Supplementary Material
NeurIPS 2021 submission ID 6474

1 Generated Samples and Overview Video

Please see the attached video for a short overview and samples of video clips that were sounded by RhythmicNet. Please turn **Audio ON**. Additional video samples are located in the **video_samples** folder. Due to the 100MB size limit of supplementary material, we posted additional samples on **figshare** and here is the anonymous private link to access them.

2 Human Perceptual Evaluation Generated Music by RhytmicNet

In addition to the objective evaluation of the different components of RhythmicNet presented in the main paper, we also performed human perceptual survey using Amazon Mechanical Turk. We performed the survey in order to provide an additional source of evaluation to the generated samples. In the survey we asked people (non-experts) to watch the same human activity video with different soundtracks and answer the question: "in which video the sounds best match the movements". The given options of the generated soundtracks were Random, Shuffle and RhythmicNet. The Random drum track was generated from *Rhythm2Drum* method with a random rhythm with 50% chance to be ON or OFF at each time step - orange in Fig. 1. The chance of 50% was chosen such that there is a significant probability that the a rhythm that sounds like a real rhythm will be sampled.will be sampled. We found that sampling with lower probability would generate rhythms that do not sound well. The Shuffle drum track was generated from Rhythm2Drum method but the order is shuffled - grey in Fig. 1. Ours option corresponded to drum track generated from Rhythm2Drum method using our generated rhythm - blue in Fig. 1. No background on the survey or the RhythmicNet project was given to the participants to avoid any perceptual biases. We surveyed 50 participants individually, where each participant was asked to evaluate 10 videos each with around 10 seconds (500 segments in total) along with three generated soundtracks. The reward per assignment was 0.05$. The results are shown in Fig. 1. We observe a clear indication that the drum tracks generated by our method are chosen to be the best match to the movements more frequently (41.4% (Ours) v.s. 30.8% (Random) and 27.8% (Shuffle)).

3 Additional Implementation Details

3.1 Data Preprocessing

Video2Rhythm. For all videos, we use the BODY_25 pose model to extract keypoints. We remove the noisy keypoints correspond to "eyes", "ears", "nose", and "small toe". After removing these keypoints we end up with a total of 17 keypoints:: "neck", "shoulders", "elbows", "wrists", "mid hip", "hips", "knees", "ankles", "heels" and "big toe". For missed detections, we use the spline interpolation to recover them. The keypoints are defined by their pixel position (x, y) within a video frame on the basis of the OpenPose output. We convert the absolute position of the keypoint at time t into its displacement in time: $(\Delta x_t, \Delta y_t) = (x_t - x_{t-1}, y_t - y_{t-1})$.

Rhythm2Drum. We follow the preprocessing in [1]. We map all drum hits in the Groove Midi Dataset to a smaller set of 9 canonical drum categories. These categories represent the most common instruments in standard drum kits: bass drum, snare drum, hi-hats, toms, and cymbals. We slide fixed size windows across all full sequences to create drum patterns of fixed length. We use 2 bar patterns for all experiments, sliding

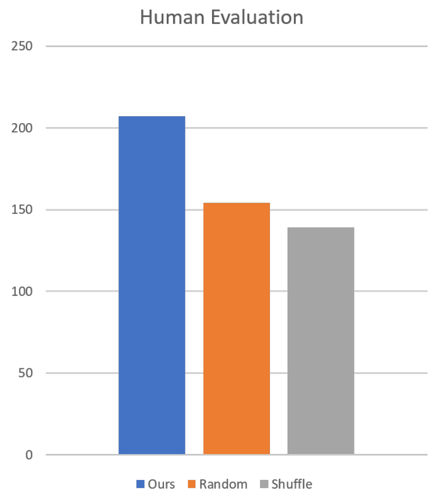

Figure 1: Human evaluation result

the window with a hop size of 1 bar. Finally, we take 16th notes as the fundamental timestep of data. Each drum hit is associated with the closest 16th note metrical position. For more details, please refer to [1].
Drum2Music. For Drum2Piano and Drum2Guitar, we extract two subsets of Lakh Midi dataset [2]. We select the Midi files that are have 4/4 time signature and has at least 16 bars. We convert all Midi files into Remi representation following the [3]. Unlike the original implementation, we save each bar separately to allow our model to learn the bar level correspondence between drum and another instrument. The Remi representation contain Note_On, Note_Duration, Position, Tempo_Class, Tempo_Value, Chord and Bar events, resulting in a vocabulary size of 308.

3.2 Architecture Details

Video2Rhythm. The detailed configuration of Video2Rhythm model is listed in Table 1.

Video2Rhythm Configuration	Value
Keypoints Dimension	17 × 2
input length	300
ST-GCN layers	10
Motion features dimension	64
Embedding Dimension	64
Self-Attention Layers	2
Self-Attention Hidden Size	64
Self-Attention Inner linear hidden Size	256
Self-Attention Heads	2
Dropout(Attention and Residual)	0.1
Optimizer	Adam
Learning rate	0.001
batch size	64

Table 1: Video2Rhythm model configuration.

Rhythm2Drum. The detailed configurations of the transformer and Unet are listed in Table 2 and Table 3, respectively.

Drum2Music. The detailed configuration of Drum2Music model is listed in Table 4.

Rhythm2Drum Transformer Configuration	Value
Vocabulary Size	152
Encoder & Decoder input length	32
Embedding Dimension	64
Encoder Layers	3
Decoder Layers	3
Encoder & Decoder Hidden Size	128
Encoder & Decoder Inner linear hidden Size	256
Encoder & Decoder Attn. Heads	4
Dropout(Attention and Residual)	0.1
Optimizer	Adam
Learning rate	0.001
batch size	64

Table 2: Rhythm2Drum transformer configuration.

Rhythm2Drum Unet Configuration	Value
Input Dimension	$1 \times 9 \times 32$
Down-sample depth levels	5
Down-sample channels	16, 32, 64, 128, 128
Up-sample depth levels	5
Up-sample channels	128, 64, 32, 16, 4
Activation function	ReLU, LeakyReLu(0.2)
Output Dimension	$2 \times 9 \times 32$
Dropout	N/A
Optimizer	Adam
Learning rate	0.001
batch size	64

Table 3: Rhythm2Drum Unet model configuration.

4 Dataset License

The AIST Dance Video Database is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). The Groove Midi Dataset is made by Google LLC under a Creative Commons Attribution 4.0 International License (CC BY 4.0). The Lakh Midi Dataset is distributed with a CC-BY 4.0 license.

Drum2Music Configuration	Value
Vocabulary Size	308
Encoder & Decoder input length	256
Encoder & Decoder memory length	256
Embedding Dimension	512
Encoder Layers	4
Decoder Layers	8
Encoder & Decoder hidden size	512
Encoder & Decoder Inner linear size	2048
Encoder & Decoder Attn. Heads	8
Dropout	0.1
Optimizer	Adam
Learning rate	0.0002
batch size	16
Total Number of Parameters	49M

Table 4: Drum2Music model configuration.

References

[1] Jon Gillick, Adam Roberts, Jesse Engel, Douglas Eck, and David Bamman. Learning to groove with inverse sequence transformations. In *International Conference on Machine Learning*, pages 2269–2279. PMLR, 2019.

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

[3] 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*, pages 1180–1188, 2020.