sections, such as $ABACADA$. Melody of Rondo Form can be generated by leveraging the method in Section 3.1 to construct the refrained section and contrasting sections, and arrange them sequentially based on the given Rondo Form.

Variational Form is a musical form where the main section is followed by its variations. We represent Variational Form as $AA'A''$ to describe the main section and its variations. To address the challenge of deriving variations from the main section, we can treat the main section as a motif, and use the development strategies in 3.1.2 to develop this longer motif into another higher-level section. Another way is to generate melodies with same sections like AAA by expert systems, and predict the melodies from each section in different iteration steps. There are many other ways for creating the Variational Form, so users are encouraged to provide their own creative design.

Sonata Form is generally consisted of three sections: an exposition, a development, and a recapitulation, which can be represented as ABA' in our system. For the exposition, we generate two phrases with contrasting motifs, where the second phrase is a transposition of the first one by controlling the tonality token when refining the second phrase. The development section is a variation of the exposition, which can be generated by leveraging the methods for constructing variational form. The recapitulation is a repetition of the exposition, in which the second phrase should go back to the tonic key by controlling the tonality token in condition from neural networks.

5. CONCLUSIONS

In this paper, we propose MeloForm, a system to generate melody with musical form based on expert systems and neural networks. It combines the advantages from expert systems to precisely control musical form and neural networks to refine melody for better musical richness without changing musical form. Experimental results demonstrate MeloForm achieves 97.79% accuracy in musical form control, and outperforms baseline systems in structure, thematic,richness and overall quality in terms of subjective evaluation. We will release the dataset of the synthetic melodies generated by our expert system and the refined version from our Transformer model to facilitate the future research on music form modeling. Furthermore, we will explore the generation of intro, outro and bridge, and investigate the arrangement generation with musical form to accomplish a complete music composition.

6. REFERENCES

[1] W. E. Caplin and W. E. Caplin, *Analyzing classical form: an approach for the classroom*. Oxford University Press, 2013.

[2] J. Wu, C. Hu, Y. Wang, X. Hu, and J. Zhu, "A hierarchical recurrent neural network for symbolic melody generation," *IEEE transactions on cybernetics*, vol. 50, no. 6, pp. 2749–2757, 2019.

[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] Y.-J. Shih, S.-L. Wu, F. Zalkow, M. Muller, and Y.-H. Yang, "Theme transformer: Symbolic music generation with theme-conditioned transformer," *IEEE Transactions on Multimedia*, 2022.

[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] Z. Ju, P. Lu, X. Tan, R. Wang, C. Zhang, S. Wu, K. Zhang, X. Li, T. Qin, and T.-Y. Liu, "Telemelody: Lyric-to-melody generation with a template-based two-stage method," *arXiv preprint arXiv:2109.09617*, 2021.

[7] X. Zhang, J. Zhang, Y. Qiu, L. Wang, and J. Zhou, "Structure-enhanced pop music generation via harmony-aware learning," *arXiv preprint arXiv:2109.06441*, 2021.

[8] Y. Zou, P. Zou, Y. Zhao, K. Zhang, R. Zhang, and X. Wang, "Melons: generating melody with long-term structure using transformers and structure graph," *arXiv preprint arXiv:2110.05020*, 2021.

[9] J. Wu, X. Liu, X. Hu, and J. Zhu, "Popmnet: Generating structured pop music melodies using neural networks," *Artificial Intelligence*, vol. 286, p. 103303, 2020.

[10] S. Dai, Z. Jin, C. Gomes, and R. B. Dannenberg, "Controllable deep melody generation via hierarchical music structure representation," *arXiv preprint arXiv:2109.00663*, 2021.

[11] J. Zhao and G. Xia, "Accomontage: Accompaniment arrangement via phrase selection and style transfer," *arXiv preprint arXiv:2108.11213*, 2021.

[12] H. Young, "A categorial grammar for music and its use in automatic melody generation," in *Proceedings of the 5th ACM SIGPLAN International Workshop on Functional Art, Music, Modeling, and Design*, 2017, pp. 1–9.

[13] D. Quick and P. Hudak, "Grammar-based automated music composition in haskell," in *Proceedings of the first ACM SIGPLAN workshop on Functional art, music, modeling & design*, 2013, pp. 59–70.

[14] M. Kikuchi and Y. Osana, "Automatic melody generation considering chord progression by genetic algorithm," in *2014 Sixth World Congress on Nature and Biologically Inspired Computing (NaBIC 2014)*. IEEE, 2014, pp. 190–195.

