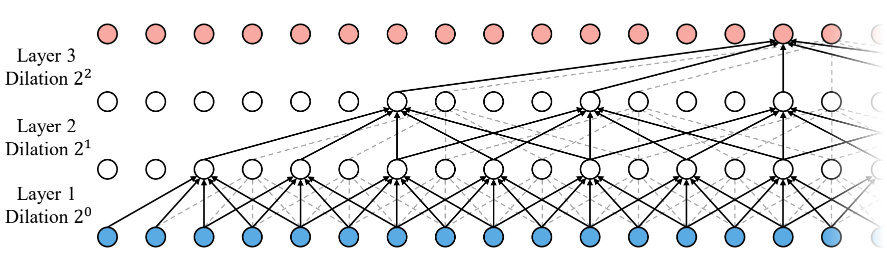
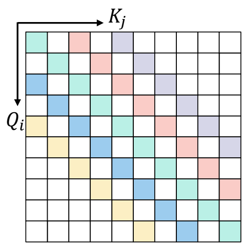


(a) Layer-wise view of dilated self-attention (Adapted from TCN structures [9,27])



(b) Attention matrix view at layer 2

Figure 1: Illustration of dilated self-attention (with a non-causal short window of size 5) over a three-layer Transformer. Part (a) shows the hierarchical connectivity across layers, which shares the same pattern as TCN in [9]. Part (b) shows the attention matrix at layer 2, with colours indicating relative position. The white colour indicates unattainable positions.

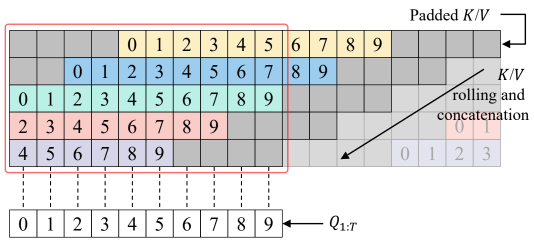


Figure 2: Illustration of efficient DSA implementation, with dilation rate $r = 2$ and window size $l_{\text{win}} = 5$.

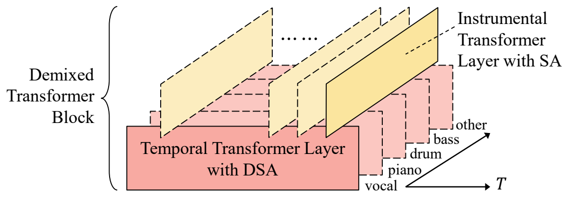


Figure 3: Demixed Transformer block. Two Transformer layers are stacked "orthogonally", each handling time-wise dilated self-attention and instrument-wise self-attention.

Then, the Q-K attention e_{ik} is normalized via softmax:

$$p_{ik} = \frac{\exp(e_{ik})}{\sum_{k=-m}^{n} \exp(e_{ik})} \tag{4}$$

The output of DSA is a sequence $z_{1:T}$, where z_i is the weighted average of V under the same dilated window:

$$z_i = \sum_{k=-m}^{n} p_{ik}(V_{i+rk}) \tag{5}$$

We apply DSA with exponentially increasing dilation rates and relative positional embedding (RPE) [32] to a stack of Transformer layers. Each layer consists of two sub-layers: DSA, and a position-wise feed-forward layer. We place residual connections across each sub-layer and perform layer normalization [33] before each sub-layer.

3.1.3 Memory Efficient Implementation of DSA

A straightforward implementation of DSA is to mask SA. Concretely, a square mask takes the same form as in Figure 1b, where all uncolored positions are filled with `-inf`, rendering a rather sparse attention matrix. This way, however, still requires quadratic complexity, because e_{ij} is explicitly computed for all masked positions.

Our implementation takes a "rolling" strategy to eliminate redundant computation. As in Figure 2, l_{win} copies of $K_{1:T}$ sequences are padded and rolled along the time axis, and then concatenated on a new axis. In this way, each Q_i sees K_j directly and only at $j = i + rk$ for $-m \le k \le n$, which is exactly the coverage of the dilated window.

Formally, given $K_{1:T} \in \mathbb{R}^{T \times d_{\text{f}}}$, dilation rate r, and window size l_{win}, the rolling strategy takes three steps:

1. Pad $K_{1:T}$ with $\lfloor \frac{l_{\text{win}}}{2} \rfloor \times r$ steps on both sides;
2. Make l_{win} copies of padded K. Starting from 0, each copy is cyclically rolled r more steps;
3. Concatenate each copy along a new axis and retrieve the first T steps. The output has shape $T \times l_{\text{win}} \times d_{\text{f}}$.

The same procedure applies to $V_{1:T}$ as well. In this way, computing DSA is essentially as simple as Equation (1). Here, instead of $Q, K, V \in \mathbb{R}^{T \times d_{\text{f}}}$, we have $Q \in \mathbb{R}^{T \times 1 \times d_{\text{f}}}$ and $K, V \in \mathbb{R}^{T \times l_{\text{win}} \times d_{\text{f}}}$, with T treated as a batch dimension. The computation complexity is $\mathcal{O}(T \times l_{\text{win}})$, while l_{win} is fixed and small enough to be left uncounted.

