

2024 edition

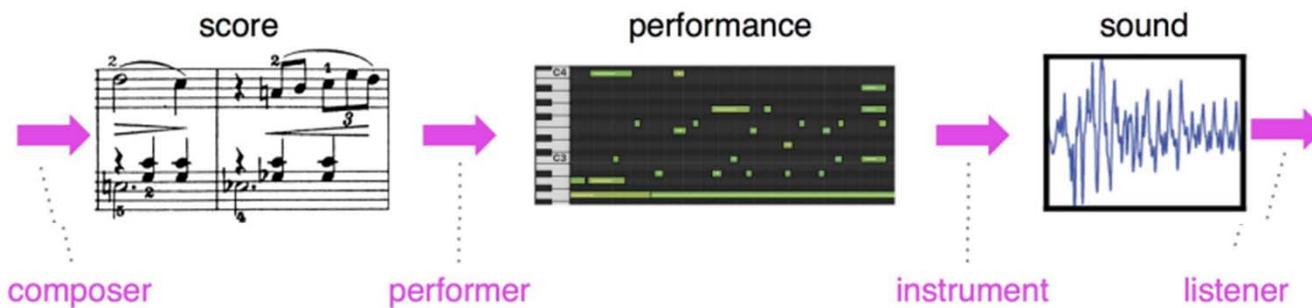
Deep Learning for Music Analysis and Generation

Fundamentals for Symbolic Music



Yi-Hsuan Yang Ph.D.
yhyangtw@ntu.edu.tw

Why Symbolic Music



- There are meaningful problems in symbolic music **analysis** and **generation**
- Reasons to work on symbolic music generation
 - Focus on the musical content instead of sounds
 - Care about long-range structure instead of acoustic quality or diversity
 - Less compute-demanding (compared to text-to-music generation)
 - Good first step toward generative systems for music

Outline

- **Sheet music & symbolic representations for music**
 1. Sheet music
 2. Piano roll
 3. MIDI
 4. MusicXML
 5. ABC
- Token representation of MIDI music

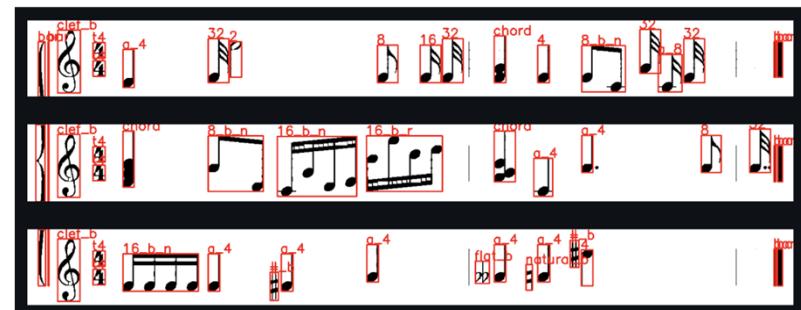
Sheet Music

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S1_SheetMusic.html

- A *visual* representation (*.PDF, *.PNG, etc)
 - A **guide** for performing a piece of music leaving room for different interpretations
 - Musicians may vary the **tempo**, **dynamics**, and **articulation**, thus resulting in a personal interpretation of the given musical score
- **Rarely** directly used as input to neural network models
 - Exception: *optical music recognition* (OMR)



Figure 1.1 from [Müller, FMP, Springer 2015]



Sheet Music: Key Signature & Note Duration

<https://nicechord.com/post/major-and-minor-modes/>

<https://nicechord.com/post/notation-essentials/>

C major: 全全半全全全半



A musical staff with a treble clef. It consists of eight notes: a whole note (C4), a half note (D4), a quarter note (E4), a half note (F4), a whole note (G4), a half note (A4), a whole note (B4), and a half note (C5). The notes are separated by vertical bar lines.

C minor: 半全全全全半全



A musical staff with a treble clef. It consists of eight notes: a half note (C4), a whole note (D4), a half note (E♭4), a whole note (F4), a half note (G4), a half note (A♭4), a whole note (B♭4), and a half note (C5). The notes are separated by vertical bar lines.

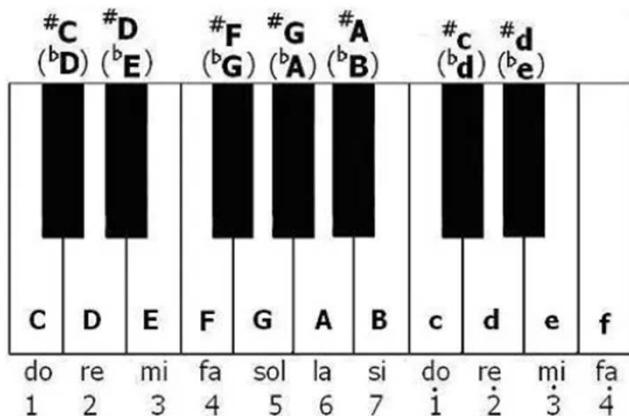
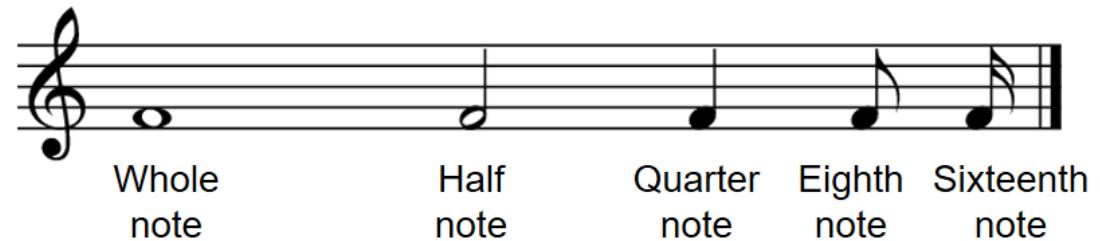


Figure 1.7 from [Müller, FMP, Springer 2015]

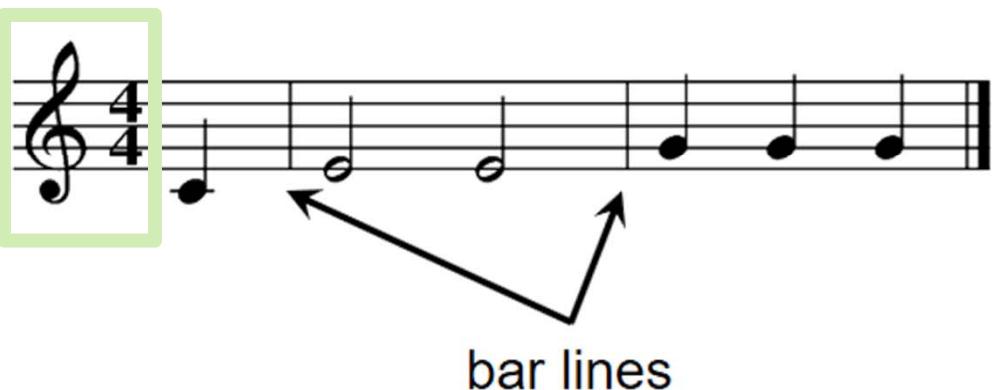


A musical staff showing five notes of decreasing size. From left to right, the notes are: Whole note, Half note, Quarter note, Eighth note, and Sixteenth note. Below each note, its name is written: Whole note, Half note, Quarter note, Eighth note, Sixteenth note.

