TIME-SHIFTED TOKEN SCHEDULING FOR SYMBOLIC MUSIC GENERATION

Ting-Kang Wang *Chih-Pin Tan* *Yi-Hsuan Yang*

Graduate Institute of Communication Engineering, National Taiwan University, Taiwan

ABSTRACT

Symbolic music generation faces a fundamental trade-off between efficiency and quality. Fine-grained tokenizations achieve strong coherence but incur long sequences and high complexity, while compact tokenizations improve efficiency at the expense of intra-token dependencies. To address this, we adapt a **delay-based scheduling mechanism (DP)** that expands compound-like tokens across decoding steps, enabling autoregressive modeling of intra-token dependencies while preserving efficiency. Notably, DP is a lightweight strategy that introduces no additional parameters and can be seamlessly integrated into existing representations. Experiments on symbolic orchestral MIDI datasets show that our method improves all metrics over standard compound tokenizations and narrows the gap to fine-grained tokenizations.

***Index Terms*—** Symbolic music generation, music representation, compound tokens, delay pattern

1. INTRODUCTION

Symbolic music generation with Transformer-based language models has shown potential once MIDI sequences are tokenized into discrete events. A key challenge lies in designing effective tokenization schemes. REMI [1, 2, 3] decompose a note into multiple attributes, such as pitch, duration, and velocity, producing flattened sequences that align well with language modeling. However, this leads to long token sequences, which substantially increase memory cost and inference latency.

To address this, token grouping strategies have been introduced. Compound Word Transformer [4] demonstrated that grouping the attributes of a note into a single compound token can significantly shorten the sequence. In this framework, K attribute embeddings are first combined (by summation or concatenation) into a single vector, then fed into the Transformer for sequence modeling. At the output stage, K parallel linear heads predict each attribute, thereby allowing all attributes to be decoded in parallel.

Similar multi-attribute representations have also appeared in the audio domain: MusicGen [5] uses residual vector quantization (RVQ) to encode each audio frame into multiple coarse-to-fine layers, where each layer captures different levels of musical attributes.

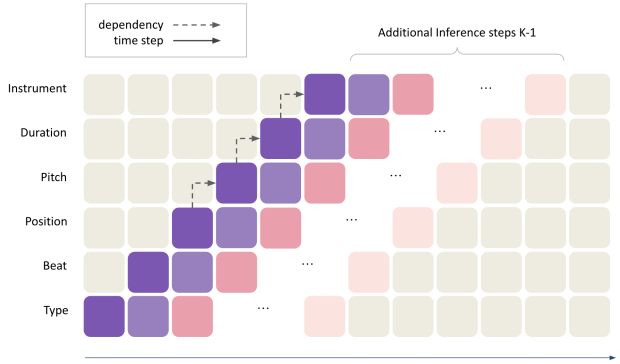

Fig. 1: Visualization of delay-based interleaving mechanism on compound tokenization

However, token grouping introduces a subtle trade-off. Since attributes are predicted in parallel from a single Transformer output, intra-token dependencies (e.g., pitch–duration correlation) may be ignored.

To mitigate this issue, the Nested Music Transformer (NMT) [6] extends the standard Transformer with a sub-decoder module. Instead of predicting all attributes by parallel linear heads, the sub-decoder autoregressively unfolds each attribute from this hidden vector, using the outputs of previously predicted attributes as additional context. To strengthen this process, NMT introduces cross-attention mechanisms in its sub-decoder. The intra-token decoder performs cross-attention between the main decoder's hidden state and the sequence of already generated attributes, ensuring that each new attribute prediction is conditioned on both the compound token context and its intra-token history. This allow NMT to model intra-token attributes correlations more effectively, achieving quality comparable to REMI while still benefiting from compact sequences. Yet, this improvement comes at a cost: the additional sub-decoders and cross-attention layers increase memory footprint and inference time, partly undermining the original motivation of token grouping.

