Expressor: A Transformer Model for Expressive MIDI Performance

Tolly Collins and Mathieu Barthet

Centre for Digital Music, Queen Mary University of London
tollycollins@gmail.com, m.barthet@qmul.ac.uk

Abstract. The Transformer neural network has been used to generate new music with expressive features with significant success, but it has not previously been applied to generate an expressive performance of an existing score. We propose Expressor, a Transformer model with a novel encoder-decoder skip connection design for expressive performance rendering. The model shows promise in applying coherent temporal and dynamics expressive features based on human performance. We develop a new tokenisation scheme to overcome challenges in representing interrelated expressive performance features.

1 Introduction and Related Work

We outline here a work in progress on how deep learning can be used to alter temporal and dynamics properties of a MIDI score to add similar expressive properties to those present in a human performance. Previous studies have applied Recurrent Neural Networks to model expressive timing [1, 2], and while they found success in modelling periodic expressive events, they performed less well for isolated events used to convey emotion or meaning. Transformers have shown promise in music generation tasks [3], where they have been more adept at modelling the longer-term structural properties of a musical score. We propose Expressor, a new Transformer model for expressive performance rendering with skip connections between corresponding encoder and decoder layers, and a new tokenisation scheme to represent expressive features. To our knowledge, this is a novel architecture and we find that the skip connections improve performance over the original design.

2 Methodology

Dataset. We use the ASAP dataset [4], with 1067 professionally-performed classical piano pieces with paired performed and unperformed MIDI versions.
Tokenisation. We use a compound word tokenization [5], with a metric rather than absolute timing representation inspired by the REMI approach [6]. The perceptual, hierarchical and interdependent nature of expressive attributes poses a significant

challenge in determining ground truth values. For example, note onset deviations are relative to local tempo, but tempo is itself a subjective measure that continually fluctuates over time. Furthermore, preceding notes may themselves deviate from precise metrical timings. Our solution is to provide the model with ground truths calculated relative to a piecewise constant tempo function with jumps at beat times (see Fig. 1). For example, timing deviations are calculated as the difference (as a proportion of beat length) between the actual note onset and expected onset given by adding a linear proportion of the beat length on from the start time of the beat. Expressive features for dynamics follow a similar hierarchical classification [7], and our model also considers articulation by varying note length relative to the notated version to produce more staccato or legato phrasings.

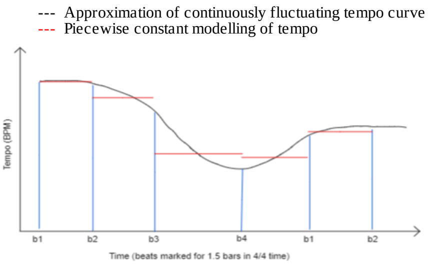

Fig. 1. Illustration of tempo modelling.

Table 1. Token Descriptions

Name	Type	Description
Type	Meta	Determines if a word is *meta* (for start- and end-of-sequence), *metric* (occurring at the start of each beat) or *note* (each word corresponds with exactly one note).
Beat	Metric	Hold the number of the beat in a bar.
IBI	Metric	Inter-beat interval. Express the tempo as a quantized beat length in seconds.
Local vel. band	Metric	Coarse measure of MIDI velocity.
Local IBI	Metric	The median IBI over a number of beats spanning closest to 4 seconds, centred on the beat relating to the given metric word.
Pitch	Note	The MIDI pitch number of a note (integer between 1 and 127).
Start	Note	Score-based start position of a note relative to the beat, given as a proportion of the beat (quantized to 1/60 beats).
Duration	Note	Number of beats a note is designated to last for in the score, quantized to 1/60 beats.
Rubato	Note	Designates any beat marked with rubato in the ASAP dataset annotations, meaning that the music departs from standard metrical timing during this beat.
Timing flux	Metric	Mean deviation in onset of notes in a beat from the precise division of the IBI.
Dynamic flux	Metric	The average number of absolute standard deviations for the velocity of each note in a given beat from the local mean.
Accent	Note	Designed to represent an accent score notation. Calculated as any performed note having a velocity of more than 2 standard deviations above the local mean.
Staccato	Note	Whether or not a note should last for < 25% of the expected IBI proportion.
Local vel. mean	Metric	The mean note velocity over a given number of beats, centred on the current beat.
Tempo difference	Metric	Difference between a beat's IBI and the local tempo, measured in BPMs.
Articulation	Note	How long a note will last for, relative to the expected duration taken from the score. The value is a number of beats, quantized to a given sub-interval.
Timing deviation	Note	The sub-interval of a beat by which the note onset differs the score.
Vel. difference	Note	Difference between a note's velocity and the local velocity mean.

Model. We use the Transformer with Linear Attention design [8] in an encoder-decoder format. The aim is for the encoder to create a representation of score-specific structural information such as note pitches, medium-term tempo and general dynamics. We design for additional attribute tokens input directly to the encoder output latent space, allowing for control to be imposed on the generation akin to score markings guiding a pianist. The decoder layers then output words containing tokens with expressive properties such as *timing deviation* per note or *local mean velocity*. We also introduce the use of skip connections (see Fig. 2) between the outputs of individual encoder layers and the attention mechanism in the corresponding decoder layer. The idea is to encourage corresponding hierarchical representations of the information throughout the encoder and decoder stacks.