3.2 Demixed Beat Tracking

3.2.1 Demixed Transformer Block

We use Spleeter 5-stems model [22] to demix an input piece into spectrograms with $|C|$ instrument channels, where $C = \{\texttt{vocal}, \texttt{piano}, \texttt{drum}, \texttt{bass}, \texttt{other}\}$. As shown in Figure 3, we stack two Transformer layers to perform time-wise and instrument-wise attention in turn. Let the input at layer l be $x^l_{1:T,1:|C|} \in \mathbb{R}^{T \times |C| \times d_{\text{f}}}$, a *temporal* Transformer layer (TTL) first takes $x^l_{1:T,c}$ for $1 \le c \le |C|$:

$$x^{l+1}_{1:T,c} = \text{TTL}(x^l_{1:T,c}) \tag{6}$$

Then, an *instrumental* Transformer layer (ITL), on the *orthogonal* direction, takes $x^{l+1}_{t,1:|C|}$ for $1 \le t \le T$:

$$x^{l+2}_{t,1:|C|} = \text{ITL}(x^{l+1}_{t,1:|C|}) \tag{7}$$

A TTL followed by ITL forms a *demixed Transformer block*. TTL consists of DSA as described in Section 3.1.2.

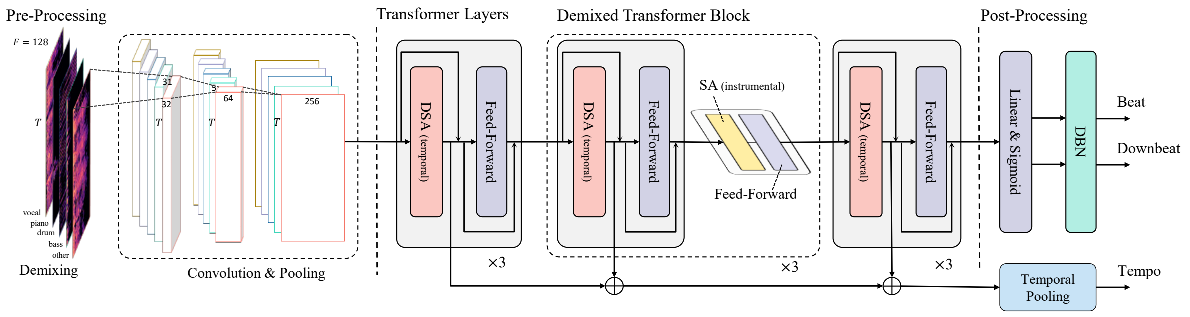


Figure 4: Beat Transformer architecture. For conciseness, layer normalization and dropout layers are not shown.

For ITL, we use vanilla SA because there is only 5 instrument channels. As instruments are not sequentially ordered, we do not add any positional encoding to ITL.

Through demixed Transformer blocks, our model can capture the rhythmic evolution of each instrument, as well as the harmonic coordination among all instruments.

3.2.2 *Partial Demix Augmentation*

Spleeter may produce empty channels when a certain instrument does not present. To avoid potential effects of such a situations, we develop a partial demix strategy for data augmentation. Partial demix creates new stems by summing up existing instrument channels of a default 5-stem demixed input sample. For example, an augmented data sample may have three channels corresponding to $C' = \{\texttt{vocal\&piano}, \texttt{drum}, \texttt{bass\&other}\}$.

In our case, we randomly sum up 2, 3, or 4 instrument channels of a 5-stem input with a probability 30%, 10%, and 10% during training. In this way, our model is encouraged to pay attention to *instrument-agnostic* musical contents and thus is less affected by empty channels where no valid music content is present. As our augmentation strategy also adds to the demix diversity and the data quantity, we believe it brings general benefits to training as well.

3.3 Markov Chain Interpretation

In Equation (4), we formulate the attention matrix of DSA as $P = [p_{ij}]_{1 \le i,j \le T}$, where $p_{ij} \ge 0$ if and only if $j = i + rk$ for $-m \le k \le n$. Here, r is the dilation rate, and m, n are components of the attention window. Moreover, P satisfies $\sum_{j=1}^{T} p_{ij} = 1$ for all i. Therefore, P can be regarded as the transition matrix of a finite-state Markov chain, where each state is a position of the input sequence.

For a stack of temporal Transformer layers (TTL), where DSA is employed, layer l essentially learns a unique one-step transition P^l, by which our model can attend to local neighbours covered by the attention window. Through L layers, our model makes an L-step transition, during which it attends to global positions hierarchically. The overall L-step transition matrix $P^{(L)}$ satisfies:

$$P^{(L)} = \prod_{l=1}^{L} P^l \tag{8}$$

Note that P^l itself is a rather sparse matrix (due to *short* attention window), while $P^{(L)}$ is densely connected. Its components $[p_{ij}^{(L)}]_{1 \le i,j \le T}$ represent the hierarchical attention weights across the whole L layers, which can tell us much richer attention patterns (more in Section 4.4).