This motivates us to ask: is there a lightweight approach that enhances intra-token dependency modeling without increasing model complexity? Inspired by MusicGen's interleaving schedule for RVQ codebooks [5], we adapt a **delay-based interleaving mechanism (denoted as DP)**

	Inference Speed (Notes Per Second)	Complexity (Big-O)
MMT [7]	63.53	$O(N^2)$
MMT-DP (Ours)	62.47	$O((N+(K-1))^2)$
NMT [6]	41.99	$O(E^2)+O(NK)$
REMI+ [8]	20.42	$O((NK)^2)$

Table 1: Comparison of inference speed and complexity. N denotes the number of note events in a sequence, and K is the number of sub-fields per token (e.g., type, beat, pitch, etc.).

for symbolic music representation. This design allows the model to autoregressively capture dependencies between attributes while still preserving a single causal stream. Importantly, DP is not a new tokenization scheme, but rather a lightweight scheduling strategy that can be plugged into existing compound-like tokenizations, introducing no additional parameters and requiring only minimal changes to the data loader.

In experiments, we evaluate the scheduling mechanism by applying it to Multitrack Music Transformer (MMT) [7], a general multitrack extension of compound tokenization. We refer to this variant as MMT-DP and evaluate it on orchestral continuation tasks. Results show that applying DP narrow the quality gap to REMI representations while preserving lightweight inference efficiency. To ensure reproducibility and encourage adoption, we will open-source the implementation and offer an demo page[1] with our generated samples.

2. RELATED WORK

Language models, particularly Transformer-based architectures [9], have become dominant in natural language processing tasks [10]. Several studies have adapted language modeling techniques to music. For example, [11] proposed a MIDI-event-like representation that creates a time-ordered sequence for each track and concatenates multiple tracks into one sequence. In [12], a permutation-invariant BERT-like language model was introduced for symbolic music understanding. In [13], the author proposed a permutation-invariant architecture with a novel multitrack representation, 3-D positional embeddings, and a byte pair encoding scheme for music tokenization. Building on these advances, symbolic music has been studied in various aspects. Some prior work focused on unconditioned generation, including generating piano music [14, 1], guitar tabs [15], lead sheets [16, 17] and multitrack music [11, 13, 7, 6] from scratch. Others studied have explored controllable music generation [8, 18], song conver generation [19, 20], music style transfer [21, 22], polyphonic score infilling [23] and general-purpose pretraining for symbolic music understanding [24, 12].

[1]Demo Page: `https://tklovln.github.io/dptoken-demo/`

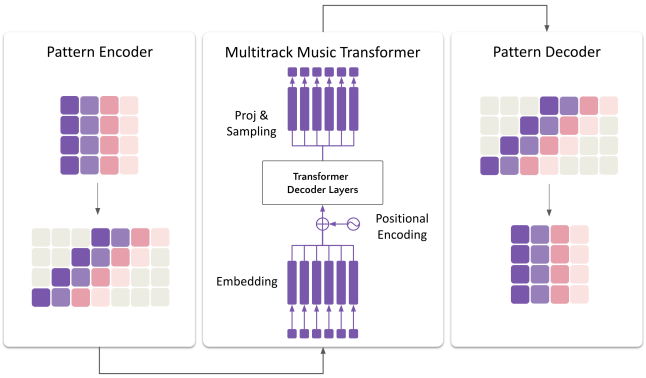

Fig. 2: Overview of the system architecture. A delay-pattern encoder decoder framework is integrated into a multitrack music transformer.

3. PROPOSED METHOD

3.1. Token Representation

Fig. 1 shows the music representation of our proposed delay scheduling, we adopt a compound representation with six discrete sub-fields per event:

$$\texttt{type}, \texttt{beat}, \texttt{position}, \texttt{pitch}, \texttt{duration}, \texttt{instrument}$$

where onset $= \texttt{beat}\cdot r + \texttt{pos}$ at resolution $r = 12$, following prior practice [4, 1]. In Figure 1, horizontal axis corresponds to the autoregressive inference time steps of the transformer decoder, while the vertical axis denotes the sub-fields within each compound token. Dashed lines indicate potential dependencies among musical attributes across different sub-fields, illustrating how structural relations may extend beyond the strict autoregressive order. It can also be observed that this modification to the compound token only increases the token sequence by a constant term, namely $K-1$.