[15] A. Garay Acevedo, "Fugue composition with counterpoint melody generation using genetic algorithms," in *International symposium on computer music modeling and retrieval*. Springer, 2004, pp. 96–106.

[16] K. Wakui, Y. Hatori, and Y. Osana, "Automatic melody generation considering chord progression using genetic algorithm," *IEICE Proceedings Series*, vol. 48, no. B2L-E-7, 2016.

[17] M. Takano and Y. Osana, "Automatic melody generation considering user's evaluation using interactive genetic algorithm," in *Asia-Pacific Conference on Simulated Evolution and Learning*. Springer, 2014, pp. 785–797.

[18] Z. Guo, M. Dimos, and H. Dorien, "Hierarchical recurrent neural networks for conditional melody generation with long-term structure," *arXiv preprint arXiv:2102.09794*, 2021.

[19] Y. Yu, A. Srivastava, and S. Canales, "Conditional lstm-gan for melody generation from lyrics," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 1, pp. 1–20, 2021.

[20] A. Mishra, K. Tripathi, L. Gupta, and K. P. Singh, "Long short-term memory recurrent neural network architectures for melody generation," in *Soft Computing for Problem Solving*. Springer, 2019, pp. 41–55.

[21] Y. Yu, F. Harscoët, S. Canales, G. Reddy M, S. Tang, and J. Jiang, "Lyrics-conditioned neural melody generation," in *International Conference on Multimedia Modeling*. Springer, 2020, pp. 709–714.

[22] 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.

[23] S. Li, S. Jang, and Y. Sung, "Automatic melody composition using enhanced gan," *Mathematics*, vol. 7, no. 10, p. 883, 2019.

[24] Z. Sheng, K. Song, X. Tan, Y. Ren, W. Ye, S. Zhang, and T. Qin, "Songmass: Automatic song writing with pre-training and alignment constraint," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 15, 2021, pp. 13 798–13 805.

[25] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1180–1188.

[26] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5156–5165.

[27] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs," *arXiv preprint arXiv:2101.02402*, 2021.

[28] 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.

[29] X. Wu, C. Wang, and Q. Lei, "Transformer-xl based music generation with multiple sequences of time-valued notes," *arXiv preprint arXiv:2007.07244*, 2020.

[30] C. Zhang, L. Chang, S. Wu, X. Tan, T. Qin, T.-Y. Liu, and K. Zhang, "Relyme: Improving lyric-to-melody generation by incorporating lyric-melody relationships," *arXiv preprint arXiv:2207.05688*, 2022.

[31] A. Lv, X. Tan, T. Qin, T.-Y. Liu, and R. Yan, "Re-creation of creations: A new paradigm for lyric-to-melody generation," *arXiv e-prints*, pp. arXiv–2208, 2022.

[32] H. Jhamtani and T. Berg-Kirkpatrick, "Modeling self-repetition in music generation using generative adversarial networks," in *Machine Learning for Music Discovery Workshop, ICML*, 2019.

[33] D. Herremans and E. Chew, "Morpheus: automatic music generation with recurrent pattern constraints and tension profiles," in *Proceedings of IEEE TENCON,-2016 IEEE Region 10 Conference*. IEEE, 2016, pp. 282–285.

[34] S. A. Hedges, "Dice music in the eighteenth century," *Music & Letters*, vol. 59, no. 2, pp. 180–187, 1978.

[35] P. Wiriyachaiporn, K. Chanasit, A. Suchato, P. Punyabukkana, and E. Chuangsuwanich, "Algorithmic music composition comparison," in *2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE, 2018, pp. 1–6.

[36] A. Pati, S. Gururani, and A. Lerch, "dmelodies: A music dataset for disentanglement learning," *arXiv preprint arXiv:2007.15067*, 2020.

[37] B. Bozhanov, "Computoser-rule-based, probability-driven algorithmic music composition," *arXiv preprint arXiv:1412.3079*, 2014.

[38] A. Elowsson and A. Friberg, "Algorithmic composition of popular music," in *The 12th international conference on music perception and cognition and the 8th triennial conference of the European society for the cognitive sciences of music*, 2012, pp. 276–285.

[39] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "Mass: Masked sequence to sequence pre-training for language generation," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5926–5936.

[40] C. Raffel, *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching*. Columbia University, 2016.

[41] 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.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[43] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv preprint arXiv:1904.09751*, 2019.

