Suzuki
EURASIP Journal on Audio, Speech, and Music Processing (2023) 2023:52
https://doi.org/10.1186/s13636-023-00321-7

EURASIP Journal on Audio, Speech, and Music Processing

METHODOLOGY

Open Access

Piano score rearrangement into multiple difficulty levels via notation-to-notation approach

Masahiro Suzuki[1*]

Abstract

Musical score rearrangement is an emerging area in symbolic music processing, which aims to transform a musical score into a different style. This study focuses on the task of changing the playing difficulty of piano scores, addressing two challenges in musical score rearrangement. First, we address the challenge of handling musical notation on scores. While symbolic music research often relies on note-level (MIDI-equivalent) information, musical scores contain notation that cannot be adequately represented at the note level. We propose an end-to-end framework that utilizes tokenized representations of notation to directly rearrange musical scores at the notation level. We also propose the ST+ representation, which includes a novel structure and token types for better score rearrangement. Second, we address the challenge of rearranging musical scores across multiple difficulty levels. We introduce a difficulty conditioning scheme to train a single sequence model capable of handling various difficulty levels, while leveraging scores from various levels in model training. We collect commercial-quality pop piano scores at four difficulty levels and train a MEGA model (with 0.3M parameters) to rearrange between these levels. Objective evaluation shows that our method successfully rearranges piano scores into other three difficulty levels, achieving comparable difficulty to human-made scores. Additionally, our method successfully generates musical notation including articulations. Subjective evaluation (by score experts and musicians) also reveals that our generated scores generally surpass the quality of previous rule-based or note-level methods on several criteria. Our framework enables novel notation-to-notation processing of scores and can be applied to various score rearrangement tasks.

Keywords Symbolic music processing, Music rearrangement, Token representation, Musical score

1 Introduction

Musical scores represent music as notation, structuring musical notes and also conveying performance instructions. The structure and instructions in musical scores often go beyond what can be represented in simple note-level forms like MIDI. When playing or practicing a musical instrument, having scores that match an individual's skill level is also crucial. These scores not only ensure that the player can play the piece but also provide an appropriate level of challenge that facilitates skill improvement. However, preparing scores of varying levels of difficulty can require a considerable amount of effort. Therefore, an alternative method of rearranging scores into different levels of difficulty is desirable, because it would benefit both individual players and music education as a whole. In this context, this paper proposes a novel method for rearranging musical notation on scores into various difficulty levels.

In the field of music research, the difficulty of musical scores has been a noted topic of study. Several recent works on piano reduction [1, 2] have considered difficulty when rearranging ensemble scores into solo

*Correspondence:
Masahiro Suzuki
masahiro1.suzuki@music.yamaha.com
[1] Music Informatics Group, R&D Division, Yamaha Corporation, Shizuoka, Japan

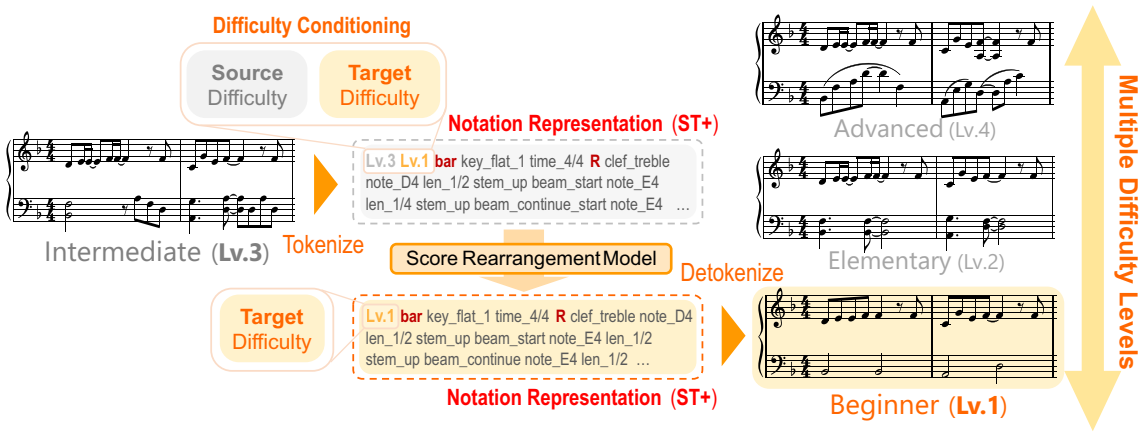

Fig. 1 Overview of notation-level end-to-end score rearrangement framework handling multiple difficulty levels

scores. Difficulty estimation for solo scores has also been researched in various ways [3, 4]. However, the task of rearranging solo scores into different difficulty levels is a relatively new research area and has not been extensively studied. Fukuda et al. [5] have conducted pioneering research using a *rule*-based approach, which simplified scores by removing notes without rearranging them. However, this approach was limited to generating easier scores. More recently, Gover et al. [6] have shown that a data-driven approach can be employed to tackle this task. They showed that a sequence of musical notes can be translated into another sequence of notes with a lower difficulty level. However, the study was limited to *note*-level (MIDI-equivalent) rearrangement of scores. In this study, we extend the data-driven approach to address the score rearrangement task more comprehensively. We process *notation*-level musical information and handle *multiple* difficulty levels to achieve a more thorough approach to score rearrangement.

First, we aim to handle musical information extensively on the *notation* level. There are two types of symbolic music representation used in music research: *note*-level and *notation*-level [7]. The former expresses MIDI-equivalent information (e.g., note timing and pitch), whereas the latter represents musical symbols (e.g., clef, chord symbols and notes) and musical attributes (e.g., beam, tie and slur). Although the former is widely used in music research [8–13], it cannot represent abstract musical information such as playing instructions and note groupings in musical scores, which are indispensable for authentic scores. Recent studies have shown that the latter representations, which can contain such information, are also successfully used with sequence models [7, 14] and that a designed score token representation outperforms notation formats [15]. Based on these findings, we employ and improve the score token representation to rearrange piano scores in the notation domain, handling musical expressions in scores.

Second, we aim to rearrange scores into *multiple* levels of difficulty using a *single* model. Musical scores are commonly assigned discrete difficulty levels, typically ranging from 3 to 9 levels [3, 4]. However, previous studies of score difficulty conversion [5, 6] have only looked at a *single* conversion to an *easier* level. In this study, we address the challenge of converting scores into *multiple* levels of difficulty, including both *easier* and *harder* levels. Handling *multiple* levels within a *single* model can also be beneficial when dealing with limited availability of musical scores. Recent deep learning models typically require a large amount of training data. However, acquiring a large amount of scores is not always feasible. In addition, the complexity of *notation*-level information surpasses that of *note*-level information, making its training more challenging and requiring more scores. To overcome these challenges, we propose a method to train a single model capable of handling various difficulty levels, enabling learning from a limited amount of scores.