Sheet Music: Meter, Bar, Beat, Rhythm

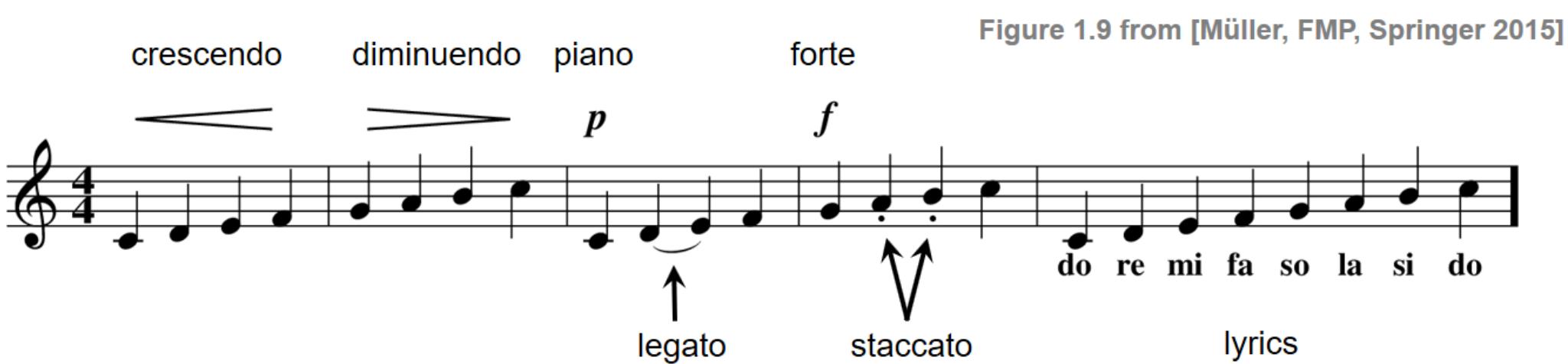
- Dividing music into measures (bars) not only reflects its rhythmic nature, but also provides regular reference points within it

Figure 1.6 from [Müller, FMP, Springer 2015]



Sheet Music: Articulation Marks

- How certain notes are to be played



Sheet Music vs. Lead Sheet

A S I T W A S for string orchestra

Words and Music by Harry Styles,
Thomas Hull, and Tyler Johnson
arr. Tate Commission

Driving Rock ♩ = 174

Violin I

Violin II

Viola

Violoncello

Contrabass

Vln. I

Vln. II

Vla.

Vc.

Ch.

D 4/4

015 一件美事

C E_m A_m D_m F

5 | 1 1 2 3 5 5 3 2 2 1 - - - | 0 2 2 3 4 3 2 |

已 過 二十 世 紀 以 來， 千 千 萬 萬 寶 貴

A_m G₇ E_m A_m D_m G₇ E_m

1. 2 2 3 4 | 5. 1 1 1 2 3 | 4. 3 2 3 4 | 5 5 5 1 |

的 性 命 心 愛 的 奇 珍， 崇 高 的 地 位 以 及 燦 爛 的 前

F E_m A_m D_m G₇ C F G₇

6 6 5 4 | 5. 1 1 2 3 | 4. 3 2 1 7 | 1 - - 1 1 | 6. 6 5 |

途， 都 曾 枉 費 在 主 耶 穌 身 上， 對 這 些 愛 主

C F G₇ C D_m G₇

2 | 3 - - 1 1 | 6 - 5 2 | 3 - - 2 3 | 4 4 4 3 2 2 2 3 |

的 人， 祂 是 全 然 可 愛， 祂 是 全 然 可 愛 配

D_m G₇ F G₇ C A_m

4 4 3 4 5 5 6 | 5. 0 1 || 6. 6 7 . 7 | 1 7 6 6 6 7 |

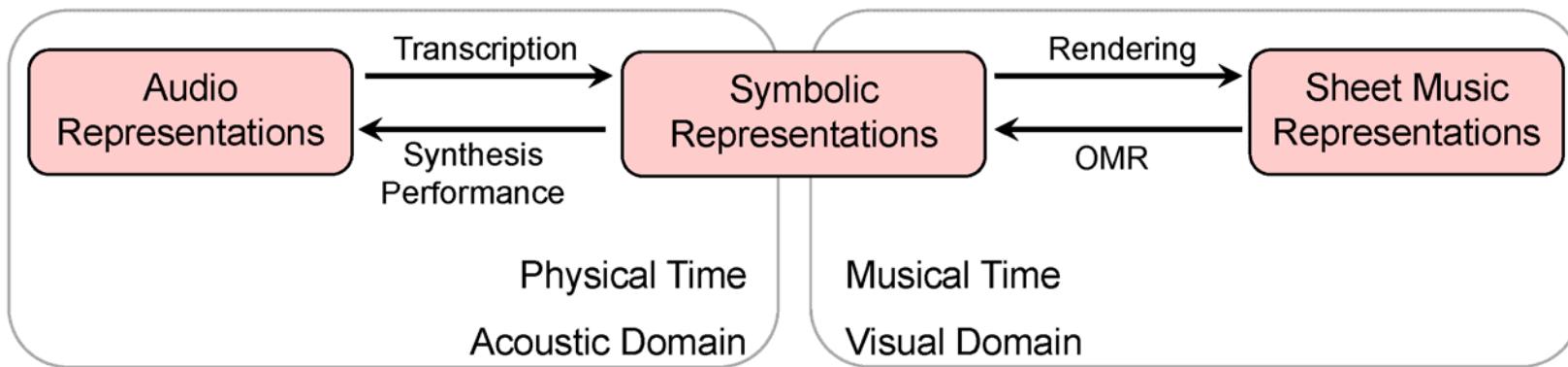
得 我 們 獻 上 一 切。 我 們 濡 在 主 身 上 的 不 是

C D_m G₇ D_m G₇ C C

1 5 5 - 4 3 | 4 4 4 3 2 2 2 3 | 4. 3 2 2 | 3 -- 0 1 || 1 -- ||

枉 費， 乃 是 馨 香 的 見 證， 見 證 祂 的 甘 甜。 我 甜。

Sheet Music and Symbolic-domain Representations



- **Musical time:** 1/4, 1/8, 1/16 etc, good for representation **scores**
- **Sheet music:** visual representations of a musical score either given in printed form or encoded digitally in some image format
- **Physical time:** in milliseconds or alike, good for representation **performances** (*tempo, dynamics, and articulation*)
- **Symbolic representations**

Ref: <https://www.mdpi.com/2624-6120/2/2/18>

Piano Roll Representation

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S2_PianoRoll.html

- A *visual, image-like* representation of *.MIDI (also how DAWs visualize MIDI files)

- Horizontal axis: **Time**
 - Vertical axis: **Pitch**
 - Rectangle: **Note**
 - Leftmost point: **Onset**
 - Lowermost point: **Pitch**
 - Width: **Duration**
 - (Can also indicate **Velocity**)