TOWARDS ROBUST MUSIC SOURCE SEPARATION ON LOUD COMMERCIAL MUSIC

Chang-Bin Jeon, and Kyogu Lee
Department of Intelligence and Information
Music and Audio Research Group (MARG)
Seoul National University
{vinyne, kglee}@snu.ac.kr

ABSTRACT

Nowadays, commercial music has extreme loudness and heavily compressed dynamic range compared to the past. Yet, in music source separation, these characteristics have not been thoroughly considered, resulting in the domain mismatch between the laboratory and the real world. In this paper, we confirmed that this domain mismatch negatively affect the performance of the music source separation networks. To this end, we first created the out-of-domain evaluation datasets, *musdb-L* and *XL*, by mimicking the music mastering process. Then, we quantitatively verify that the performance of the state-of-the-art algorithms significantly deteriorated in our datasets. Lastly, we proposed *LimitAug* data augmentation method to reduce the domain mismatch, which utilizes an online limiter during the training data sampling process. We confirmed that it not only alleviates the performance degradation on our out-of-domain datasets, but also results in higher performance on in-domain data.

1. INTRODUCTION

Recent commercial music has extreme loudness compared to the past [1, 2]. Since many artists and producers want their music to be perceptually louder, it has become so trendy that it rather harms the quality of music [3] and even there exists the expression 'loudness war' [4].

A dynamic range compressor [5] is used to increase the loudness of music while keeping the digital level under 0 decibels relative to full scale (dBFS), which is the maximum possible level in the digital domain. It is a time-varying non-linear processor that adjusts the level of the signal when the level exceeds the threshold. Especially, a limiter refers to a dynamic range compressor that strongly compresses the signal with a high ratio parameter above 1:10 and increases the gain of the signal by the headroom obtained through compression.

In the last stage of music production, so-called master-

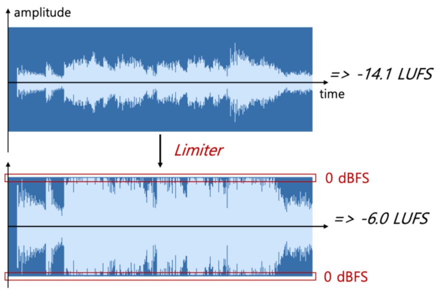


Figure 1. The short example of a limiter applied music source in our *musdb-XL* dataset. Recent commercial music has this loud volume and distorted signal characteristics.

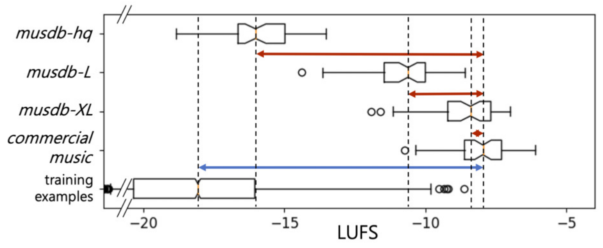


Figure 2. The boxplot representing the loudness distributions of different evaluation datasets, recent commercial music, and training examples generated from the official implementation of *Open-unmix* [6].

ing, engineers are often asked to increase the overall loudness of music. A limiter is a core tool for achieving it; it is used to make small parts of music to be louder and loud parts of music fit under 0 dBFS [7, 8]. In this process, original signals are distorted and become louder. The example of a limiter usage is depicted in Figure 1.

However, these characteristics have not yet been thoroughly considered in the music source separation. Although many data augmentation techniques have been proposed, such as random gain scaling and mixing of different instrumental stem sources [9, 10], the training examples have small overall loudness that is far from real-world commercial music [1], as can be seen in the blue line of Figure 2. This causes a huge domain shift between train and real-world domains. The term LUFS in Figure 2 is an abbreviate for loudness unit relative to full scale, which is a

[1] Based on the investigation of 50 songs in *'Pop Life playlist'* created by Tidal, May. 2022.

measure of music loudness [11, 12].

Furthermore, since the standard benchmark datasets for music source separation, *musdb* [13] and *musdb-hq* [14], do not reflect the characteristics of loud commercial music as shown in the red lines of Figure 2, they are unable to show performance degradation comes from the domain shift. Therefore, we conjecture that even good-performing networks based on *musdb* test subset may exhibit worse performance on real world. This problem is no exception to other datasets [15–17] for music source separation, since they do not consider the music mastering process.