3.2. Delay Scheduling Mechanism

Given event $e_i = \{e_i^{(1)}, \ldots, e_i^{(6)}\}$, we assign fixed delays $\{\Delta_d\}$ and decode fields over adjacent steps:

$$t = i + \Delta_d, \tag{1}$$

$$p(e_i^{(d)} \mid \text{ctx}_{<t}) = p\Big(e_i^{(d)} \mid \{e_i^{(d')} : \Delta_{d'} < \Delta_d\},\ \text{events} < i\Big). \tag{2}$$

We set $\Delta_{\texttt{type}}=0$, then $\Delta_{\texttt{beat}}=1$, $\Delta_{\texttt{pos}}=2$, $\Delta_{\texttt{pitch}}=3$, $\Delta_{\texttt{duration}}=4$, $\Delta_{\texttt{instrument}}=5$. The joint factorization for one event becomes:

$$p(e_i) = \prod_{d=1}^{6} p\Big(e_i^{(d)} \,\Big|\, e_i^{(1:d-1)},\ e_{<i}\Big). \tag{3}$$

In addition, we explored different Δ_d settings and permutations of sub-field ordering. Ultimately, we found that using a uniform step delay with the current ordering provides

the best empirical performance while preserving the natural causal structure of the representation.

3.3. Architecture and Objective

Following [7], we use a decoder-only Transformer with intra-token embeddings summed into a compound embedding, plus absolute positional embeddings. A lightweight output heads map the hidden state to each sub-field's vocabulary. We train with teacher forcing and the sum of cross-entropies over sub-fields and positions; inference uses top-k sampling per field with monotonicity constraints on `type` and `beat`.

4. EXPERIMENTS

4.1. Datasets

We conduct our experiments on **SymphonyNet** [13], a dataset comprising 46,359 orchestral MIDI scores with an average duration of 256 seconds per piece, amounting to a total of 3,284 hours. The dataset features multitrack arrangements that span a wide range of time signatures and encompass both contemporary and classical orchestration styles. For data augmentation we apply random pitch transpositions $s \sim \mathcal{U}(-5, 6)$ (semitones) to improve robustness. Each dataset is split into training, validation and test sets with an 80% / 10% / 10% ratio. For preprocessing and visualization, we use the `muspy` [25] and `pypianoroll` [26] libraries.

4.2. Models and Training

We train our models under the same configuration as MMT: 8 layers, 8 attention heads, $d_{\text{model}} = 512$, feed-forward dimension of 2048, dropout rate 0.1, and the Adam optimizer with an initial learning rate of 3×10^{-4}, linear warmup followed by inverse square root decay. Training is conducted with a maximum sequence length of 1024, batch size 16, for 200k steps. For the NMT baseline, we instead adopt the settings reported in its original paper. We compare (a) MMT,(b) MMT with delay-pattern scheduling (denoted as MMT-DP), (c) REMI+[8], a multitrack extension of REMI, and (d) NMT.

5. RESULTS

5.1. Subjective Listening Test

To assess perceptual quality, we conducted a blind listening test with 26 participants (each playing at least one instrument). Among the participants, 16 had advanced musical training with over four years of experience, 5 had between one and three years of training, and 5 had less than one year of experience. We adopted a 2-bar continuation setup: for each sample, the first two measures of a MIDI prompt were provided to the model, which then generated the continuation until reaching either 1024 time steps or an end-of-sequence token. All MIDI outputs were rendered to audio using the `fluidsynth` library.

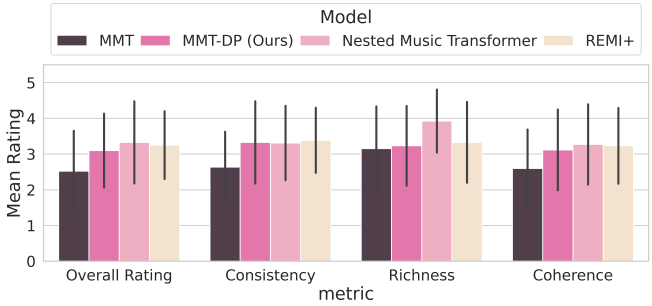

Fig. 3: Orchestra generation MOS with 95% CI

Model	Pitch Class Entropy	Scale Consistency	Groove Consistency
Ground truth	2.70 (±0.39)	0.92 (±0.08)	0.90 (±0.07)
MMT [7]	2.42 (±0.46)	0.96 (±0.05)	0.90 (±0.07)
NMT [6]	2.74 (±0.43)	0.92 (±0.07)	0.99 (±0.00)
REMI+ [8]	2.64 (±0.46)	0.92 (±0.07)	0.88 (±0.08)
MMT-DP (Ours)	2.53 (±0.46)	0.95 (±0.06)	0.93 (±0.05)

Table 2: Objective evaluation results on orchestra generation.