3.4 Complete Architecture

A complete view of Beat Transformer is presented in Figure 4. The inputs are log-scaled spectrograms demixed by Spleeter, with $F = 128$ mel-bins and $|C| = 5$ instrument channels. Subject to Spleeter, our frame rate is 43.07 fps and the frequency range is up to 11 kHz. We then use three 2D convolutional layers, shared by each demixed channel, as a front-end feature extractor. The convolutional design is the same as in [13] except that we employ more filters to reach feature dimension $d_{\text{model}} = 256$.

Beat Transformer comprises 9 temporal Transformer layers (TTL) with DSA. Each TTL has 8 attention heads with window size $l_{\text{win}} = 5$, four of which have skewed window ranges, where $m = 0, 1, 3, 4$, respectively, and $n = 4 - m$. The dilation rate grows exponentially from 2^0 to 2^8, stretching to a receptive field of 47.51 seconds. Among the 9 TTLs, the middle three are expanded to demixed Transformer blocks by interleaving ITLs. We found it sufficient to perform instrumental attention only in the middle layers, which has a proper scale of 1-5 seconds. Each Transformer layer has 8 heads ($d_{\text{f}} = 32$) followed by a feed-forward layer with hidden dimension $d_{\text{ff}} = 1024$.

We sum up the instrument channels of the output of the last Transformer layer and obtain a frame-wise beat representation of shape $T \times d_{\text{model}}$. Following multi-task learning practice [11–14], we use a linear layer to map the representation to beat and downbeat activations respectively, and add a regularization branch predicting global tempo via "skip connections" [12]. We apply DBN in Madmom package [34] as the post-processor to pick up the beat and downbeat sequence from raw activations. For DBN parameters, we set `observation_lambda` $= 6$, `transition_lambda` $= 100$, and `threshold` $= 0.2$.

4. EXPERIMENTS

4.1 Datasets

We utilize a total of 7 datasets for model training and evaluation: *Ballroom* [35, 36], *Hainsworth* [37], *RWC Popular* [38], *Harmonix* [39], *Carnetic* [40], *SMC* [41],

		Beat Accuracy			Downbeat Accuracy		
Dataset	**Model**	**F-Measure**	**CMLt**	**AMLt**	**F-Measure**	**CMLt**	**AMLt**
Ballroom	TCN+Demix	0.960	0.942	0.960	0.925	0.924	0.956
	Ours w/o Demix	0.968	0.946	0.965	0.930	0.925	0.963
	Ours w/o Aug.	0.967	0.949	**0.967**	0.928	0.931	0.958
	Ours	**0.968**	**0.954**	0.966	**0.941**	**0.944**	**0.969**
	Böck *et al.* [13]	0.962	0.947	0.961	0.916	0.913	0.960
	Hung *et al.* [15]	0.962	0.939	0.967	0.937	0.927	0.968
Hainsworth	TCN+Demix	0.887	0.827	0.918	0.739	0.708	0.861
	Ours w/o Demix	0.902	0.844	0.934	0.721	0.688	0.843
	Ours w/o Aug.	0.892	0.831	0.908	0.742	0.703	0.837
	Ours	0.902	0.842	0.918	**0.748**	0.712	0.841
	Böck *et al.* [13]	**0.904**	0.851	**0.937**	0.722	0.696	**0.872**
	Hung *et al.* [15]	0.877	**0.862**	0.915	0.748	**0.738**	0.870
Harmonix	TCN+Demix	0.954	0.903	0.956	0.901	0.866	0.923
	Ours w/o Demix	0.954	0.902	0.958	0.887	0.846	0.916
	Ours w/o Aug.	0.952	0.901	0.950	0.897	0.863	0.919
	Ours	**0.954**	0.905	0.957	0.898	0.863	0.919
	Böck *et al.* [13]*	0.933	0.841	0.938	0.804	0.747	0.873
	Hung *et al.* [15]	0.953	**0.939**	**0.959**	**0.908**	**0.872**	**0.928**
SMC	TCN+Demix	0.596	0.455	0.625			
	Ours w/o Demix	0.589	0.448	0.621			
	Ours w/o Aug.	0.595	0.450	0.626			
	Ours	0.596	0.456	0.635			
	Böck *et al.* [13]	0.552	0.465	0.643			
	Hung *et al.* [15]	**0.605**	**0.514**	**0.663**			
GTZAN	TCN+Demix	0.873	0.780	0.907	0.700	0.646	0.842
	Ours w/o Demix	0.876	0.787	0.914	0.686	0.633	0.834
	Ours w/o Aug.	0.881	0.797	0.921	0.703	0.653	0.845
	Ours	0.885	0.800	0.922	0.714	0.665	0.844
	Böck *et al.* [13]	0.885	**0.813**	**0.931**	0.672	0.640	0.832
	Hung *et al.* [15]	**0.887**	0.812	0.920	**0.756**	**0.715**	**0.881**

Table 1: Testing results of beat and downbeat tracking under 8-fold cross-validation. GTZAN is unseen from training and held out for test only. Böck *et. al.* [13] on Harmonix is reproduced by [15], as indicated by the * symbol. We use underscore to denote best results comparing with our ablation models and use **boldface** to compare with state-of-the-art models.

and *GTZAN* [42, 43]. We acquire *Harmonix* in mel-spectrogram and invert each piece to audio using Griffin-Lim Algorithm [44, 45] with Librosa package [46]. Following convention, we leave *GTZAN* for testing only and use the other datasets in 8-fold cross validation [11–13].