Based on this question, we investigated how heavily compressed music and its loudness affect the degradation of music source separation networks. To this end, we manually built new evaluation datasets by applying a limiter to the *musdb-hq* test subset [14] to get loud and compressed music imitating the modern music mastering process. One is the *musdb-L* dataset, with increased loudness to an appropriate degree for pleasant sound following [3]. The other is the *musdb-XL* dataset, which raised the loudness to an excessive degree, as often found in recent popular music. Using our datasets, we confirmed that more dynamic range compression in test domains results in more performance degradation of the networks.

Moreover, we conducted various experiments on the training data creating methods to reduce the domain shift between train and real world data from the perspective of music loudness and heavy compression. We trained music source separation networks using the training examples constructed by *(1)* linear gain increasing, *(2)* the proposed *LimitAug* method which utilizes an online limiter during the sampling process of training examples, *(3)* simple input loudness normalization, and *(4)* the loudness normalization after applying the *LimitAug*. We confirmed that all of the methods not only showed robust performance on out-of-domain *musdb-L* and *musdb-XL* data, but also showed better performance on in-domain *musdb-hq* test dataset than the baseline.

To summarize, our contributions are three-fold.

- We built *musdb-L* and *XL* datasets[2], which have comparable overall loudness to commercial music, for evaluation of music source separation algorithms.
- Using *musdb-L* and *XL*, we quantitatively confirmed that the domain shift causes performance degradation of the state-of-the-art networks that were trained without considering loud and compressed music characteristics.
- We proposed *LimitAug*[3] data augmentation method and experimentally confirmed that it is beneficial to alleviate the domain shift between train data and the *musdb-L* or *XL*.

[2] https://github.com/jeonchangbin49/musdb-XL

[3] https://github.com/jeonchangbin49/LimitAug

dataset	**Loudness [LUFS]**			
	min	max	median	mean (std)
musdb-hq	-18.84	-13.52	-16.02	-15.92 (1.27)
musdb-L	-14.39	-8.61	-10.61	-10.89 (1.19)
musdb-XL	-11.93	-6.99	-8.41	-8.61 (1.17)
commercial	**-10.75**	**-6.10**	**-7.96**	**-8.05 (1.06)**

Table 1. Loudness statistics of *musdb-hq*, *musdb-L*, *musdb-XL* datasets, and the investigated commercial music.

2. MUSDB-L AND MUSDB-XL

musdb [13] and *musdb-hq* [14] have been the standard benchmark datasets on music source separation since their presentation in Signal Separation and Evaluation Challenge (SiSEC) 2018 [18]. They consist of various genres of 150 professionally produced songs — 100 songs for train, 50 songs for test — from folk, indie to electronic, rock genres. Each song has its constituting 4 stems, *vocals*, *drums*, *bass*, and the remaining as *other*. The *musdb-hq* dataset is the uncompressed version of *musdb* with the extended frequency bandwidth from 16kHz to 22.05kHz, which is a full bandwidth of 44.1kHz sample rate. We used *musdb-hq* for high-quality dataset construction.

Since the test subset of *musdb-hq* has small overall loudness compared to commercial mastering-finished music, we conjectured that it could not fully reflect the performance degradation related to heavy dynamic range compression, which prevails in recent commercial music. In addition, to the best of our knowledge, there is no dataset for music source separation that reflects these characteristics or considers the music mastering process. Therefore, we built *musdb-L* and *musdb-XL* datasets, which have loud and compressed characteristics similar to commercial music. *L* and *XL* each stands for *Loud* and *eXtremely Loud*.

2.1 Dataset construction

To reflect the characteristics of commercial music, we created datasets by imitating the music mastering process. Both datasets were made by manually applying the commercial digital limiter, iZotope Ozone 9 Maximizer[4], to the *musdb-hq* test subset. For each *musdb-L* and *musdb-XL*, we controlled the threshold parameter of a limiter so that compression is applied about 3-4 dB and 6-7 dB in loud parts of *mixture* tracks of the *musdb-hq*. As shown in Table 1, *musdb-L* and *musdb-XL* are about 5 and 7 LUFS louder than the original *musdb-hq* dataset on average, respectively. *musdb-L* has insufficient loudness compared to loud commercial music but less distorted sources. *musdb-XL* has comparable loudness to commercial music and more distorted sound than *musdb-L*.

To make the ground truth stem tracks of each mixture, we calculated the sample-wise ratio between the limiter applied mixture and the original mixture, then multiply it to the individual stems to make the ground truth stems for *musdb-L* and *XL*.

[4] https://www.izotope.com/en/products/ozone.html