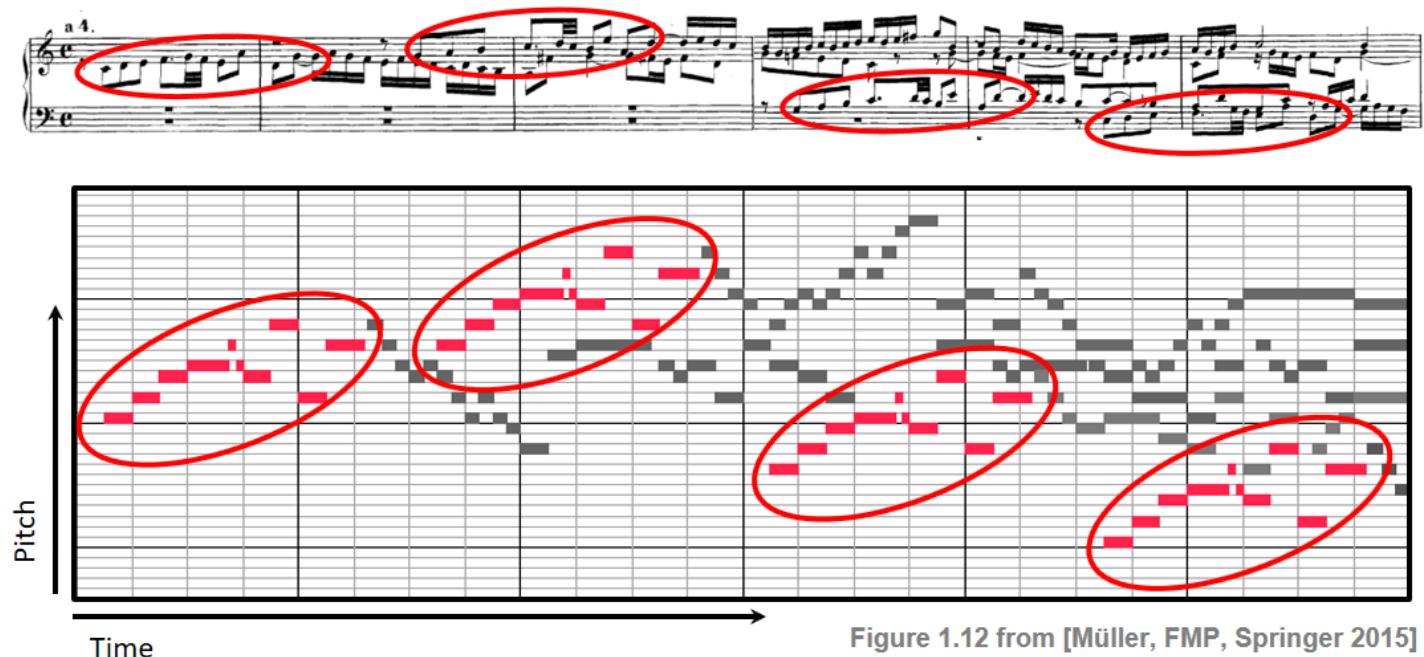
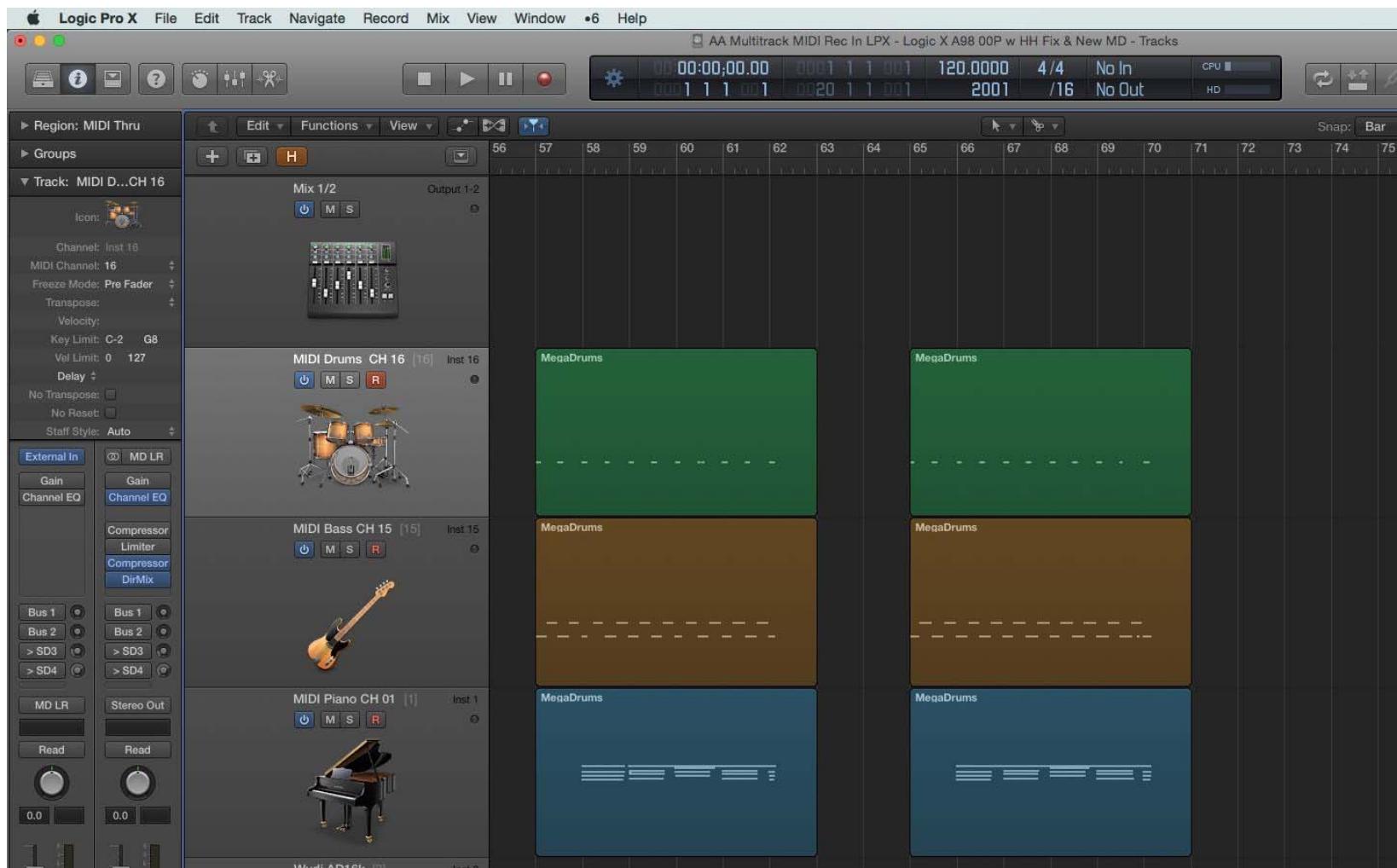


Figure 1.12 from [Müller, FMP, Springer 2015]

ps. The four occurrences of the theme of the music are highlighted

Piano Roll Representation

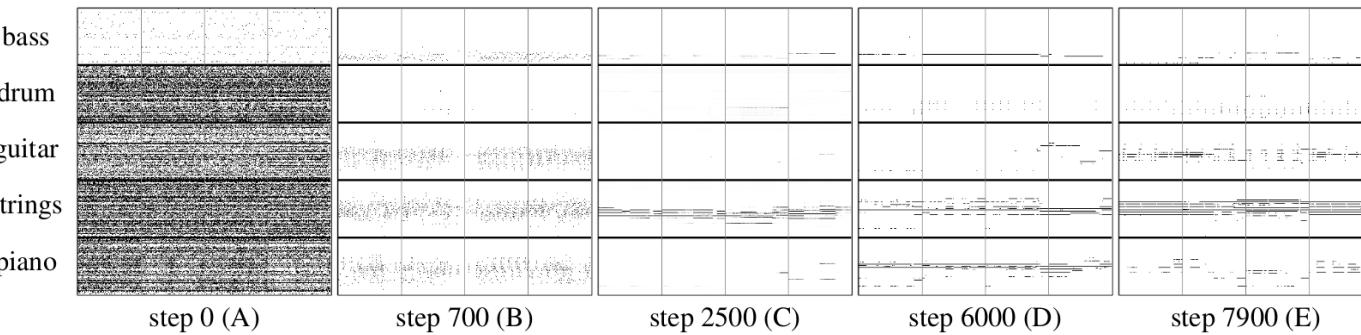


Source: <https://www.macprovideo.com/article/audio-software/logic-pro-x-a-guide-to-multitrack-midi-recording>

Piano Roll Representation

- A *visual, image-like* representation of *.MIDI
- Multi-track MIDI → multiple piano rolls (i.e., a tensor)
- **Widely** used by deep generative neural networks
 - *MidiNet* (2017) [GAN-based]
 - *MuseGAN* (2018) [GAN-based]: <https://salu133445.github.io/musegan/results>
 - *DiffRoll* (2023) [diffusion-based]: <https://polyffusion.github.io/>