Our contributions are summarized as follows:

1. We propose an end-to-end framework for rearranging musical scores on the *notation* domain (Fig. 1), which enables direct conversion of score notation.
2. We devise training schemes for a *single* model capable of rearranging scores into *multiple* difficulty levels.
3. We also propose ST+, an extended score token representation with new token types and improved structure compared to the original ST representation [15].

2 Technical background

In this section, we briefly describe the technical background of the *sequence*-based approach we adopt.

2.1 Symbolic music processing using sequence models

Previous studies in symbolic music processing have shown that sequence models, especially Transformer-based models [16], can effectively process MIDI [8] and musical scores [15] as sequences. These models have consistently exhibited impressive performance across a range of music-related tasks, including music generation [8–13], music style transfer [17], and music transcription [18]. Various Transformer variants have been explored and proved to be effective in these tasks.

Recently, novel approaches combining Transformers with other model architectures (e.g., state space models [19]) have emerged [20, 21], showing even higher performance than Transformers. One such example is the MEGA model [20], which combines the attention mechanism [16] with state space models [19] and achieved high performance in a wide range of tasks [22]. MEGA also achieved strong performance on a machine translation task, which resembles our score rearrangement task, in a sequence-to-sequence setting [20]. Leveraging both the established performance of Transformers in music-related tasks and recent architectural improvements, we adopt the MEGA model to attempt a sequence-to-sequence transformation of musical notation.

2.2 Token representations for symbolic music

To handle symbolic music with sequence models, many proposals for *note*-level (MIDI-equivalent) token representations have been made [8–13], exploring improvements in music modeling performance [8, 10] and processing efficiency [9, 12]. Some representations have also been designed to handle multi-track MIDI information [10–12], using either a *track*-major [11] or *bar*-major [10, 12] structure. In the former, the sequences of each *track* are arranged sequentially, while in the latter, the sequences of each *bar* are connected in series.

In contrast, research on *notation*-level token representations remains scarce and largely unexplored. In this study, we investigate the effective structure of notation-level token representations for rearranging scores, based on the score token (ST) representation [15] proposed in a previous study. ST represents basic musical symbols and attributes in piano scores (the † types in Table 1) and arranges them in a *staff*-major structure, where the sequences of two staves in a piano score are concatenated sequentially (Fig. 3(b)). We also consider a *bar*-major structure for the notation-level representation (Section 3.2) and compare these structures in our experiment.

Table 1 Token symbols in proposed ST+ representation. † types are inherited from ST [15]

Type	Symbol	Example
Structure†	Bar	*bar*
	Staff	*R*
	Voice	*<voice>*
Attribute†	Clef	*clef_treble*
	Key signature	*key_flat_1*
	Time signature	*time_4/4*
Note†	Pitch	*note_Db4*
	Duration	*len_1/2*
	Stem	*stem_up*
	Beams	*beam_start*
	Tie	*tie_start*
	Rest	*rest*
Articulation	Accent	*accent*
	Slur	*slur_start*
	Staccato	*staccato*
	Tenuto	*tenuto*
Chord symbol	Chord	*chord_Cm7*
	Bass	*bass_D*

3 Proposed method

We formulate score rearrangement between difficulty levels as a *sequence-to-sequence* problem and propose an end-to-end framework for rearranging musical scores on the *notation* domain (Fig. 1). We tokenize musical notation of scores into notation-level token sequences (Fig. 3(a)) and train a sequence model to translate their notation between difficulty levels. Our approach includes schemes that enable training across multiple difficulty levels (Section 3.1) and an extended notation-level score representation (Section 3.2). This allows for the direct conversion of musical notation, in a "*notation-to-notation*" manner.

3.1 Multiple difficulty training

We propose training schemes that allow training a *single* sequence model that translates musical notation between *multiple* difficulty levels. Our scheme consists of the following two steps:

3.1.1 Score pairing

To fully leverage available scores, we first make all available score combinations for the same song. We then train a model with paired scores bidirectionally (to easier and harder levels) (Fig. 2(a)). The resulting number of paired score data for a song is ${}_nC_2$, where n is the number of available scores for the song. It significantly increases along with the number of available scores for the song

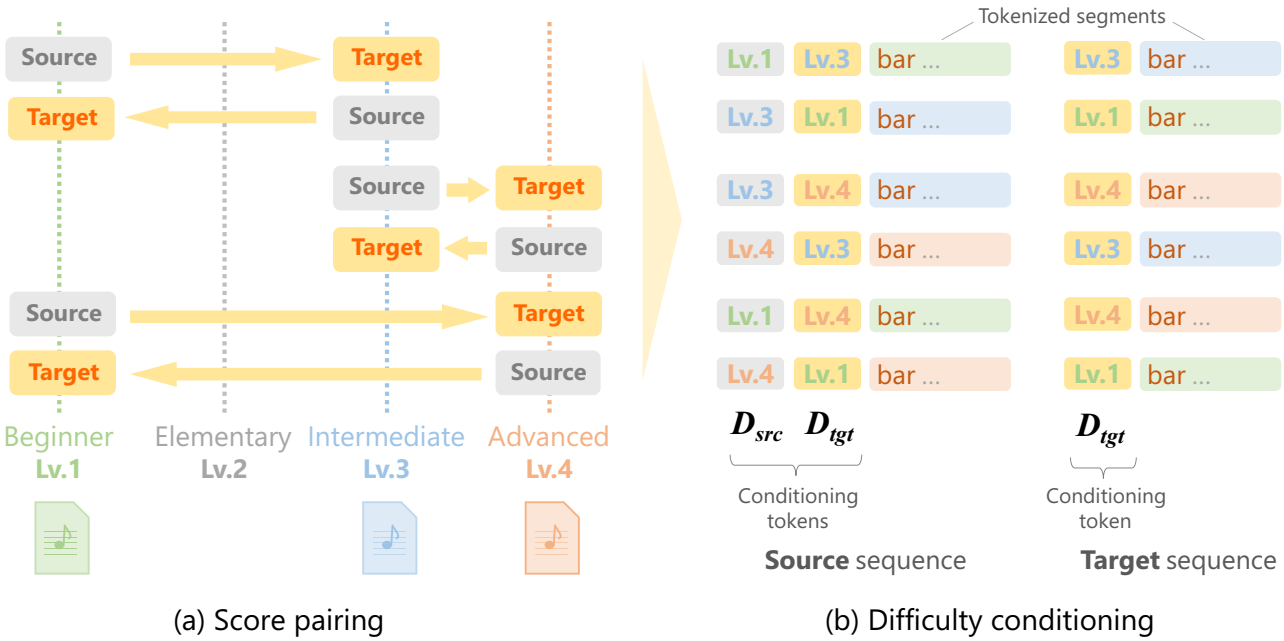

Fig. 2 Illustration of score pairing (Section 3.1.1) and difficulty conditioning (Section 3.1.2) schemes, showcasing a scenario where three levels of scores are available for the same song