4.2 Training

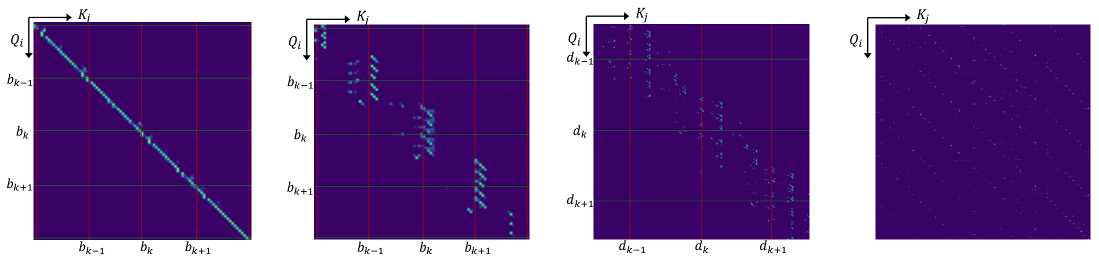
Our model is supervised in a multi-task learning fashion, where beat, downbeat, and tempo are predicted jointly [13]. Beat and downbeat annotations are each represented as a 1D binary sequence that indicates beat (1) and non-beat (0) states at each input frame. Following [13], we widen beat and downbeat states to ± 2 neighbours of annotated frames with weights 0.5 and 0.25. Following [12], we derive tempo target from beat annotation for the tempo prediction branch. We found the use of tempo branch generally beneficial to beat tracking, as it may serve as a regularization term that helps reaching better convergence.

For training, we combine the binary cross entropy loss over beat, downbeat, and tempo by weighing them equally. We use a batch size of 1 to train on whole sequences with different lengths. For excessively long songs, we split them into 3-minute (8k-frame) clips. We apply RAdam [47] plus Lookahead [48] optimizer with an initial learning rate of $1\mathrm{e}{-3}$, which is reduced by a factor of 5 whenever the validation loss gets stuck for 2 epochs before being capped at a minimum value of $1\mathrm{e}{-7}$. We use dropout [49] with rate 0.5 for the tempo branch and 0.1 for other parts of the network. We apply *partial demix augmentation* described in Section 3.2.2 as the only means of data augmentation. Our model has 9.29M trainable parameters and is trained with an RTX-A5000-24GB GPU. Each training fold generally takes 20 epochs (in 11 hours) to fully converge.

4.3 Evaluation

4.3.1 Baseline Methods

We first compare three ablation models to validate our module design. The first ablation model is *Ours* trained without partial demix augmentation (***Ours w/o Aug.***). The second model removes all ITL layers and tracks beat with a single channel of non-demixed mixture (***Ours w/o Demix***). The last model replaces each TTL layer with a TCN layer [13] of the same dilation rate (***TCN+Demix***). We imple-



(a) $P^{(1)}$ lower scale (b) $P^{(3)}$ beat scale (c) $P^{(5)}$ downbeat scale (d) $P^{(9)}$ higher scale

Figure 5: Visualization of temporal attention matrix based on the product rule in Equation (8) for L-step transition on a Markov chain. $L = 1, 3, 5$, and 9 from part (a) to (d), which model different hierarchies of the metrical structure.

ment TCN layers following [23] while setting the input and output shape to be the same as our model. The resulting model yields a comparable amount of 10.25M parameters and is trained without augmentation.

In addition, we compare our model with two recent works that have achieved state-of-the-art performance. Specifically, Böck *et al.* [13] is based on TCN architectures and Hung *et al.* [15] is based on SpecTNT.

4.3.2 Results and Discussion

In Table 1, we first observe *Ours w/o Aug.* yields generally better performance than *TCN+Demix*, especially on the unseen *GTZAN* dataset. As both models share a comparable amount of parameters, this result demonstrates the capability of Transformer (DSA) versus TCN (dilated convolution), which also corroborates with previous findings on Transformer's comparability to convolution on general tasks [50–52]. Considering that Transformer is notoriously data-inefficient to train, it is remarkable that our model is well-trained with limited data without augmentation. We owe this merit to DSA, which not only prevents redundant computation but also makes musical sense in terms of the hierarchical structure of music metrical modelling.