MuseGAN

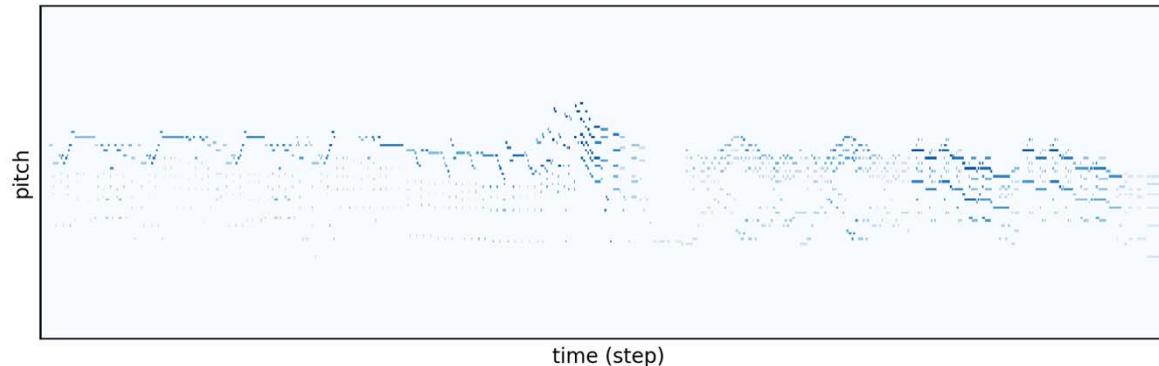


DiffRoll



Library: PyPianoroll

<https://salu133445.github.io/pypianoroll/>



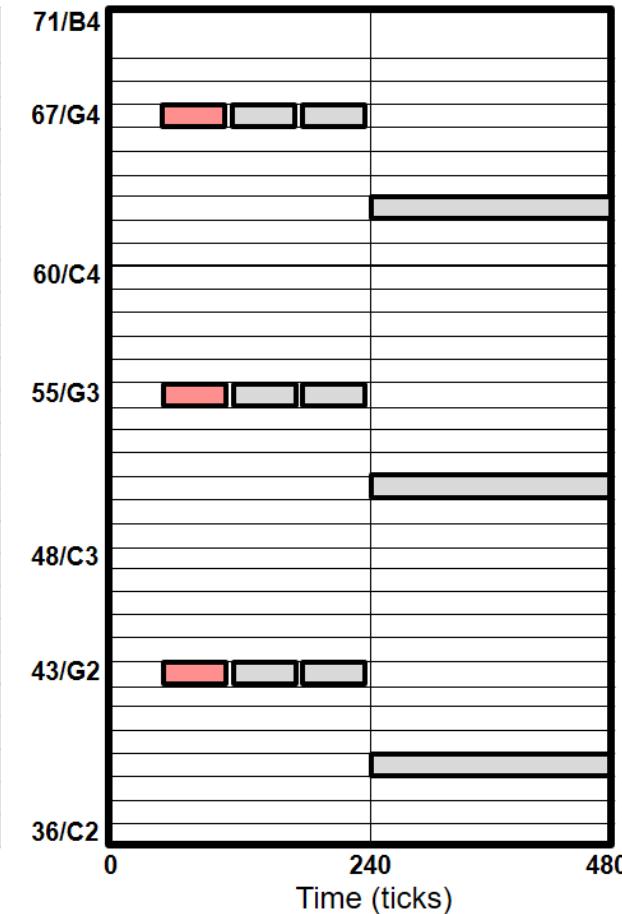
- Features
 - Manipulate multitrack piano rolls intuitively
 - Visualize multitrack piano rolls beautifully
 - Save and load multitrack piano rolls in a space-efficient format
 - Parse **MIDI** files into multitrack piano rolls
 - Write multitrack piano rolls into **MIDI** files

MIDI Representation

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S2_MIDI.html

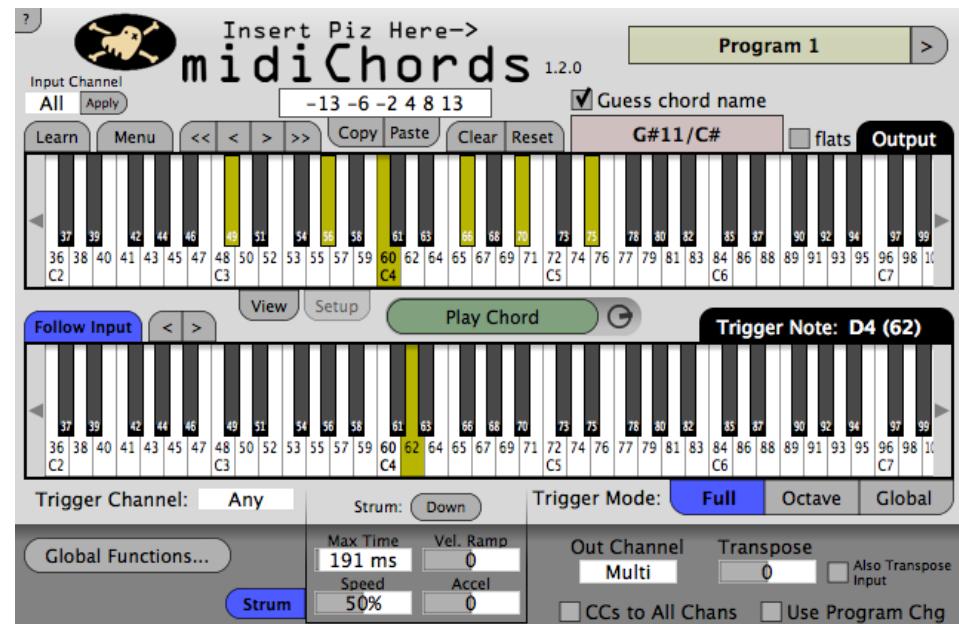
- A ***text-like*** representation of
*.MIDI, using **MIDI messages**
and **timestamps**
 - *MIDI note number* (0-127)
 - *Key velocity* (0-127): intensity
of the sound
 - *MIDI channel* (different
instruments)
 - *Time stamp*: how many *clock
pulses or ticks to wait* before
the command is executed

Time (Ticks)	Message	Channel	Note Number	Velocity
60	NOTE ON	1	67	100
0	NOTE ON	1	55	100
0	NOTE ON	2	43	100
55	NOTE OFF	1	67	0
0	NOTE OFF	1	55	0
0	NOTE OFF	2	43	0
5	NOTE ON	1	67	100
0	NOTE ON	1	55	100
0	NOTE ON	2	43	100
55	NOTE OFF	1	67	0
0	NOTE OFF	1	55	0
0	NOTE OFF	2	43	0
5	NOTE ON	1	67	100
0	NOTE ON	1	55	100
0	NOTE ON	2	43	100
55	NOTE OFF	1	67	0
0	NOTE OFF	1	55	0
0	NOTE OFF	2	43	0
5	NOTE ON	1	63	100
0	NOTE ON	2	51	100
0	NOTE ON	2	39	100
240	NOTE OFF	1	63	0
0	NOTE OFF	2	51	0
0	NOTE OFF	2	39	0



Piano Roll vs MIDI Representation

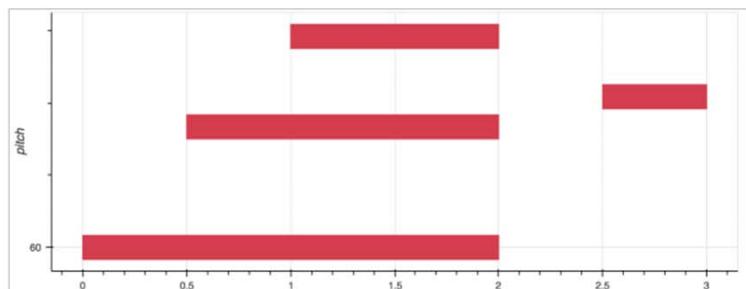
- Both represent a *.MIDI file
 - Piano roll: image-like
 - MIDI Representation: text-like
- Piano roll are like fixed-shape images
- MIDI representation is more flexible as it allows for *additional information* to be added
 - For example, **chord tokens**, **key tokens**, etc