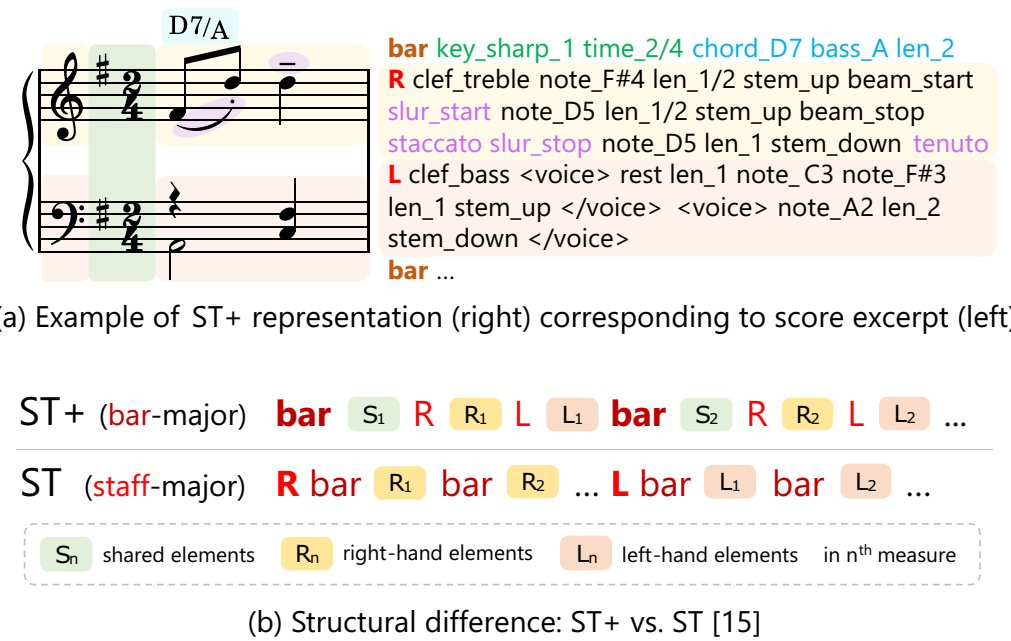

Fig. 3 Proposed ST+ representation (Section 3.2) and its structural difference from original ST [15] representation

and thus facilitates model generalization by increasing training data.

3.1.2 Difficulty conditioning

We also propose a conditioning scheme that enables handling *multiple* difficulty levels in a *single* model. We adopt a recent finding in multilingual translation [23] that facilitates translation between low-resource language pairs. We represent difficulty levels of scores as conditioning tokens (similar to *language* tokens in translation) and prepend them to score sequences. We prepend $\{D_{src}, D_{tgt}\}$ to the *source* sequence, and D_{tgt} to the *target*

sequence (Fig. 2(b)), where D_{src} and D_{tgt} are conditioning tokens that denote difficulty levels of the *source* and *target* scores, respectively. Following these conditioning tokens, score notation is represented using the notation-level token representation described in the next subsection (Section 3.2).

3.2 Extended score token representation (ST+)

We represent musical notation as token sequences. We propose ST+ score token representation, which extends and improves the original ST representation [15] in two aspects:

3.2.1 Articulations and chord symbols

First, we extend ST's expression by introducing new token types for articulations and chord symbols (Table 1), aiming to incorporate a wider range of musical elements into score rearrangement. For articulation, we include four types commonly used in piano scores: *accent*, *slur*, *staccato*, and *tenuto*. Each articulation type is represented by a single token, except for a *slur*, which requires two tokens to indicate its *start* and *end* (Fig. 3(a)). As for chord symbols, we introduce two types of tokens: *chord* and *bass*. The *chord* token represents a root note and a chord quality, whereas the *bass* token represents a bass note (Fig. 3(a)). The latter appears only when a bass note is specified. The subsequent *len* token represents the duration of a chord in the same way as for notes and rests. These chord-related tokens are included as shared elements (Fig. 3(b)) only in the *source* sequences to facilitate efficient inference. All other token types are inherited unchanged from [15] (Table 1).

3.2.2 Bar-major structure

Second, we reorganize tokens from a *staff*-major order to a *bar*-major order (Fig. 3(b)), with the goal of (1) improving score modeling by placing elements in the same bar close together, and (2) sharing common tokens between staves (e.g., time and key signature) for consistency and efficiency. This structural change also allows other musical notation to be efficiently included as shared elements (Fig. 3(b)), making future extensions more convenient.

4 Experimental setup

4.1 Dataset

We created our dataset by collecting piano scores from a commercial sheet music store[1]. We collected 1957 solo pop piano scores on four difficulty levels (102 beginner, 691 elementary, 738 intermediate, and 428 advanced scores), arranged by various human arrangers, with the condition that multiple scores of different difficulty levels are available for the same song. These difficulty levels show relatively consistent trends; for instance, the *beginner* level typically focuses on one note per hand at a time, the *elementary* and *intermediate* levels allow for up to two and three simultaneous notes per hand, respectively, and the *advanced* level has no such limits. In addition, *note density* and *pitch width* (see definitions in Section 5.1.1) also tend to increase as the difficulty level rises (as partially shown in Fig. 4).

To establish a mapping between these scores, we paired the scores of the same song (Section 3.1) and aligned them. The aligned scores were then fragmented together into 4- to 8-measure segments allowing for overlap. We tokenized these segments into sequence pairs (Fig. 2(b)), resulting in a total of 130,930 paired segments. To ensure the matching of keys in the paired segments, we transposed the *source* scores if necessary. With this transposition step, the model (described in Section 4.2) learned key-independent musical transformations. We split the segments song-wise 8:1:1 for the training, validation, and test sets, respectively. To facilitate model generalization, we applied modest pitch augmentation only to the training set, independently of the key-matching step, by transposing both the *source* and *target* scores by the same pitch interval. The pitch interval was limited to the range of -2 to +2 semitones to ensure that model learns to generate pitch-sensitive notation (e.g., clef and stem direction) correctly.