Each participant was presented with 2 randomly selected samples. For each sample, participants first heard the 2-bar prompt, followed by continuations from 4 models presented in randomized order. Ratings were collected on a 1–5 scale, where higher values indicate better quality, across four dimensions: (i) **Coherence**, reflecting the smoothness of transitions and phrasing; (ii) **Richness**, capturing the variety and interest of texture and harmony; (iii) **Consistency**, assessing the absence of compositional errors and overall unity; and (iv) **Overall Rating**, representing the general impression of the continuation.

Fig. 3 shows the mean opinion scores (MOS) of all models. We observe that applying delay scheduling leads to steady improvements over the MMT baseline across all metrics, especially in consistency, coherence, and overall listening quality. The results reach a level comparable to NMT and REMI+, confirming the strength of the proposed method. These findings highlight the benefit of applying delay scheduling to compound tokenization, which helps narrow the quality gap between compact compound tokens and fine-grained tokenization.

5.2. Objective Evaluation

In addition to the subjective listening test, we follow [7, 6] and measure pitch class entropy, scale consistency, and groove consistency to evaluate the performance of the proposed model. For these metrics, values closer to the ground truth are considered better. As shown in Table 2, our generated samples achieve scores that are consistently close to the ground truth across all three dimensions, suggesting that the model is able to produce musically plausible outputs.