Ref: Figure from <https://www.kvraudio.com/product/midichords-by-insert-piz-here>

MIDI Representation

- A ***text-like*** representation of *.MIDI
- ***Widely*** used by deep generative neural networks, especially Transformers
 - By viewing the MIDI messages as “**tokens**”
 - *Music Transformer* (2019) : <https://magenta.tensorflow.org/music-transformer>
 - *MuseNet* (2019): <https://openai.com/research/musenet>
 - *Pop Music Transformer* (2020): <https://github.com/YatingMusic/remi>
 - *MuseCoco* (2023): <https://ai-muzic.github.io/musecoco/>



```
SET_VELOCITY<80>, NOTE_ON<60>
TIME_SHIFT<500>, NOTE_ON<64>
TIME_SHIFT<500>, NOTE_ON<67>
TIME_SHIFT<1000>, NOTE_OFF<60>, NOTE_OFF<64>,
NOTE_OFF<67>
TIME_SHIFT<500>, SET_VELOCITY<100>, NOTE_ON<65>
TIME_SHIFT<500>, NOTE_OFF<65>
```

Timing in Piano Roll & MIDI

- The time axis can either be in *absolute timing* or in *symbolic timing*
 - For **absolute** timing, the actual timing of note occurrence is used, for example by indicating the tempo for each bar or each beat
 - For **symbolic** timing, the **tempo** information is removed and thereby each beat has the same length
- MIDI subdivides a **quarter note** into basic time units referred to as *clock pulses* or *ticks*
 - Pulses per quarter note (PPQN): commonly 120
 - PPQN determines the resolution of the time stamps associated to note events

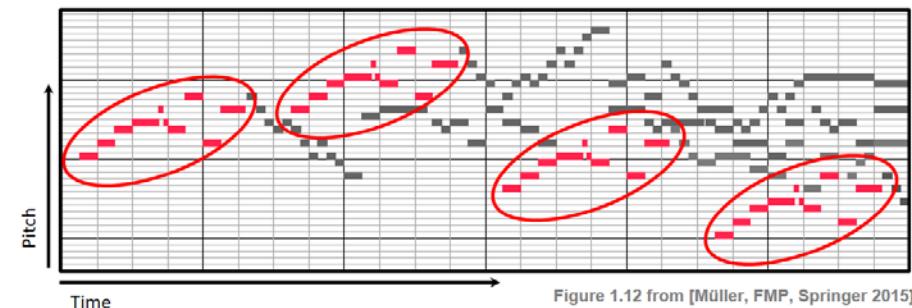
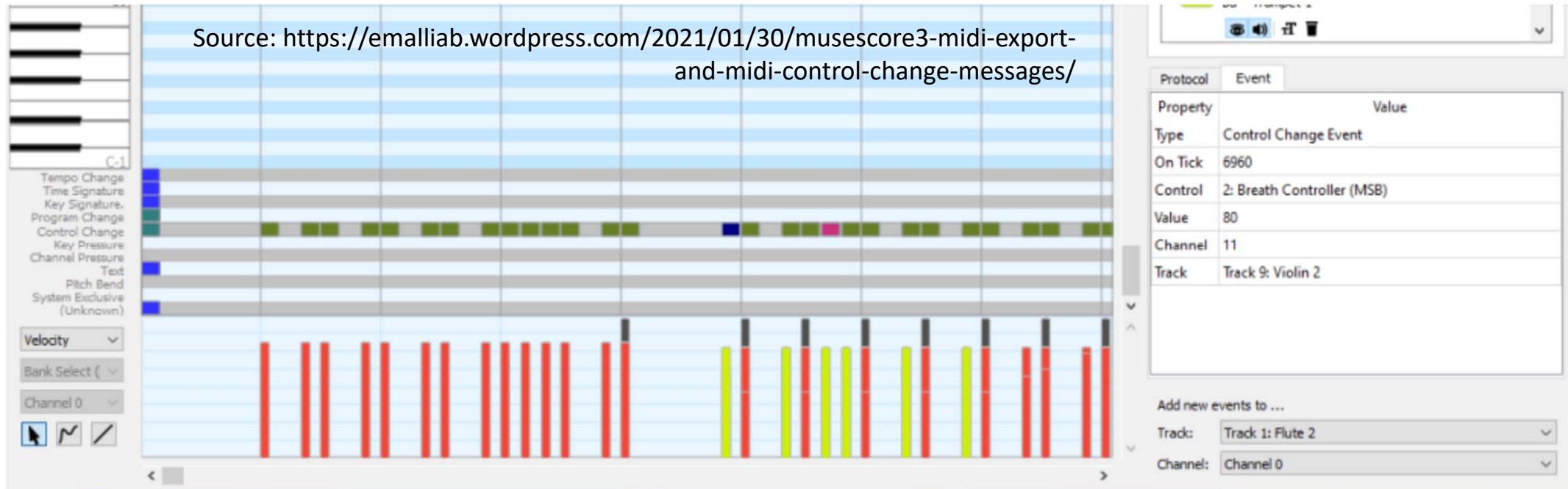


Figure 1.12 from [Müller, FMP, Springer 2015]

Ref1: <https://salu133445.github.io/lakh-pianoroll-dataset/representation.html>

Ref2: https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S2_MIDI.html

Program Change or Control Change (CC) Messages in MIDI



- Tempo change
- Program change
- Control change

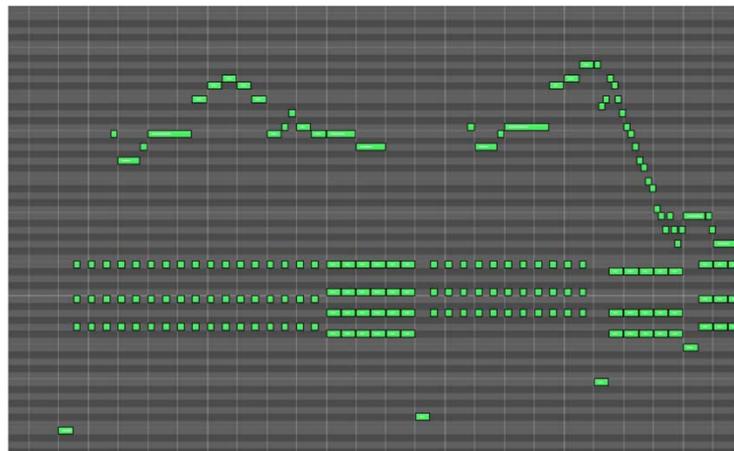
Protocol	Event	Protocol	Event	Protocol	Event
Property	Value	Property	Value	Property	Value
Type	Program Change Event	Type	Program Change Event	Type	Program Change Event
On Tick	768	On Tick	1535	On Tick	3069
Program	1: Bright Acoustic Piano	Program	2: Electric Grand Piano	Program	3: Honky-tonk Piano
Channel	0	Channel	0	Channel	0
Track	Track 0: Asturias(Leyenda)	Track	Track 0: Asturias(Leyenda)	Track	Track 0: Asturias(Leyenda)

Source: <https://gigperformer.com/how-to-automate-switching-rackspace-variations-and-song-parts-using-the-midi-file-player/>

MIDI Score vs MIDI Performance

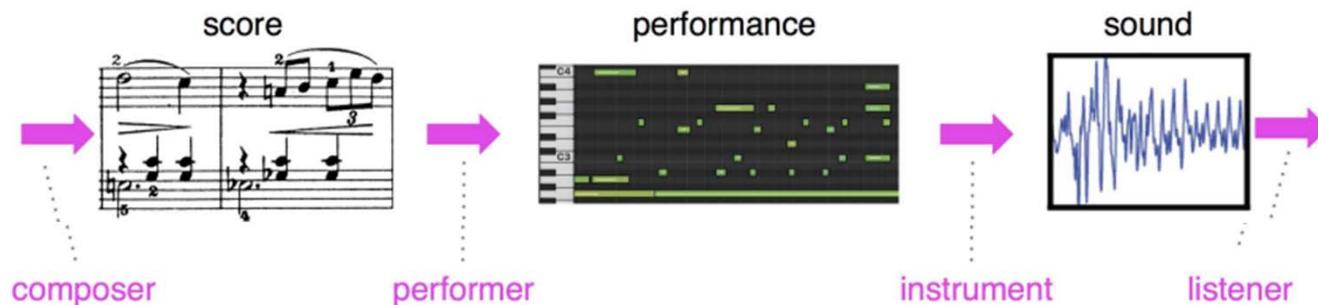


Figure 1: Excerpt from the score of Chopin's Piano Concerto No. 1.