4.2 Model

We employed the MEGA model [20] (Section 2.1) to handle and compare the performance of various token representations. We used the official implementation[2] of the model with an encoder-decoder architecture built on the `fairseq` framework [24]. We used the following small model configuration: embedding sizes $d_{\text{model}} = 48$, $d_{\text{FFN}} = 96$, $v = 96$, and $z = 24$; number of layers is 3 for both the encoder and decoder. The resulting number of parameters was approximately 0.3M, which is less than 1/200 of the "MEGA-base" setting used for a machine translation task [20]. We avoided using the "MEGA-chunk" variant, which incorporates chunk partitioning mechanism, due to frequent grammatical errors.

4.3 Baselines

We utilized three token representations with the MEGA model: ST+, ST [15], and REMI+ [10]. We employed ST as a *notation*-level baseline and REMI+ as a *note*-level baseline. To ensure a fair comparison, we

[1] https://www.print-gakufu.com/

[2] https://github.com/facebookresearch/mega/

Table 2 Comparison of methods and representations employed in our experiment. Bold shows the same approach as ours

	Methodology	Representation	Structure
ST+ (ours)	***sequence***-based	***notation***-level	***bar***-major
ST [15]	***sequence***-based	***notation***-level	*staff*-major
REMI+ [10]	***sequence***-based	*note*-level	***bar***-major
Rule-based [5]	*rule*-based	—	—

represented articulations and chord symbols also in ST, in the same way as in ST+. With this modification, the only difference between ST+ and ST in the experiment lies in their structure (Table 2). Among note-level representations (Section 2.2), we selected REMI+ to utilize its representations for chord symbols, time signatures, and instrument types. We used its instrument-type tokens to represent the two staves (i.e., right and left hands) of piano scores for our purpose. Additionally, we represented a chord as notes on the same timing using REMI+. The resulting *note*-level representation corresponds to that used in a previous study of note-level score rearrangement [6].

We also employed another baseline method referred to as "rule-based" [5], which simplifies scores using a set of rules consisting of three patterns to remove notes. We re-implemented these rules, adjusting the thresholds to match the difficulty levels of our dataset. Table 2 summarizes the methods and representations used in the experiment.

5 Results

To simplify the presentation, we show the results for the typical cases in which *intermediate* scores are rearranged into the remaining three difficulty levels (as illustrated in Fig. 1). We used all 3810 segments that conform to these cases in the test set for objective evaluation (Section 5.1) and a random selection from the same subset for subjective evaluation (Section 5.2).

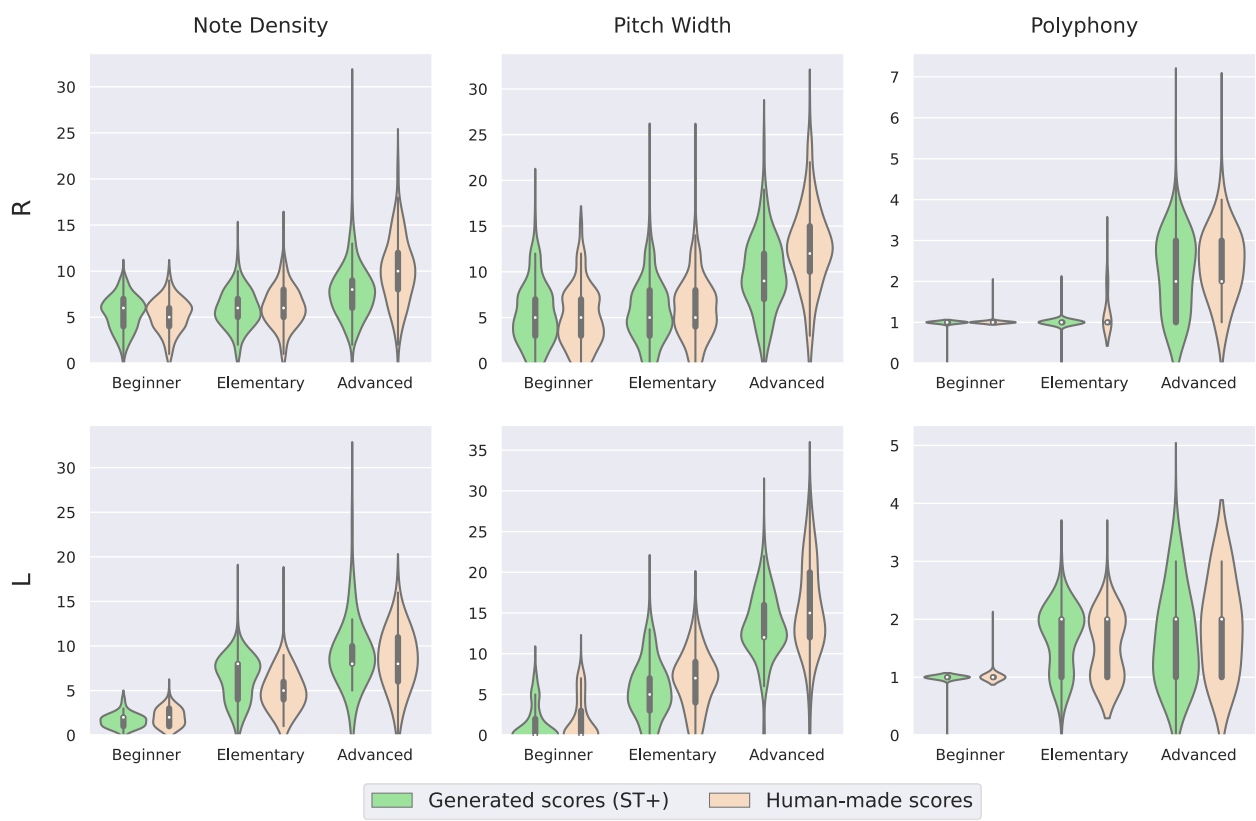

Fig. 4 Quantitative difficulty evaluation with three metrics (Section 5.1.1) for right-hand staff (top) and left-hand staff (bottom)

Table 3 JS divergence from human-made scores averaged over all the difficulty metrics. Bold indicates lowest JS divergence, i.e. closest metric distributions to human-made scores