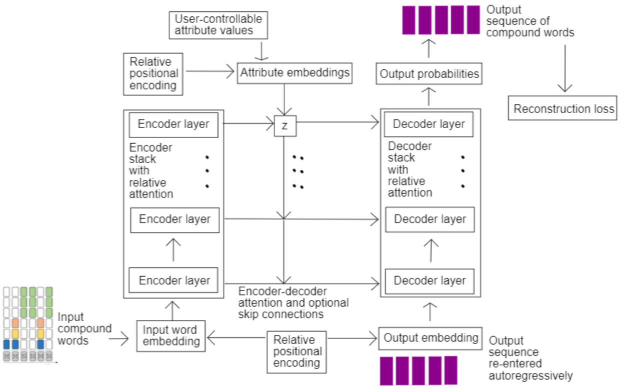
User-controllable attribute values
Output sequence of compound words
Relative positional encoding
Attribute embeddings
Output probabilities
Reconstruction loss
Encoder layer
z
Decoder layer
Encoder stack with relative attention
Decoder stack with relative attention
Encoder layer
Decoder layer
Encoder layer
Decoder layer
Encoder-decoder attention and optional skip connections
Input compound words
Input word embedding
Relative positional encoding
Output embedding
Output sequence re-entered autoregressively

Fig. 2. Expressor Architecture

Output tokens are then combined with the input information to render back into MIDI format. This results in a version of the original piece that incorporates expressive performance features. As each compound word is made up of separate tokens, the network decoder is followed by one head for each output token which consists of a separate feed-forward network to map latent space vectors to logits for the relevant values in the token's vocabulary. The network can therefore be viewed as a multi-task network, and the loss is made up of a linear combination of the reconstruction losses for each head.

3 Results Discussion and Conclusion

Model. With hyperparameter tuning, we found the best performing model had encoder and decoders both with dimension 256, 8 layers and 8 attention heads per layer. Fig. 3 shows the results from two training runs with these same model parameters, but one with added skip connections between corresponding encoder and decoder layers. The results suggest that training is improved by the addition of the skip connections.

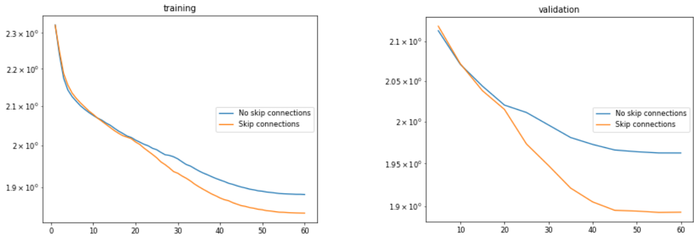

Fig. 3. Training and validation loss curves for identical Expressor models with and without skip connections between the encoder and decoder layers

Intuitively, the skip connections may encourage the model to match hierarchical levels in the music between encoder and decoder stacks.

Music Transformers often use embedding dimensions larger than vocabulary sizes [3]. The use of compound words allows for tailored embedding sizes for each token type, and we found that embedding sizes between 4 and 16 performed better than larger values. As described in Table 1, many of the measures used in Expressor represent a quantized linear scale, such as IBI or pitch, and although there may be some higher-dimensional relationships such as the chroma dimension for pitch, in general this data should not require large numbers of dimensions to represent.

Qualitative evaluation and discussion. Selected audio examples can be found at the link below[1]. While we have yet to conduct independent listening tests, we suggest these demonstrate that the model shows considerable promise in mapping general expressive performance features onto a MIDI score in a realistic manner. The features often follow locally coherent patterns such as *crescendi* or *staccato*. We did notice that the expressive features could often be inappropriate in relation to the musical period or the commonly interpreted emotional content. Additional tokens such as *composer* or *period*, alongside planned latent space semantic guidance tokens could help. We have yet to analyse statistically how well the model relates expressive features to structural features in the score (such as musical phrasing or unexpected harmonic moments), but our intuition is that pre-training the structural modelling of the encoder section may improve performance in this area. We also intend to conduct an ablation study to further understand the impact of the encoder-decoder skip connections.

References

1. Shi, Z.: Computational analysis and modeling of expressive timing in Chopin Mazurkas. In: Proc. of the 22nd Int. Society for Music Information Retrieval Conf., Online (2021).
2. Jeong, D., Kwon, T., Kim, Y., Lee, K., and Nam, J.: VirtuosoNet: A Hierarchical RNN-based system for modeling expressive piano performance. In: 20th International Society for Music Information Retrieval Conference, Delft, The Netherlands (2019).
3. Wu, S.-L., and Yang, Y.-H.: MuseMorphose: Full-Song and Fine-Grained Music Style Transfer with One Transformer VAE. IEEE (2021). arXiv:2105.04090v3
4. Foscarin, F., McLeod, A., Rigaux, P., Jacquemard, F. and Sakai, M.: ASAP: a dataset of aligned scores and performances for piano transcription. In: 21st International Society for Music Information Retrieval Conference, Montréal, Canada (2020).
5. Hsiao, W.-Y., Liu, J.-Y., Yeh, Y.-C., and Yang, Y.-H.: Compound Word Transformer: Learning to Compose Full-Song Music Over Dynamic Directed Hypergraphs. In: Proc. of the AAAI Conference on Artificial Intelligence (2021).
6. Huang, Y.-S., and Yang, Y.-H.: Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions. In: Proc. 28th Int. Conf. on Multimedia (2020).
7. Oore, S., Simon, I., Dieleman, S, Eck, D.: This time with feeling: Learning expressive musical performance. In*:* Neural Computing and Applications 32.4, pp. 955–967 (2018).
8. Katharopoulos, A., Vyas, A., Pappas, N. and Fleuret, F: Transformers are RNNs: Fast autoregressive Transformers with linear attention. In: Proc. Int. Conf. Machine Learning (2020).

[1] https://drive.google.com/drive/folders/1JwENHB5iOSYsl5FPYeeoWW2q46PESTfy?usp=share_link