- MIDI score
 - <https://clyp.it/jhdkgħso>
- MIDI performance
 - <https://clyp.it/x24hp1pq>

MIDI Score vs MIDI Performance



- **MIDI score**
 - A score that has been rendered **directly** into a MIDI file
 - Rendered with **NO** dynamics and **NO** expressive timing (i.e., exactly according to the written metrical grid)
- **MIDI performance**
 - A score has been performed, and that performance has been encoded into a MIDI stream, through either **MIDI keyboards** or by **automatic music transcription**
 - With expressive timing, tempo changes, dynamics, and articulation

Library: Miditoolkit

<https://github.com/YatingMusic/miditoolkit>

- **Miditoolkit** is designed for handling MIDI in **symbolic timing** (ticks), which is the native format of MIDI timing
- **PrettyMIDI** (<https://github.com/craffel/pretty-midi>) can parse MIDI files and generate pianorolls in absolute timing
- **PyPianoroll** can parse MIDI files into pianorolls in symbolic timing
- **mido** (<https://github.com/mido/mido>) processes MIDI files in the lower level such as messages and ports
- **music21** (<https://web.mit.edu/music21/>) provides analysis modules

MusicXML

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S2_MusicXML.html

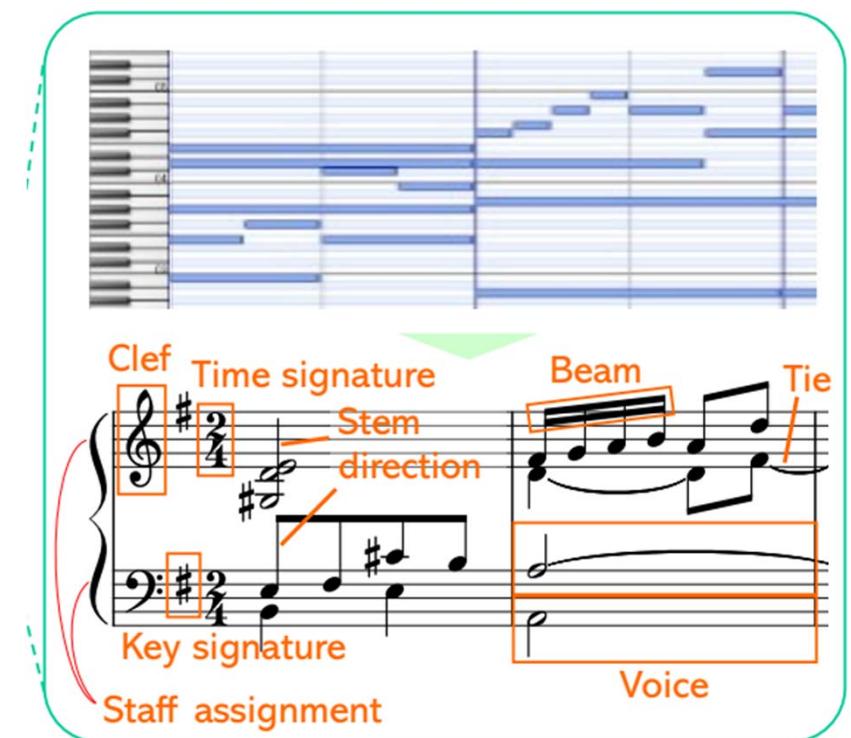
- A ***text-like*** representation of **sheet music**
 - To represent a <note>
 - <pitch>: <step>, <alter> <octave>
 - <duration>
 - <type>
- Can be **tokenized**
- Musical (not physical) onset times are specified
- Avoid information loss of the piano roll representation
 - Example 1, the notes **D♯4** and **E♭4** are distinguishable in MusicXML but not in MIDI

```
<note>
  <pitch>
    <step>E</step>
    <alter>-1</alter>
    <octave>4</octave>
  </pitch>
  <duration>2</duration>
  <type>half</type>
</note>
```



MusicXML

- A *text-like* representation of **sheet music**
- Readily tokenized
- Musical (not physical) onset times
- Avoid information loss of the piano roll representation
 - Example 1, D♯4 and E♭4
(enharmonic equivalents; 同音異名)
 - <https://www.youtube.com/watch?v=Sm-2e2DsdhE>
 - Example 2: can describe **voices**



Ref: Suzuki, "Score Transformer: Generating musical score from note-level representation," MMAAsia 2021

Voices

- SATB
 - S: soprano
 - A: alto
 - T: tenor
 - B: bass

Voice type

Female

Soprano

Mezzo-soprano

Contralto

Male

Countertenor

Tenor

Baritone

Bass

Aus tiefer Not First Section

J. S. Bach

The musical score consists of two staves. The top staff is for the piano, indicated by the text "Piano" to its left. The bottom staff is for the vocal part. The music is in common time, with a key signature of one sharp (F#). The vocal line begins with a long note followed by a series of eighth notes. The piano accompaniment features eighth-note chords.

<https://www.youtube.com/watch?v=WD0jd1QcMQE>

MusicXML

- A *text-like* representation of **sheet music**
- *Relatively less* used by deep generative neural networks
 - *Score Transformer* (2021)



(a) Score

R bar clef_treble key_sharp_1 time_2/4 note_E4
note_D4 note_G#3 len_2 stem_up bar ... L bar
clef_bass key_sharp_1 time 2/4 <voice> note_E3
len_1/2 stem_up beam_start note_F#3 len_1/2
stem_up beam_continue note_C#4 stem_up
beam_continue note_B3 len_1/2 stem_up beam_stop
</voice> <voice> note_B2 len_1 stem_down note_E3
len_1 stem_down </voice> bar ...

(b) Notation-level Representation

Ref: Suzuki, “Score Transformer: Generating musical score from note-level representation,” MMAAsia 2021

Software: MuseScore

The screenshot shows the MuseScore application interface. At the top, there's a navigation bar with 'musescore', a search bar ('Search for Sheet music'), and buttons for 'Browse' and 'Learn'. A 'Start Free Trial' button is also visible. The main window displays a musical score for 'Amazing Grace' in 3/4 time with a key signature of one sharp. The score consists of two staves: treble and bass. The tempo is set to 72 BPM. Dynamics like 'mp' (mezzo-forte) and 'mf' (mezzo-forte) are indicated. A context menu is open over the score, titled 'Download this score', listing options: 'Musescore', 'PDF', 'MusicXML', 'MIDI', and 'Audio'. To the right of the score, there's a sidebar for the piece 'Amazing Grace' by user 'matt1738', showing statistics (35K views, 2.3K likes, 39 comments), download/print/share buttons, a rating section (5 stars), and a 'Try Shutter FLEX for f' advertisement.