	Beginner	Elementary	Advanced	Average
ST+ (ours)	**.028**	**.054**	**.088**	**.057**
ST [15]	.031*	**.054**	.116**	.067**
REMI+ [10]	.034**	.078**	.106**	.073**
Rule-based [5]	.054**	.067**	—	.061*

* and ** denote $p < 0.05$ and $p < 0.01$, respectively, vs. ST+

Table 4 Rates of notes (or chords) with articulations (%) for notation-level representations. Bold shows closest value to human-made scores

	Beginner	Elementary	Advanced
ST+ (ours)	**0.00**	**0.27**	**16.21**
ST [15]	**0.00**	0.08***	12.57***
Human (ref.)	0.00	0.60	17.33

*** denotes $p < 0.001$ vs. ST+

Table 5 Sequence-wise error rates (%) for three token representations

	Syntax error	Structure error
ST+ (ours)	**0.00**	0.76
ST [15]	**0.00**	0.92
REMI+ [10]	0.03	**0.53**

5.1 Objective evaluation

5.1.1 Difficulty

Referring to a previous study on score reduction considering difficulty [2], we employed the following three simple metrics for assessing difficulty quantitatively:

- *Note density*: the number of notes in a measure;
- *Pitch width*: the semitone range between the highest and lowest pitches in a measure;
- *Polyphony*: the maximum number of simultaneous notes in a measure.

We calculated these metrics for the *right-hand* and *left-hand* staff and compared the distributions of their metric values across different difficulty levels.

Figure 4 shows the distributions of metric values for both the generated and ground-truth (human-made) scores on each difficulty level. We can see that the metric value distributions of the generated scores (ST+) exhibit a shift towards higher values as the difficulty levels increase (from left to right). Additionally, many of these distributions resemble those of the human-made scores at the corresponding difficulty levels. These observations indicate that our proposed method effectively controlled the difficulty levels of the generated scores and successfully generated scores that were comparable in difficulty to human-made scores across three targeted difficulty levels.

Table 3 presents the aggregated results by averaging the Jensen-Shannon (JS) divergence values between the distributions of the generated and human-made scores across the employed metrics. Bootstrap testing (one-sided, 1000 iterations) reveals that our ST+ attained significantly lower JS divergence values than other methods across all three difficulty levels and their average, except for ST+ vs. ST on *elementary*, where both performed equally well. The result indicates that ST+ was able to generate scores that most closely resemble the difficulty levels of human-made scores.

5.1.2 Articulations

Articulations are one of the characteristics of score notation that can only be handled by notation-level representations. Table 4 shows the observed rates of articulation, indicating that the trained models successfully generated articulations. The rates of articulation varied with difficulty in a manner similar to human-made scores, demonstrating the successful control of articulation generation using our method. When comparing the two representations, ST+ achieved significantly closer rates ($p < 0.001$, z-test) to the human scores at all difficulty levels (including *beginner*, where two representations performed equally well), suggesting that ST+ is superior to ST in accurately emulating the quantitative tendency of human notation. We also present the example of generated articulations later in Section 5.3.

5.1.3 Output validity

The trained sequence model should also generate grammatically correct and properly structured sequences in order to rearrange scores validly. Table 5 shows two types of error rates observed in sequence-based models:

- *Syntax error*: grammatical errors related to token ordering;
- *Structure error*: disagreement in the number of measures between staves (for right and left hands) or between the input and output.

Overall, *syntax* errors were quite seldom (only found in a REMI+ segment) and *structure* errors were also infrequent in all representations. The error rates did not differ

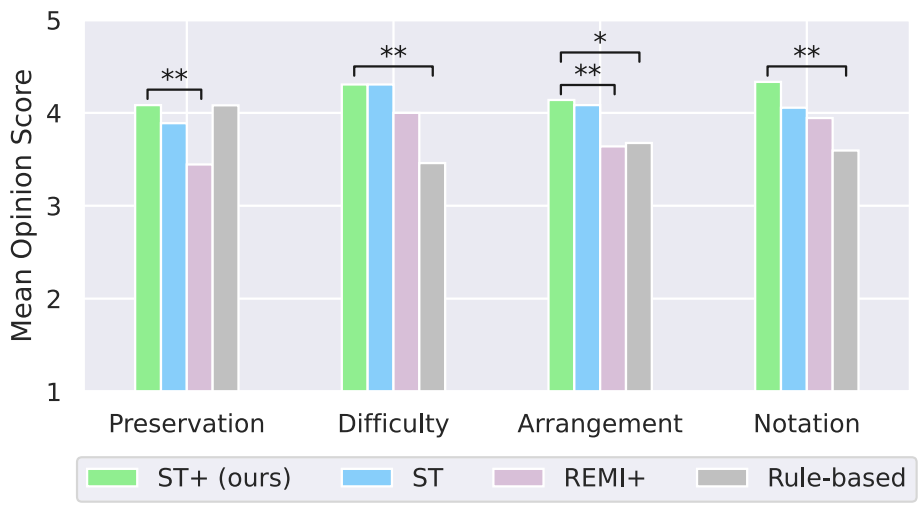

Fig. 5 Mean opinion scores (MOS) of all methods averaged over beginner and elementary levels. * and ** denote $p < 0.05$ and $p < 0.01$, respectively; shown only vs. ST+

significantly between representations on z-tests, except for ST vs. REMI+ ($p < 0.05$) for *structure* errors. When comparing ST+ and REMI+, although ST+ involves a greater variety and complexity of tokens, its error rates were not significantly larger, implying that ST+ could be used without compromising output validity compared to note-level counterparts.

5.2 Subjective evaluation

We also conducted a subjective evaluation using randomly selected 18 samples (6 samples each from 3 levels) from the test set. For each sample, the scores rearranged by different methods were presented in a random order alongside the original (*source*) score and rated by 9 participants (6 score experts and 3 musicians) on five-point scales ranging from 1 (poor) to 5 (good). We used the following four criteria for the evaluation:

- *Preservation*: Are the melody and chords well preserved?
- *Difficulty*: Does the difficulty change adequately?
- *Arrangement*: Is the score playable and naturally arranged?
- *Notation*: Is the score readable and well-formatted, including articulations?

Prior to the score evaluation, participants were screened to ensure their ability to read scores. All scores were rendered by MuseScore 3[3], an open-source music notation software, and were eliminated clues to infer employed methods.