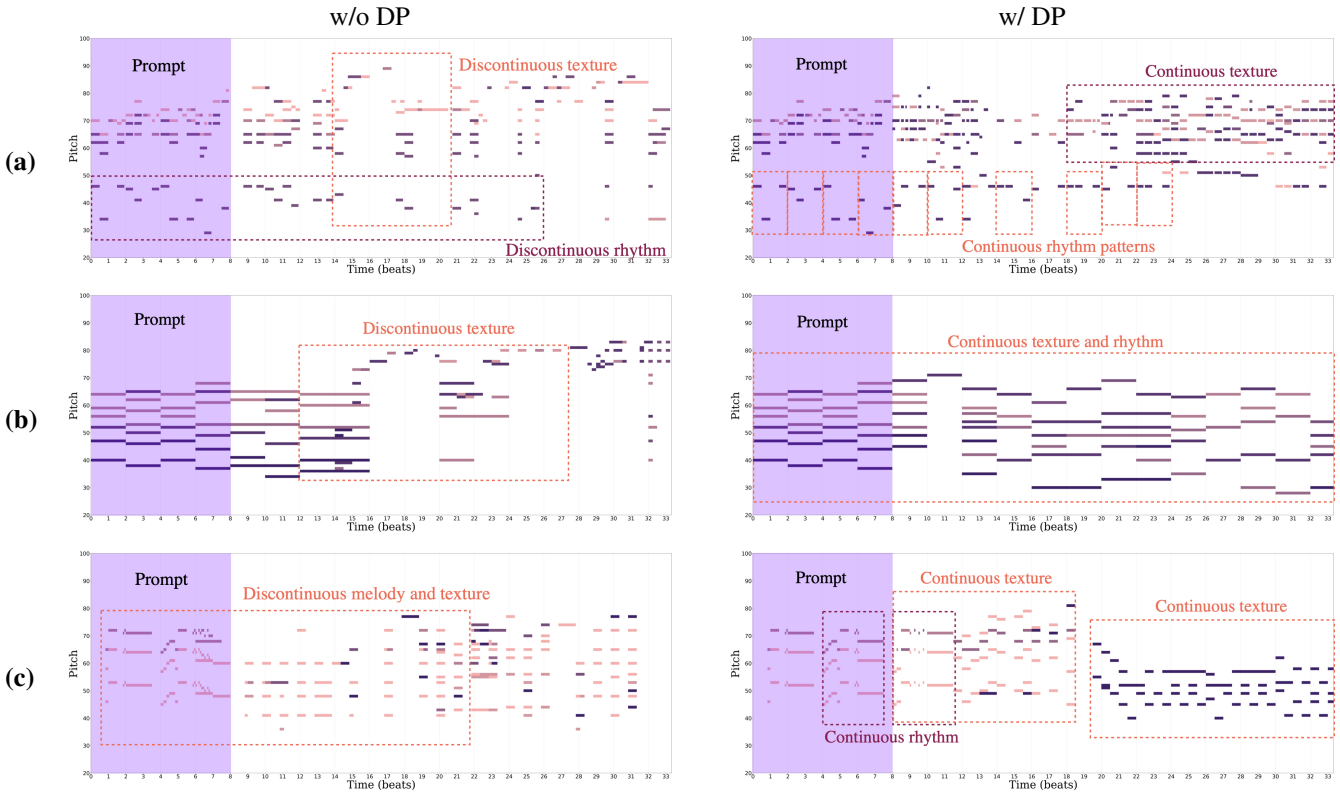

Fig. 4: Comparison of 3 test prompts, each row showing outputs from the same model w/o (left) and w/ (right) delay scheduling. The shaded region indicating the prompt, and colors indicate different instrument tracks.

5.3. Case Study on Multitrack Generation

To demonstrate the effectiveness of our approach, we present a case study on a 2-bar continuation task, using the same configuration described in Section 5.1. Figure 4 illustrates examples of 2-bar continuation, comparing outputs generated with and without delay-pattern scheduling (DP).

In case (a), sample without DP quickly diverges from the prompt, exhibiting discontinuous texture and fragmented rhythm patterns. With DP, however, both rhythmic and textural structures remain consistent. In case (b), the baseline maintains continuous texture and rhythm at first, but later produces abrupt gaps and unstable patterns. The DP variation model mitigates this by producing rhythmically stable sequences. In case (c), we observe that the melody line is scattered across the continuation and fails to align with the prompt's texture and rhythmic pattern, leading to incoherent pieces. In contrast, samples with DP preserves rhythm patterns at the beginning and maintain consistent textures by multiple measures, producing a more coherent and musically plausible continuation.

Beyond improving stability and coherence, the continuations also demonstrate creativity through novel chord progressions and melodic development, showing that the model is not merely imitative but musically expressive.

5.4. Inference-time Complexity

In addition to quality, a key advantage of our approach is its lightweight nature at inference time. We evaluated efficiency by generating sequences up to 1024 steps or until an EOS token on a single RTX 3090 GPU with default decoding settings in Section 4.2. For each piece, we computed notes-per-second (NPS) and averaged across the test set (see Section 4.1). As shown in Table 1, MMT reaches 63.53 NPS, while MMT-DP achieves 62.47 (only 1.7% slower). In contrast, NMT runs at 41.99 NPS and REMI+ at 20.42. These results confirm that delay scheduling adds virtually no overhead, with MMT-DP over 50% faster than NMT and nearly 3$\times$ faster than REMI+, making it suitable for real-time applications.

6. CONCLUSION

We demonstrate that the delay-based scheduling mechanism, previously explored in the audio domain, can also be applied symbolic music representation. It strengthens intra-token dependency modeling without adding complexity or inference overhead, and achieves promising results on orchestra generation. For future work, we plan to further explore how different delay orders influence music generation models, as well as validate the approach across broader musical styles and specific instrument corpora such as piano.

7. REFERENCES

[1] Yu-Siang Huang and Yi-Hsuan Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM International Conference on Multimedia (MM 2020)*. 2020, pp. 1180–1188, ACM.

[2] Jingyue Huang, Ke Chen, and Yi-Hsuan Yang, "Emotion-driven piano music generation via two-stage disentanglement and functional representation," *arXiv preprint arXiv:2407.20955*, 2024.

[3] Hsiao-Tzu Hung, Joann Ching, Seungheon Doh, Nabin Kim, Juhan Nam, and Yi-Hsuan Yang, "Emopia: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation," *arXiv preprint arXiv:2108.01374*, 2021.

[4] Wen-Yi Hsiao, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang, "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021, vol. 35, pp. 178–186, AAAI Press.

[5] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez, "Simple and controllable music generation," *Advances in Neural Information Processing Systems*, vol. 36, pp. 47704–47720, 2023.