Comparing *Ours w/o Demix* to *Ours w/o Aug.*, while both models are highly competitive in beat tracking, the latter demonstrates more superiority in downbeat tracking. Downbeat tracking is generally more difficult than beat tracking because it is involved with deeper musical knowledge, such as chord and bass progression, behind the apparent spectrogram energy. In our model, the instrumental attention captures the instrumental coordination as hints to the harmonic cues that are orthogonal to the temporal axis, and thus acquires better metrical modelling.

Comparing *Ours w/o Aug.* to *Ours*, we observe a consistent improvement across datasets brought by partial demix augmentation, which indicates the general usefulness of this augmentation strategy to model training.

Compared to state-of-the-art models, our improvement in downbeat accuracy is more significant than that in beat accuracy. On the test-only *GTZAN* dataset, we obtain 4% point gain in *F-measure* over Böck *et al.* [13] in downbeat tracking. Compared to Hung *et al.* [15], which is also based on Transformer, our model can be more flexibly trained (owing to the efficient DSA mechanism) on a 24GB GPU in contrast to four 32GB GPUs reported in [15].

4.4 Attention Matrix Visualization

We visualize the attention matrix that our model learns by interpreting it as a *multi-step Markov transition matrix* as defined in Equation (8). Specifically, the L-step matrix is the product of L one-step matrices through L layers. Here we only consider TTLs with dilated self-attention, as ITLs work on an orthogonal axis. Figure 5 shows the attention matrix $P^{(L)}$ for $L = 1, 3, 5$, and 9 of the `drum` channel, inferred from the piece `hiphop.00090` chosen from *GTZAN*.

Figure 5a shows a one-step transition. Each position Q_i can only attend to its neighbours covered by the attention window. Still, we observe that beat positions (denoted by b_k) are likely to get more attention. In Figure 5b where we step to the beat scale, most attention spots are aligned with beats. Moreover, we observe that the attention at b_k is typically prolonged after Q_i leaves b_k, and is formed before Q_i reaches b_k for every k. This means that our model learns to transition its attention from the *offbeat* phase following the last beat to the *upbeat phase* preceding the next beat. Figure 5c further stretches the view to the downbeat scale, and we can see similar patterns aligned with downbeat positions (denoted by d_k). Finally, in Figure 5d, the attention reaches further positions and displays a structural pattern.

The above visualization demonstrates the inner logic that our model exploits for beat and downbeat tracking. We see that our model gathers information from both local and global scales with an organized hierarchy.

5. CONCLUSION

In conclusion, we contribute a novel Transformer architecture for audio beat and downbeat tracking. The main novelty lies first in our design of dilated self-attention, which brings down the computation complexity of Transformer from quadratic to linear level. In addition, we successfully enhance beat and downbeat tracking by utilizing off-the-shelf progress in music demixing. Our model not only captures deeper harmonic cues for better metrical inference but also discerns beat and downbeat in a visualizable hierarchical manner. Our model is efficient, interpretable, and potentially generalizable with highly competitive sequential modelling power. We hope our model encourages future MIR research toward universal music understanding.

6. REFERENCES

[1] A. Holzapfel and E. Benetos, "The sousta corpus: Beat-informed automatic transcription of traditional dance tunes," in *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR*, 2016, pp. 531–537.

[2] G. Shibata, R. Nishikimi, and K. Yoshii, "Music structure analysis based on an LSTM-HSMM hybrid model," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020, pp. 23–29.

[3] R. Nishikimi, E. Nakamura, M. Goto, K. Itoyama, and K. Yoshii, "Bayesian singing transcription based on a hierarchical generative model of keys, musical notes, and F0 trajectories," *IEEE ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1678–1691, 2020.

[4] R. Ishizuka, R. Nishikimi, and K. Yoshii, "Global structure-aware drum transcription based on self-attention mechanisms," *Signals*, vol. 2, no. 3, pp. 508–526, 2021.

[5] C. Donahue and P. Liang, "Sheet sage: Lead sheets from music audio," 2021, Late Breaking Demo in the 22nd International Society for Music Information Retrieval Conference, ISMIR. [Online]. Available: https://archives.ismir.net/ismir2021/latebreaking/000049.pdf

[6] T. Bi, P. Fankhauser, D. Bellicoso, and M. Hutter, "Real-time dance generation to music for a legged robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. IEEE, 2018, pp. 1038–1044.

[7] K. Yamamoto, "Human-in-the-loop adaptation for interactive musical beat tracking," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 794–801.

[8] Z. Cai, R. J. Ellis, Z. Duan, H. Lu, and Y. Wang, "Basic evaluation of auditory temporal stability (beats): A novel rationale and implementation," in *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR*, 2013, pp. 541–546.

[9] E. P. MatthewDavies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *27th European Signal Processing Conference, EUSIPCO*. IEEE, 2019, pp. 1–5.

[10] F. Krebs, S. Böck, and G. Widmer, "An efficient state-space model for joint tempo and meter tracking," in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR*, 2015, pp. 72–78.

[11] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR*, 2016, pp. 255–261.