The results are presented in two parts (in Fig. 5 and Table 6) because the rule-based method [5] is limited to generating scores on easier difficulty levels (i.e., *beginner* and *elementary*). We compare the proposed method with the rule-based method in the former part, and with other methods in the latter part. We use a one-sided Welch's t-test to test for significance.

Figure 5 shows mean opinion scores (MOS) averaged over the easier levels. Although ST+ and the rule-based method perform equally on *preservation*, ST+ outperforms the rule-based method on *difficulty* and *notation* ($p < 0.01$) as well as *arrangement* ($p < 0.05$), suggesting that our method generated more appropriate scores than the rule-based method in many aspects.

Table 6 shows the overall MOS for all difficulty levels, where ST+ outperformed REMI+ ($p < 0.01$) on *preservation* and *arrangement* and also had higher scores on other criteria. The result suggests that our method (ST+) rearranged scores with quality better than the *note*-level method (REMI+) on several criteria. When comparing ST+ and ST, although not statistically significant, ST+ had higher scores than ST

Table 6 Mean opinion scores (MOS) averaged over all three levels of difficulty

	Preservation	Difficulty	Arrangement	Notation
ST+ (ours)	**4.00**	4.00	**4.00**	**4.09**
ST [15]	3.81	**4.02**	3.98	3.93
REMI+ [10]	3.41**	3.74	3.52**	3.87

** denotes $p < 0.01$ vs. ST+

[3] https://musescore.org/

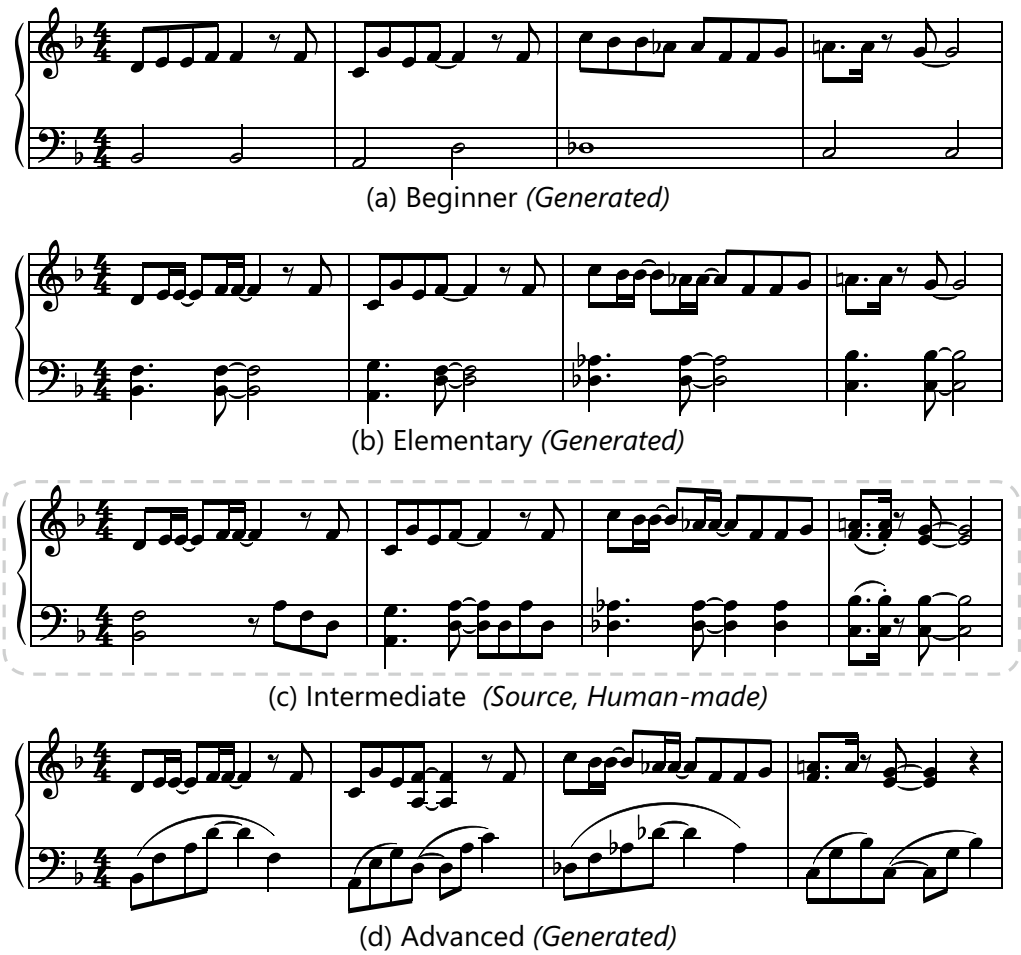

Fig. 6 Example of rearranged piano scores from a intermediate-level score (c) into other three difficulty levels (a, b, d), which were generated by the model trained with ST+ representation

on some criteria (especially, on *preservation* and *notation*), implying that the *bar-major* structure in ST+ (Fig. 3) is a promising alternative to the *staff-major* structure [15].

5.3 Rearrangement example

Figure 6 shows the rearrangement example of the proposed method. We can see that the method generated valid scores with different playing difficulties while preserving the musical contexts of the original score (Fig. 6(c)). We observe that the generated scores are completely rearranged, rather than simply by deleting or adding notes. We also see the articulations (*slurs*) properly generated in Fig. 6(d). The characteristics of the generated scores are generally consistent with those observed in the original dataset (Section 4.1).

6 Ablation studies

In this section, we look at some ablation studies to see what factors are effective in training. We used the proposed ST+ token representation for all ablations and evaluated the same pairs of difficulty levels as in Section 5. We also used z-tests for significance testing between conditions.

Table 7 Structure error rates (%) when training with multiple difficulty pairs vs. single difficulty pair

	Multiple difficulty pairs (ours)	**Single difficulty pair**
Beginner	**1.00*****	83.60***
Elementary	**0.93*****	32.36***
Advanced	**0.35*****	44.43***

*** denotes $p < 0.001$

Table 8 Error rates (%) when training with or without pitch augmentation

	w/ augmentation	*w/o* augmentation
Syntax error	**0.00**	**0.00**
Structure error	**0.76*****	27.17***

*** denotes $p < 0.001$

Table 9 Reproduced rates (%) of chord constituent notes when training with or without chord symbol representations. Bold shows closest value to human-made scores

	w/ chord symbols	*w/o* chord symbols	*Human (ref.)*
Root or bass	**95.0*****	92.2***	95.6
All	**70.8*****	69.6***	72.8

*** denotes $p < 0.001$

6.1 Multiple difficulty training

First, to evaluate the effectiveness of our training schemes (Section 3.1), we trained separate models for each difficulty level without utilizing our schemes. These models were trained under the following conditions: (1) no difficulty conditioning was applied; (2) only scores from a single difficulty pair were used, such as *beginner-intermediate* pair for training the *beginner* model. We used the difficulty pair in one direction only (e.g., *intermediate* to *beginner*) because bi-directional training is only possible in the presence of a conditioning scheme. For convenience, we refer to this training method as *single-pair* and our proposed one as *multi-pair*. The pitch augmentation procedure (Section 4.1) was used to isolate the effect of difficulty conditioning and score pairing.