Search for Sheet music

Browse Learn

Start Free Trial

musescore

Download this score

The score can be downloaded in the format of your preference:

Musescore

Open in Musescore

PDF

View and print

MusicXML

Open in various software

MIDI

Open in editors and sequencers

Audio

Listen to this score

Amazing Grace

matt1738

35K 2.3K 39 ★

Download Print

Favorite Share

Please rate this score

Try Shutter FLEX for f

Software: KernScores

Kern Scores

A library of virtual musical scores in the Humdrum **kern data format.
Total holdings: 7,866,496 notes in 108,703 files.

search: [browse](#) | [shortcuts](#) [Text](#) anchored

A guided tour of the KernScores website
Recent additions to the KernScores library
Data Collection Highlights

Composers				
Adam	Chopin	Giovannelli	Lassus	Schubert
Alkan	Clementi	Grieg	Liszt	Schumann
J.S. Bach	Corelli	Haydn	MacDowell	Scriabin
Banchieri	Dufay	Himmel	Mendelssohn	Sinding
Beethoven	Dunstable	Hummel	Monteverdi	Sousa
Billings	Field	Isaac	Mozart	Turpin
Bossi	Flecha	Ives	Pachelbel	Scarlatti
Brahms	Foster	Joplin	Prokofiev	Vecchi
Buxtehude	Frescobaldi	Josquin	Ravel	Victoria
Byrd	Gershwin	Landini	Scarlatti	Vivaldi
				Weber

Genres				
Ballate	Etudes	Motets	Scherzos	Symphonies
Ballads	Fugues	Preludes	Sonatas	Virelais
Chorales	Madrigals	Ragtime	Sonatina	Waltzes
Contrafacta	Mazurkas	Quartets		

VerovioHumdrumViewer Scarlatti, Sonata in C minor, L.10, K.84

File View Edit Analysis Scores Help A

```

1 !!!COM: Scarlatti, Domenico
2 !!!CDT: 1685-1757
3 !!!OTL: Sonata in C minor, L.10, K.84
4 !!!SCT: K. 84
5 !!!SCT: L. 10
6 !!!OMD: Allegro ([quarter]=152)
7 **kern **kern **dynam
8 *staff2 *staff1 *staff1/2
9 *Icemb a *Icemb a *
10 *I" L.10 (K.84) *I" L.10 (K.84)
11 *>[A,A1,A,A2,B,B1,B,B2] *>[A,A1,A,A2,B,
12 *>norep[A,A2,B,B2] *>norep[A,A2,B,
13 *>A *>A *>A
14 *clefF4 *clefG2 *
15 *k[b-e-a-] *k[b-e-a-] *
16 *M3/4 *M3/4 *
17 *MM152 *MM152 *
18 =1- =1- =1-
19 4CC 4C 8r f
20 . 8c'L .
21 4r 8e' .
22 . 8g' .
23 4r 8cc' .
24 . 8ee'-J .
25 =2 =2 =2
26 2.r (8gg^L .
27 . 8ccc) .
28 . 8gg' .
29 . 8ee'- .
30 . 8cc' .
31 . 8g'J .
32 =3 =3 =3
33 8r 4c' .
34 8c'l

```

ABC

https://en.wikipedia.org/wiki/ABC_notation

The Legacy Jig

jig

The musical score consists of two staves of music. The first staff starts at measure 1, and the second staff begins at measure 7. Both staves are in G major (one sharp) and 6/8 time. The music features sixteenth-note patterns. A digital player interface at the bottom shows a play button, a progress bar at 0:54 / 0:54, a repeat button, and a colon.

```
<score lang="ABC">
X:1
T:The Legacy Jig
M:6/8
L:1/8
R:jig
K:G
GFG BAB | gfg gab | GFG BAB | d2A AFD |
GFG BAB | gfg gab | age edB | 1 dBA AFD :| 2 dBA ABd |:
efe edB | dBA ABd | efe edB | gdB ABd |
efe edB | d2d def | gfe edB | 1 dBA ABd :| 2 dBA AFD | ]
</score>
```

ABC

<https://abcnotation.com/examples>

C, D, E, F, | G, A, B, C | D E F G | A B c d | e f g a | b c' d' e' | f' g' a' b' |

A/4 A/2 A/ A A2 A3 A4 A6 A7 A8 A12 A16

A>A A2>A2 | A<A A2<A2 | A>>A A2>>>A2 | A<<A A2<<<A2

(2AB (3ABA (4ABAB (5ABABA (6ABABAB (7ABABABA|

(AA) (A(A)A) ((AA)A) (A|A) A-A A-A-A A2-|A4|

__A _A =A ^A ^^A

"A"A "Gm7"D "Bb" F "F#" A



{g}A3 A{g}AA | {gAGAG}A3 {g}A{d}A{e}A |

ABC

https://abcnotation.com/tunePage?a=trillian.mit.edu/~jc/music/abc/src/jcabc2ps-20090401/abc/V_SAT1B2/0000

- Can support multiple voices

TEST: Vocal SATB score.

soprano
alto
tenor
bass

C D E F G A B c c d e f g a b c'
C D E F G A B c c d e f g a b c'
C, D, E, F, G, A, B, C C D E F G A B c
C,, D,, E,, F,, G,, A,, B,, C C,, D,, E,, F,, G,, A,, B,, C

www.abcnotation.com/tunes

```
X:1
T: TEST: Vocal SATB score.
N: All parts written "on the staff" with commas.
N: All parts should show scale from C below to above staff.
K: C
V:S name="soprano" clef=treble middle=B
V:A name="alto" clef=treble middle=B
V:T name="tenor" clef=tenor middle=B,
V:B name="bass" clef=bass middle=D,
%
V:S
| CDEF GABC | cdef gabc' |
w: C D E F G A B c | c d e f g a b c' |
V:A
| CDEF GABC | cdef gabc' |
w: C D E F G A B c | c d e f g a b c' |
V:T
| C,D,E,F, G,A,B,C | CDEF GABC |
w: C, D, E, F, G, A, B, C | C D E F G A B c |
V:B
| C,,D,,E,,F,, G,,A,,B,,C, | C,D,E,F, G,A,B,C |
w: C,, D,, E,, F,, G,, A,, B,, C, | C, D, E, F, G, A, B, C |
```

ABC

- A ***text-like*** representation of **sheet music**
- Readily tokenized
- Musical (not physical) onset times
- Can support multiple voices
- Support **score** only, not performance
- **Can be used by large language models (LLMs)**



can you generate music?



While I can't directly generate or play music files, I can help create music in ABC notation based on your specifications! If you provide details like the key, melody, or chords, I can generate the notation text for you, which you can then input into an ABC music editor (like EasyABC or ABCjs) to see the sheet music and hear the playback.

Would you like me to create something specific?

LLM ❤ ABC

Web-sourced Musical Knowledge

A chord, in music, is any harmonic set of pitches consisting of multiple notes (also called "tones") that are sounded simultaneously ...

Construct smooth-flowing chord progressions for the supplied music.
[: BA | G2 g2"(C)" edeg | B2 BA"(D7)"
BcBA | G2 g2 edeg | dBAG A2 BA | G2 ...]

Score Generation Instructions

Training



ChatMusician

Music Theory QA

Which of the following chord progressions best describes the above example?

Composing

Produce sheet music by implementing ABC notation based on the given text.
Rhythm: hornpipe, Key: Gmin, Note Length: 1/8, Meter: 4/4

X:1 L:1/8 Q:1/8=200 M:4/4 K:Gmin |:"Gm" BGdB ...

Chat

Please recommend a song from Taylor Swift.

Sure! One highly recommended song is "Blank Space" from her 2014 album "1989"...