[12] S. Böck, M. E. P. Davies, and P. Knees, "Multi-task learning of tempo and beat: Learning one to improve the other," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR*, 2019, pp. 486–493.

[13] S. Böck and M. E. P. Davies, "Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020, pp. 574–582.

[14] T. Oyama, R. Ishizuka, and K. Yoshii, "Phase-aware joint beat and downbeat estimation based on periodicity of metrical structure," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 493–499.

[15] Y. Hung, J. Wang, X. Song, W. T. Lu, and M. Won, "Modeling beats and downbeats with a time-frequency transformer," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*. IEEE, 2022, pp. 401–405.

[16] W.-T. Lu, J.-C. Wang, M. Won, K. Choi, and X. Song, "SpecTNT: A time-frequency transformer for music audio," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 396–403.

[17] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, "Sequence-to-sequence piano transcription with transformers," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 246–253.

[18] L. Ou, Z. Guo, E. Benetos, J. Han, and Y. Wang, "Exploring transformer's potential on automatic piano transcription," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*. IEEE, 2022, pp. 776–780.

[19] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. H. Engel, "MT3: multi-task multitrack music transcription," in *The Tenth International Conference on Learning Representations, ICLR*. OpenReview.net, 2022.

[20] M. Won, S. Chun, and X. Serra, "Toward interpretable music tagging with self-attention," *arXiv preprint arXiv:1906.04972*, 2019.

[21] M. Won, K. Choi, and X. Serra, "Semi-supervised music tagging transformer," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 769–776.

[22] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: a fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, vol. 5, no. 50, p. 2154, 2020, deezer Research. [Online]. Available: https://doi.org/10.21105/joss.02154

[23] M. E. Davies, S. Böck, and M. Fuentes, *Tempo, Beat and Downbeat Estimation*, 2021. [Online]. Available: https://tempobeatdownbeat.github.io/tutorial/intro.html

[24] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *arXiv preprint arXiv:2106.04554*, 2021.

[25] S. Böck and M. Schedl, "Enhanced beat tracking with context-aware neural networks," in *Proceedings of the 14th International Conference on Digital Audio Effects, DAFx*, 2011, pp. 135–139.

[26] S. Böck, F. Krebs, and G. Widmer, "A multi-model approach to beat tracking considering heterogeneous music styles," in *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR*, 2014, pp. 603–608.

[27] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *The 9th ISCA Speech Synthesis Workshop*. ISCA, 2016, p. 125.

[28] C. Chiu, J. Ching, W. Hsiao, Y. Chen, A. W. Su, and Y. Yang, "Source separation-based data augmentation for improved joint beat and downbeat tracking," in *29th European Signal Processing Conference, EUSIPCO*. IEEE, 2021, pp. 391–395.

[29] C. Chiu, A. W. Su, and Y. Yang, "Drum-aware ensemble architecture for improved joint musical beat and downbeat tracking," *IEEE Signal Processing Letters*, vol. 28, pp. 1100–1104, 2021.

[30] R. Dai, S. Das, L. Minciullo, L. Garattoni, G. Francesca, and F. Brémond, "PDAN: pyramid dilated attention network for action detection," in *IEEE Winter Conference on Applications of Computer Vision, WACV*. IEEE, 2021, pp. 2969–2978.

[31] N. Moritz, T. Hori, and J. L. Roux, "Capturing multi-resolution context by dilated self-attention," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*. IEEE, 2021, pp. 5869–5873.

[32] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, Volume 2*. Association for Computational Linguistics, 2018, pp. 464–468.

[33] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[34] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "madmom: A new python audio and music signal processing library," in *Proceedings of the 2016 ACM Conference on Multimedia Conference, MM*. ACM, 2016, pp. 1174–1178.

[35] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An experimental comparison of audio tempo induction algorithms," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1832–1844, 2006.

[36] F. Krebs, S. Böck, and G. Widmer, "Rhythmic pattern modeling for beat and downbeat tracking in musical audio," in *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR*, 2013, pp. 227–232.

[37] S. W. Hainsworth and M. D. Macleod, "Particle filtering applied to musical tempo tracking," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 15, pp. 1–11, 2004.

[38] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical and jazz music databases," in *ISMIR 2002, 3rd International Conference on Music Information Retrieval*, 2002.

[39] O. Nieto, M. McCallum, M. E. P. Davies, A. Robertson, A. M. Stark, and E. Egozy, "The harmonix set: Beats, downbeats, and functional segment annotations of western popular music," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR*, 2019, pp. 565–572.

[40] A. Srinivasamurthy and X. Serra, "A supervised approach to hierarchical metrical cycle tracking from audio music recordings," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*. IEEE, 2014, pp. 5217–5221.

[41] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon, "Selective sampling for beat tracking evaluation," *IEEE Transactions on Speech and Audio Processing*, vol. 20, no. 9, pp. 2539–2548, 2012.