Table 7 clearly shows that *single-pair* training leads to high error rates, making the resulting output unusable. The error rates were significantly lower when employing *multi-pair* training ($p < 0.001$), demonstrating the effectiveness of our training scheme in avoiding basic errors. This result suggests that even with an insufficient amount of score data for *single-pair* training, our *multi-pair* training enables sufficient generalization of the model in terms of structural validity.

6.2 Pitch augmentation

Next, we trained the integrated model for multiple difficulty levels without the pitch augmentation procedure (Section 4.1) to investigate the effectiveness of the procedure.

Table 8 shows the effectiveness of the augmentation procedure. While syntax errors did not occur even without augmentation, structure errors were frequently observed when the model was trained without augmentation ($p < 0.001$). The result suggests that the augmentation procedure also promotes the generalization of the model in terms of structural validity across various musical lengths of input.

The finding also provides insight into the previous ablation study (Section 6.1). Without pitch augmentation, which reduced the number of training data, there was a significant increase in structure errors, similar to single-pair training. This suggests that the increase in errors observed in single-pair training was also attributable to the reduction in the number of training data.

6.3 Chord symbols

Finally, to evaluate the effectiveness of chord symbol representation (Section 3.2.1), we compared the model's performance in generating scores that align with the harmonic structure of the original (*source*) scores. We trained and inferred both with and without chord symbol representations and evaluated using two criteria:

- *Root or bass*: whether the root or bass note[4] of each chord is reproduced as a note;
- *All*: whether each constituent note of each chord is reproduced as a note[5].

Table 9 shows that including chord symbol representations improves the reproduction rates of chord constituents on both criteria ($p < 0.001$). These results indicate that chord symbol representations in the proposed ST+ representation work effectively, as they facilitate the consideration of harmonic structure during score rearrangement.

7 Discussion

In this section, we summarize our findings in terms of three key points, discussing extensions and limitations.

7.1 Notation-level score rearrangement

Our *notation*-level method (Section 3) generally yields favorable outcomes compared to existing *note*-level or *rule*-based methods in both the objective (Section 5.1) and subjective (Section 5.2) evaluation. Additionally, our method successfully handles articulations (Sections 5.1.2 and 5.3). These findings reveal that (1) data-driven score rearrangement is possible not only at the *note* level [6] but also at the *notation* level, and (2) the *notation*-level

[4] We checked for the specified bass note if present (e.g., D7/A), or otherwise checked for the root note.

[5] For simplicity and clarity, we checked for all constituent notes, although not all of them need to be expressed as notes to represent a chord.

approach can generate better rearrangements than *note*-level [6] or *rule*-based [5] ones. Our proposed approach enables a comprehensive rearrangement of musical scores, effectively handling notation-specific symbols and instructions. Future research will reveal which aspects of the *notation*-level approach (or representation) are particularly influential.

7.2 Multiple difficulty levels

Our training schemes (Section 3.1) are effective in handling *multiple* difficulty levels within a *single* model (Section 5.1.1). The schemes also contribute to reducing errors significantly (Section 6.1). These results highlight the effectiveness of our training schemes, revealing that a *single* model can successfully handle *multiple* types of musical scores using these schemes. This approach could also be applied to score rearrangement in other musical aspects, such as musical style or instrumentation, where considering multiple types of scores is desirable. From a difficulty perspective, incorporating multi-faceted or continuous difficulty conditioning will also be meaningful future extensions.

7.3 ST+ representation

New token types introduced in ST+ (Section 3.2.1) enable the successful handling of articulations (Sections 5.1.2 and 5.3) and consideration of chord symbols (Section 6.3) in score rearrangement. Additionally, the *bar*-major structure in ST+ (Section 3.2.2) contributed to superior quantitative results to the *staff*-major structure in ST [15] (Sections 5.1.1 and 5.1.2). Although the qualitative evaluation did not reveal a significant difference, it also show promising results for the *bar*-major structure. Overall, ST+ can serve as a better representation than ST in score rearrangement. ST+ could be further extended to represent other musical symbols not included in this study.

7.4 Limitations and extensions on difficulty changing

From the perspective of comprehensive score difficulty conversion, particularly into the entry-level scores, there are other aspects to consider: score shortening, performance aids (such as fingerings), and transposition. Although this study did not address these aspects, they can be treated rather independently from the rearrangement of score notation addressed in this study. For instance, score shortening and performance aids can be treated as tasks related to music structure analysis [25, 26] or fingering estimation [27–30], respectively. Additionally, transposition could be considered by employing a relatively simple key-change algorithm. The transposed scores could be handled robustly by our model because it was already trained on the dataset where the source scores were also partially transposed (Section 4.1). By performing these tasks independently in the pre-processing or post-processing stage of our score rearrangement, a more comprehensive score difficulty conversion could be achieved.

8 Conclusion

We proposed a novel *notation*-level score rearrangement method with a *single* sequence model for *multiple* difficulty levels. Our method successfully rearranges scores into various levels of difficulty in the form of *notation*, uniquely handles articulations through *notation*-level rearrangement. Our evaluations reveal that our method generally produces more appropriate scores than existing *note*-level or *rule*-based approaches. We also introduced the ST+ representation, which contributes to better quantitative results and can be a better alternative to ST [15]. Our framework enables direct *notation-to-notation* rearrangement, which provides a novel way to process musical scores and is also applicable to other score rearrangement tasks (e.g., style transfer and instrumentation).

Abbreviations

MIDI	Musical Instrument Digital Interface
REMI	Revamped MIDI (representation)
ST	Score token (representation)
MEGA	Moving average equipped gated attention
JS	Jensen-Shannon
MOS	Mean opinion score

Acknowledgements
We would like to thank all the participants who took part in the subjective evaluation.

Authors' contributions
Not applicable.

Funding
Not applicable.