[6] Jiwoo Ryu, Hao-Wen Dong, Jongmin Jung, and Dasaem Jeong, "Nested music transformer: Sequentially decoding compound tokens in symbolic music and audio generation," *arXiv preprint arXiv:2408.01180*, 2024.

[7] Hao-Wen Dong, Ke Chen, Shlomo Dubnov, Julian J. McAuley, and Taylor Berg-Kirkpatrick, "Multitrack music transformer," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023)*. 2023, pp. 1–5, IEEE.

[8] Dimitri von Rütte, Luca Biggio, Yannic Kilcher, and Thomas Hofmann, "Figaro: Generating symbolic music with fine-grained artistic control," *arXiv preprint arXiv:2201.10936*, 2022.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[11] Jeff Ens and Philippe Pasquier, "Mmm: Exploring conditional multi-track music generation with the transformer," *arXiv preprint arXiv:2008.06048*, 2020.

[12] Mingliang Zeng, Xu Tan, Rui Wang, Zeqian Ju, Tao Qin, and Tie-Yan Liu, "Musicbert: Symbolic music understanding with large-scale pre-training," *arXiv preprint arXiv:2106.05630*, 2021.

[13] Jiafeng Liu, Yuanliang Dong, Zehua Cheng, Xinran Zhang, Xiaobing Li, Feng Yu, and Maosong Sun, "Symphony generation with permutation invariant language model," *arXiv preprint arXiv:2205.05448*, 2022.

[14] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck, "Music transformer: Generating music with long-term structure," in *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*. 2019, OpenReview.net.

[15] Yu-Hua Chen, Yu-Hsiang Huang, Wen-Yi Hsiao, and Yi-Hsuan Yang, "Automatic composition of guitar tabs by transformers and groove modeling," *arXiv preprint arXiv:2008.01431*, 2020.

[16] Shih-Lun Wu and Yi-Hsuan Yang, "The jazz transformer on the front line: Exploring the shortcomings of ai-composed music through quantitative measures," *arXiv preprint arXiv:2008.01307*, 2020.

[17] Shih-Lun Wu and Yi-Hsuan Yang, "Compose & embellish: Well-structured piano performance generation via a two-stage approach," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[18] Yi-Jen Shih, Shih-Lun Wu, Frank Zalkow, Meinard Müller, and Yi-Hsuan Yang, "Theme transformer: Symbolic music generation with theme-conditioned transformer," *IEEE Transactions on Multimedia*, vol. 25, pp. 3495–3508, 2022.

[19] Chih-Pin Tan, Shuen-Huei Guan, and Yi-Hsuan Yang, "Picogen: generate piano covers with a two-stage approach," in *Proceedings of the 2024 International Conference on Multimedia Retrieval*, 2024, pp. 1180–1184.

[20] Chih-Pin Tan, Hsin Ai, Yi-Hsin Chang, Shuen-Huei Guan, and Yi-Hsuan Yang, "Picogen2: Piano cover generation with transfer learning approach and weakly aligned data," *arXiv preprint arXiv:2408.01551*, 2024.

[21] Shih-Lun Wu and Yi-Hsuan Yang, "Musemorphose: Full-song and fine-grained piano music style transfer with one transformer vae," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1953–1967, 2023.

[22] Dinh-Viet-Toan Le and Yi-Hsuan Yang, "Meteor: Melody-aware texture-controllable symbolic orchestral music generation," *arXiv preprint arXiv:2409.11753*, 2024.

[23] Chin-Jui Chang, Chun-Yi Lee, and Yi-Hsuan Yang, "Variable-length music score infilling via xlnet and musically specialized positional encoding," *arXiv preprint arXiv:2108.05064*, 2021.

[24] Ziyu Wang and Gus Xia, "Musebert: Pre-training music representation for music understanding and controllable generation.," in *ISMIR*, 2021, pp. 722–729.

[25] Hao-Wen Dong, Ke Chen, Julian McAuley, and Taylor Berg-Kirkpatrick, "Muspy: A toolkit for symbolic music generation," *arXiv preprint arXiv:2008.01951*, 2020.

[26] Hao-Wen Dong, Wen-Yi Hsiao, and Yi-Hsuan Yang, "Pypianoroll: Open source python package for handling multitrack pianoroll," *Proc. ISMIR. Late-breaking paper*, 2018.