MIDI Generation with PerTok and a Non-Autoregressive Transformer

1st Weixi Zhai
Tan Siu Lin Business School
Quanzhou Normal University
Quanzhou, China
wishzhai@gmail.com ORCID(0009-0006-2698-8549)

Abstract—We present a system for expressive piano performance rendering from MusicXML for RenCon 2025. The model is a non-autoregressive Transformer trained with PerTok tokenization: pre-trained on the ASAP corpus using only Beethoven and Chopin, then fine-tuned on ATEPP v1.2 with a focused sample of 200 pieces. At inference, we use a template-constrained pipeline that writes MIDI directly from the score, applying only expressive parameters predicted by the model, which guarantees structural fidelity. We submit MIDI renderings and the accompanying code and checkpoints.

I. Introduction

For this submission, I selected Beethoven: 32 Variations in C minor, WoO 80 - Theme and the first 5 variations and Rachmaninoff: (How Fair this Spot), Op. 21, No. 7 (transcribed for solo piano) as the required pieces, and Mozart's Piano Sonata No. 16 in C major, K.545, Allegro (mm.1–16) as the free-choice piece.

II. Methodology

We use a non-autoregressive Transformer with a PerTok vocabulary (covering velocity, duration, microtiming, etc.). The model is pre-trained for 40 epochs on an ASAP subset focused on Beethoven and Chopin to learn stable expressive priors, then fine-tuned for 8 epochs on ATEPP at a lower learning rate to broaden style coverage and subtlety.

At inference, we parse the input MusicXML into a note list (onset, pitch, nominal duration, part) and render MIDI directly rather than using PerTok detokenization. This template-constrained approach locks the score's structure and ensures deterministic, reproducible outputs. Expression is added by combining model predictions with three simple, musically motivated controls: (1) melody–accompaniment voicing that boosts the highest note at each time (velocity +10) and slightly attenuates accompaniment (velocity -5), (2) phrase-shaped dynamics that use slur-based phrase detection to apply an arch toward phrase peaks (up to +15 velocity), and (3) structural rubato that lingers on phrase peaks by adding small positive onset delays (+40 ticks). The model provides velocity and duration predictions from PerTok token distributions, but we disable its microtiming predictions in favor of rule-based rubato for more controlled expression. No quantization or human correction is applied; we only clamp velocities to valid ranges (0-127) and enforce proper event ordering (note off before note on for same-time events).

III. Datasets

For pre-training, we use the ASAP corpus with a focused subset of Beethoven and Chopin piano works.

For fine-tuning, we use the ATEPP dataset version 1.2. From the full ATEPP corpus containing thousands of pieces, we randomly sample 200 high-quality pieces for focused fine-tuning. Both datasets are split into train/validation/test sets (70/15/15).

IV. Post-processing

- If there is controllability of the models: Is the initial tempo being chosen? Dynamics range? Is there any other human-intervened design choices that goes into the model (e.g. annotated piece structure)?
  The initial tempo is automatically extracted from the input MusicXML file using music21 parsing.
  For dynamics, the model predicts velocity tokens (0-127 range)
  Human-intervened design:Melody +10, accompaniment −5, and phrase boost up to +15 are MIDI velocity adjustments in units 0–127 (clamped to [0,127]). The +40-tick delay is an onset timing offset in MIDI ticks; with ticks-per-beat = 480, the delay at tempo BPM is delay-ms = 60000/BPM × (40/480) (e.g., 83 ms at 60 BPM). We use a template-constrained direct-MIDI renderer applying only model-predicted velocity, onset timing offsets, and duration scaling, ensuring 100% structural fidelity with no quantization or manual correction.
- For systems producing symbolic outputs: Was there any quantization or human correction involved before generating the final MIDI?
  No quantization and no human correction were applied. We render at the score's native MIDI tick resolution using the model's predicted onset offsets and duration scaling directly; only automatic safeguards

are used (clamping velocity to [0,127], enforcing note-on/off order, and preventing negative durations).

- If submitting audio from symbolic outputs: Was sonification done via software or a player piano? Specify soundfonts, rendering tools, or piano model used.

  The symbolic outputs (MIDI) were rendered to audio using MuseScore 4 with its default soundfont, exported as WAV at 44.1 kHz.
- Was any other human intervention involved in the final submission? No