Availability of data and materials
The tools for proposed ST+ representation are publicly available at https://github.com/suzuqn/ScoreRearrangement to promote further research.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 30 June 2023 Accepted: 13 November 2023
Published online: 05 December 2023

References

1. E. Nakamura, K. Yoshii, Statistical piano reduction controlling performance difficulty. APSIPA Trans. Signal Inf. Process. **7**, 1–12 (2018)
2. M. Terao, R. Ishizuka, Y. Wu, K. Yoshii, in *Proceedings of the 47th IEEE International Conference on Acoustics, Speech and Signal Processing*, Difficulty-aware neural band-to-piano score arrangement based on note- and statistic-level criteria (2022), pp. 196–200

3. S.C. Chiu, M.S. Chen, in *Proceedings - 2012 IEEE International Symposium on Multimedia*, A study on difficulty level recognition of piano sheet music (2012), pp. 17–23
4. P. Ramoneda, N.C. Tamer, V. Eremenko, X. Serra, M. Miron, in *Proceedings of the 47th IEEE International Conference on Acoustics, Speech and Signal Processing*, Score difficulty analysis for piano performance education based on fingering (2022), pp. 201–205
5. T. Fukuda, Y. Ikemiya, K. Itoyama, K. Yoshii, in *Proceedings of the 12th International Conference in Sound and Music Computing*, A score-informed piano tutoring system with mistake detection and score simplification (2015), pp. 105–110
6. M. Gover, O. Zewi, in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, Music translation: generating piano arrangements in different playing levels (2022), pp. 36–43
7. M.A. Román, A. Pertusa, J. Calvo-Zaragoza, in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, A holistic approach to polyphonic music transcription with neural networks (2019), pp. 731–737
8. Y.S. Huang, Y.H. Yang, in *Proceedings of the 28th ACM International Conference on Multimedia*, Pop music transformer: beat-based modeling and generation of expressive pop piano compositions (2020), pp. 1180–1188
9. W.Y. Hsiao, J.Y. Liu, Y.C. Yeh, Y.H. Yang, in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, vol. 1, Compound word transformer: learning to compose full-Song music over dynamic directed hypergraphs (2021), pp. 178–186
10. D. von Rütte, L. Biggio, Y. Kilcher, T. Hofmann, in *Proceedings of the International Conference on Learning Representations*, FIGARO: generating symbolic music with fine-grained artistic control (2023)
11. J. Ens, P. Pasquier, in *Extended Abstracts for the Late-Breaking Demo Session of the 21st International Society for Music Information Retrieval Conference*, Exploring conditional multi-track music generation with the Transformer (2020)
12. H.W. Dong, K. Chen, S. Dubnov, J. McAuley, T. Berg-Kirkpatrick, in *Proceedings of the 48th IEEE International Conference on Acoustics, Speech and Signal Processing*, Multitrack Music Transformer: Learning Long-Term Dependencies in Music with Diverse Instruments (2023)
13. S. Oore, I. Simon, S. Dieleman, D. Eck, K. Simonyan, This time with feeling: learning expressive musical performance. Neural Comput. & Applic. **32**, 955–967 (2018)
14. L. Liu, V. Morfi, E. Benetos, in *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing*, Joint multi-pitch detection and score transcription for polyphonic piano music (2021), pp. 281–285
15. M. Suzuki, in *Proceedings of the 3rd ACM International Conference on Multimedia in Asia*, Score Transformer: generating musical score from note-level representation (2021), pp. 311–317
16. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Attention is all you need (2017), pp. 6000–6010
17. S.L. Wu, Y.H. Yang, MuseMorphose: Full-Song and Fine-Grained Music Style Transfer with Just One Transformer VAE. IEEE/ACM Trans. Audio Speech Lang. Process. **31**,1953-1967 (2023)
18. C. Hawthorne, I. Simon, R. Swavely, E. Manilow, J. Engel, in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, Sequence-to-sequence piano transcription with transformers (2021), pp. 246–253
19. A. Gu, K. Goel, C. Ré, in *Proceedings of the 9th International Conference on Learning Representations*, Efficiently modeling long sequences with structured state spaces (2021)
20. X. Ma, C. Zhou, X. Kong, J. He, L. Gui, G. Neubig, J. May, L. Zettlemoyer, in *Proceedings of the 10th International Conference on Learning Representations*, MEGA: moving average equipped gated attention (2022)
21. J.T.H. Smith, A. Warrington, S.W. Linderman, in *Proceedings of the 11th International Conference on Learning Representations*, Simplified state space layers for sequence modeling (2023)
22. Y. Tay, M. Dehghani, S. Abnar, Y. Shen, D. Bahri, P. Pham, J. Rao, L. Yang, S. Ruder, D. Metzler, in *Proceedings of the 9th International Conference on Learning Representations*, Long Range Arena: a benchmark for efficient transformers (2021)
23. M. ElNokrashy, A. Hendy, M. Maher, M. Afify, H.H. Awadalla, in *Proceedings of the 44th Annual Meeting and Symposium of the Antenna Measurement Techniques Association*, Language tokens: a frustratingly simple approach improves zero-shot performance of multilingual translation (2022)
24. M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, M. Auli, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, Fairseq: a fast, extensible toolkit for sequence modeling (2019), pp. 48–53
25. E. Cambouropoulos, in *Proceedings of the International Computer Music Conference*, The Local Boundary Detection Model (LBDM) and its application in the study of expressive timing (2001), pp. 17–22
26. C. Hernandez-Olivan, S.R. Llamas, J.R. Beltran, Symbolic music structure analysis with graph representations and changepoint detection methods. arXiv:2303.13881 (2023)
27. X. Guan, H. Zhao, Q. Li, Estimation of playable piano fingering by pitch-difference fingering match model. EURASIP J. Audio Speech Music Process. **2022**(7) (2022). https://asmp-eurasipjournals.springeropen.com/articles/10.1186/s13636-022-00237-8
28. M. Suzuki, in *Extended Abstracts for the Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference*, Piano fingering estimation and completion with transformers (2021)
29. P. Ramoneda, D. Jeong, E. Nakamura, X. Serra, M. Miron, in *Proceedings of the 30th ACM International Conference on Multimedia*, Automatic piano fingering from partially annotated scores using autoregressive neural networks (2022), pp. 6502–6510
30. N. Srivatsan, T. Berg-Kirkpatrick, in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, Checklist models for improved output fluency in piano fingering prediction (2022), pp. 525–531

Publisher's Note