[42] G. Tzanetakis and P. R. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[43] U. Marchand and G. Peeters, "Swing ratio estimation," in *Proceedings of the 18th International Conference on Digital Audio Effects, DAFx*, 2015.

[44] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 2, pp. 236–243, 1984.

[45] N. Perraudin, P. Balázs, and P. L. Søndergaard, “A fast griffin-lim algorithm,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA*. IEEE, 2013, pp. 1–4.

[46] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” in *Proceedings of the 14th python in science conference*, vol. 8. Citeseer, 2015, pp. 18–25.

[47] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, “On the variance of the adaptive learning rate and beyond,” in *8th International Conference on Learning Representations, ICLR*. OpenReview.net, 2020.

[48] M. R. Zhang, J. Lucas, J. Ba, and G. E. Hinton, “Lookahead optimizer: k steps forward, 1 step back,” in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2019, pp. 9593–9604.

[49] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[50] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *9th International Conference on Learning Representations, ICLR*, 2021.

[51] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *Proceedings of the 38th International Conference on Machine Learning, ICML*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 10 347–10 357.

[52] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *2021 IEEE/CVF International Conference on Computer Vision, ICCV*. IEEE, 2021, pp. 9992–10 002.

SKETCHING THE EXPRESSION: FLEXIBLE RENDERING OF EXPRESSIVE PIANO PERFORMANCE WITH SELF-SUPERVISED LEARNING

Seungyeon Rhyu[1] **Sarah Kim**[2] **Kyogu Lee**[1,3]
[1] Music and Audio Research Group (MARG), Seoul National University, South Korea
[2] Krust Universe, South Korea
[3] Graduate School of AI, AI Institute, Seoul National University, South Korea
rsy1026@snu.ac.kr, estelle.kim@krustuniverse.com, kglee@snu.ac.kr

ABSTRACT

We propose a system for rendering a symbolic piano performance with flexible musical expression. It is necessary to actively control musical expression for creating a new music performance that conveys various emotions or nuances. However, previous approaches were limited to following the composer's guidelines of musical expression or dealing with only a part of the musical attributes. We aim to disentangle the entire musical expression and structural attribute of piano performance using a conditional VAE framework. It stochastically generates expressive parameters from latent representations and given note structures. In addition, we employ self-supervised approaches that force the latent variables to represent target attributes. Finally, we leverage a two-step encoder and decoder that learn hierarchical dependency to enhance the naturalness of the output. Experimental results show that our system can stably generate performance parameters relevant to the given musical scores, learn disentangled representations, and control musical attributes independently of each other.

1. INTRODUCTION

Computational modeling of expressive music performance focuses on mimicking human behaviors that convey the music [1, 2]. For piano performance, one common task is to *render* an expressive performance from a quantized musical score. It aims to reproduce the loudness and timing of musical notes that fits to the given score. Most of the conventional studies have used musical scores of Western piano music that includes sufficient amount of guidelines for musical expressions [3–6]. Recent studies using deep learning methods have successfully rendered plausible piano performances that are comparable to those of professional pianists from the given Classical scores [7–9].

More recently, it has increased attention to *controlling* music performance by manipulating one or more *disentangled* representations from a generative model. These representations are sensitive to the variation of certain factors while invariant to other factors [10]. Maezawa *et al.* aimed to control a performer's interpretation through a conditional variational recurrent neural network (CVRNN) [11]. They intended to disentangle a time-variant representation of the personal interpretation. In the acoustic domain, Tan *et al.* proposed a generative model based on a Gaussian mixture variational autoencoder (GM-VAE) that separately controlled dynamics and articulations of the notes [12]. Their novelty lied in learning multiple representations of high-level attributes from the low-level spectrogram.

However, these studies have constrained musical creativity. Maezawa *et al.* controlled musical expression only through quantized features from the musical scores. Tan *et al.* did not consider controlling tempo or timing with a latent representation. These methods may have restricted any potential for rendering piano performances with flexible musical expression. Musical creativity can be expanded not only by composers but also by performers who can elastically choose various strategies to highlight multiple nuances or emotions [13–15]. Moreover, the music generation field can be also broadened if static music created by automatic composition systems can be easily colored with realistic and elastic expression [16].

Therefore, we attempt a new approach that renders piano performances with flexible musical expressions. We disregard a typical assumption from previous studies that a performer must follow a composer's intent [4, 17–19]. According to the literature, performers learn to identify or imitate "expressive models", or *explicit planning*, of existing piano performances [20]. We focus on this attribute, defining it as a higher-level *sketch* of the expressive attributes (i.e. dynamics, articulation, and tempo [21]) that the performer draws based on a personal interpretation of the musical piece [4, 11, 20]. We also assume that the remaining attribute represents common performing strategies that are connected to certain musical patterns, while these strategies slightly differ across performers [22, 23]. We call this attribute as a *structural attribute* that belongs to given note structures of a musical piece.

In this study, we propose a generative model that can flexibly control the entire musical expression, or the explicit planning, of symbolic piano performance [1]. Our system is based on a conditional variational autoencoder (CVAE) that is modified for sequential data [11, 24]. The system generates multiple parameters of piano performance from a note structure of a musical passage, using disentangled representations for the explicit planning and structural attribute.

We employ a self-supervised learning framework to force the latent representations to learn our target attributes [24–26]. In addition, we facilitate independent control of the three expressive attributes–dynamics, articulation, and tempo–by utilizing an existing method that aligns the latent code with a target attribute [27,28]. Finally, we design a novel mechanism that intuitively models a polyphonic structure of piano performance. In particular, we insert intermediate steps for *chordwise* encoding and decoding of the piano performance to our encoder-decoder architecture, where a *chord* denotes a group of simultaneous notes.

Our approach has several contributions as follows: 1) Our system aims to control musical expression while maintaining any characteristics induced by a given musical structure; 2) We use self-supervised learning where new supervisory signals are involved in regularizing the latent representations effectively; 3) Our system aims to control multiple expressive attributes independently of each other; 4) Lastly, we leverage an intermediate step that projects a notewise representation into the chordwise in the middle of our system to intuitively model the polyphonic structure of piano performance.

2. PROPOSED METHODS

We aim to build a generative model that factorizes expressive piano performance as the explicit planning and structural attribute. The model is based on a conditional variational autoencoder (CVAE) that reproduces performance parameters based on a given musical structure.

2.1 Data Representation

We extract features that represent a human performance and the corresponding musical score, following the conventional studies [11, 19, 29].

Performance Features. We extract three features that represent the expressive attributes of each performed note, respectively: **MIDIVelocity** is a MIDI velocity value that ranges from 24 to 104. **IOIRatio** represents an instantaneous variation in tempo. We compute an inter-onset-interval (IOI) between the onset of a note and the mean onset of the *previous* chord for both a performed note and the corresponding score note. Then, a ratio of performed IOI to score IOI is calculated, clipped between 0.125 and 8, and converted into a logarithmic scale [4]. **Articulation** represents how much a note is shortened or lengthened compared to the instantaneous tempo. It is a ratio of a performed duration to an IOI value between the onset of a note and mean onset of the *next* chord [19]. It is clipped between 0.25 and 4 and converted into a logarithmic scale.

Score Features. The features for a musical score represent eight categorical attributes for how the notes are composed: **Pitch** is a MIDI index number that ranges from 21 to 108. **RelDuration** and **RelIOI** are 11-class attributes of a quantized duration and IOI between a note onset and a previous chord, respectively. They range from 1 to 11, and each class represents a multiple of a 16th note's length with respect to a given tempo [30, 31]. **IsTopVoice** is a binary attribute of whether the note is the uppermost voice. It is heuristically computed regarding pitches and durations of surrounding notes. **PositionInChord** and **NumInChord** are 11-class attributes of a positional index of a note within its chord and the total number of notes in that chord, respectively, that range from 1 to 11. An index 1 for PositionInChord denotes the most bottom position. **Staff** is a binary attribute of the staff of a note, either of the G clef or F clef. **IsDownbeat** is a binary attribute of whether a note is at a downbeat or not.

2.2 Modeling Musical Hierarchy

Inspired by previous studies [4, 8, 9, 32], we build a two-step encoder and decoder: An encoder models both notewise and chordwise dependencies of the inputs, and a decoder reconstructs the notewise dependency from the chordwise representation and the notewise condition. We denote a *chord* as a group of notes that are hit simultaneously, regardless of the staff, so that they sound together at an instant time [33]. Thus, learning the chordwise dependency is analogous to direct modeling of the temporal progression of the piano performance. Let $\mathcal{M} \in \mathbb{R}^{C \times N}$ be a matrix that aligns serialized notes to their polyphonic structure, where C and N are the number of chords and the number of notes, respectively. Within the encoder, the notewise representation is sequentially average-pooled by $\mathcal{M}$ with dynamic kernel sizes where each size represents the number of notes in each chord. We denote this operation as *N2C*. In this way, we can directly model chord-level dependency of the note-level expressive parameters [32]. In contrast, the decoder extends the chordwise representation from the encoder back to the notewise using the transposed alignment matrix $\mathcal{M}^T$, of which process we denote as *C2N*. Along this, the notewise embedding of the score features replenishes the notewise information for the output. Consequently, notes in the same chord *share* any information of their corresponding chord, while maintaining their differences by the conditional score features:

$$\text{N2C}(e) = \frac{\mathcal{M} \cdot e}{\sum_{n=1}^{N} \mathcal{M}_{n,1:C}}, \quad \text{C2N}(e) = \mathcal{M}^{\text{T}} \cdot e \qquad (1)$$

where e denotes a notewise or chordwise representation.

2.3 Overall Network Architecture

Our proposed network is generally based on the conditional VAE framework [34, 35]. Concretely, we use the sequential VAE that is modified for generation of sequential data [11,24,36]. Let $x = \{x_n\}_{n=1}^{N}$ be a sequence of the

[1] https://github.com/rsy1026/sketching_piano_expression