ChatMusician: Understanding and generating music intrinsically with LLM, arXiv 2024

<https://github.com/hf-lin/ChatMusician>

Question: "Which of the following statements about triads is correct?",

Options:

A: "A triad can only be composed of three notes.",

B: "All triads are consonant chords.",

C: "When a triad is inverted, its properties and consonance remain consistent with its original position.",

D: "A triad is defined as a chord formed by stacking three notes in a third relationship."},

Answer: D

Library: EasyABC

<https://github.com/jwdj/EasyABC/>

[EasyABC](#) / midi2abc.py

Features

- Good ABC standard coverage thanks to internal use of abcm2ps and abc2midi
- Syntax highlighting
- Zoom support
- Import MusicXML, MIDI and Noteworthy Composer files.
- Export to MIDI, SVG, PDF (single tune or whole tune book).
- Select notes by clicking on them and add music symbols by using drop-down menus in the toolbar.
- Play the active tune as midi (using SoundFont)
- See which notes are currently playing (Follow score)

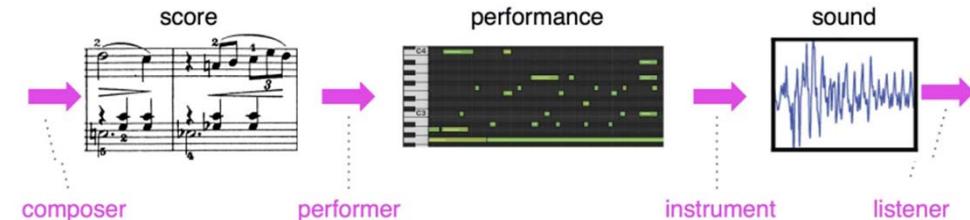
Summary

- **Sheet Music / MusicXML / ABC**

- Support **score** only, not performance
 - Use **music timing** only, not absolute timing
 - Richer information about the score itself

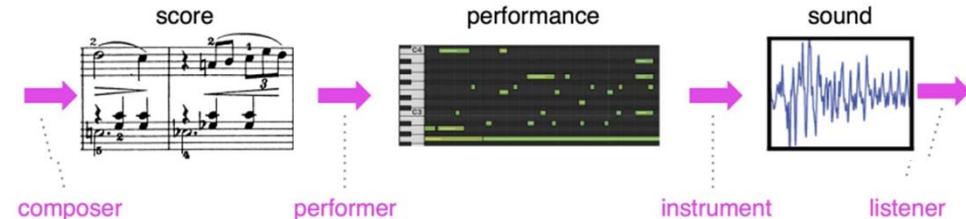
- **Piano roll / MIDI**

- Support either score or **performance**
 - Use either music timing (after quantization) or **absolute timing**
 - Rich performance information
 - Datasets more easily accessible and thereby **more widely used**



Summary (Cont')

- **ABC**
 - Good for simple melodies, chords, and basic arrangements
 - Limited for more complicated piano scores
 - Good for LLMs (but can only do composition; not performance)
- **Piano roll / MIDI / MusicXML**
 - More flexible



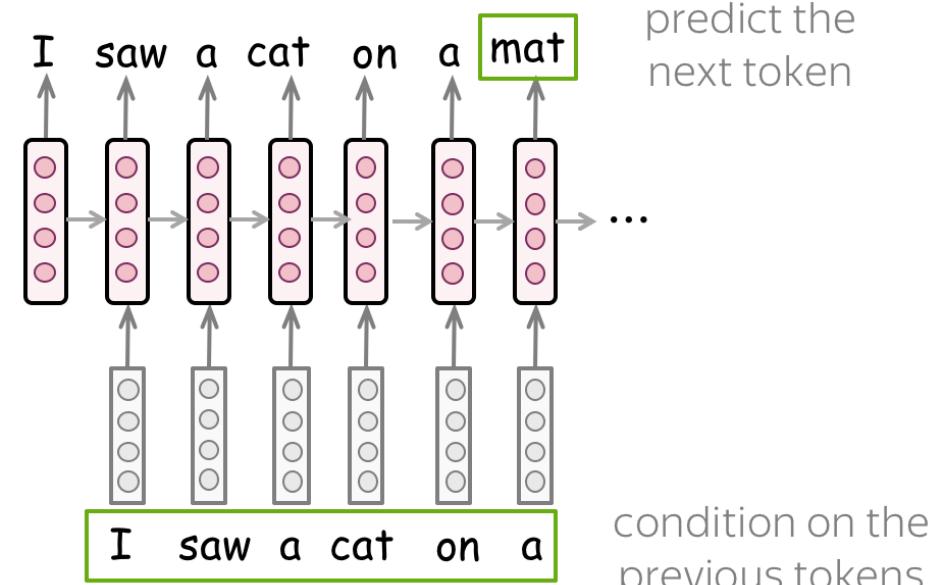
Outline

- Sheet music & symbolic representations for music
- **Token representation of MIDI music**

Text Tokens

<https://wikidocs.net/178448>

- Token
 - Words, character sets, or combinations of words
 - The **smallest unit of text data**
- First step in training LMs
- Two factors
 - Number of unique tokens (vocabulary size)
 - Length of a token sequence



Text Tokens

- Different types of text tokenizers

- Word-based tokenizers `["Let", "us", "learn", "tokenization."]`
 - Large vocabulary size
 - May have out-of-vocabulary (OOV; unknown) tokens
- Character-based tokenizers `["L", "e", "t", "u", "s", "l", "e", "a", "r", "n", "t", "o", "k", "e", "n", "i", "z", "a", "t", "i", "o", "n", "."]`
 - Small vocabulary size
 - No OOV
 - Long token sequence
- Sub-word tokenizers (e.g., byte-pair encoding) `["Let", "us", "learn", "token", "ization."]`
 - Middle ground

Ref 1: <https://medium.com/@nimritakoul01/tokenizers-for-natural-language-transformer-models-simply-explained-ee7167a844da>

Ref 2: https://blog.csdn.net/zhaohongfei_358/article/details/123379481

Text Tokens

<https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them>

<https://platform.openai.com/tokenizer>

My favorite color is red.

[3666, 4004, 3124, 318, 2266, 13]

Observations:

The more probable/frequent a token is, the lower the token number assigned to it:

- The token generated for 'red' varies depending on its placement within the sentence:
 - Lowercase in the middle of a sentence: ' red' - (token: "2266")
 - Uppercase in the middle of a sentence: ' Red' - (token: "2297")
 - Uppercase at the beginning of a sentence: 'Red' - (token: "7738")

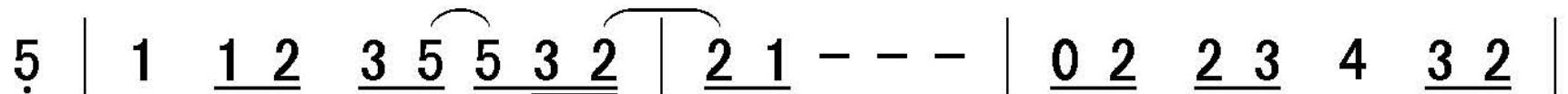
Representing Symbolic Music as Tokens

- How to represent a **melody** sequence?

5 | 1 1 2 3 5 5 3 2 | 2 1 - - - | 0 2 2 3 4 3 2 |

Representing Symbolic Music as Tokens

- How to represent a **melody** sequence?



- [pitch] [duration] [pitch] [duration] [pitch] [duration] [pitch] [duration] ...
- [pitch=G3] [duration=¼] [pitch=C4] [duration=¼] [pitch=C4] [duration=⅛] ...
- [G3] [¼] [C4] [¼] [C4] [⅛] [D4] [⅛] ...
- **Each musical note is uniformly represented by two tokens**
- **The ending of a note marks the beginning of the next note**

Representing Symbolic Music: Melody

- How to represent a **melody** sequence **with rest?**



Representing Symbolic Music: Melody

- How to represent a **melody** sequence **with rest?**



- Different methods
 1. [pitch] [duration] [rest-duration] [pitch] [duration] [pitch] [duration]
 2. [pitch] [duration] [pitch=0] [duration] [pitch] [duration] [pitch] [duration]
 3. [pitch] [duration] [rest= $\frac{1}{4}$] [pitch] [duration] [rest=0] [pitch] [duration] [rest=0]

Representing Symbolic Music: Lead Sheet

- How to represent a **lead sheet** sequence (i.e., melody+chord)?

C E_m A_m D_m F
5 | 1 1 2 3 5 5 3 2 | 2 1 - - - | 0 2 2 3 4 3 2 |

- [pitch] [duration] **[chord]** [pitch] [duration] [pitch] [duration] ...
 - Each musical note is still represented by two tokens, but there are *another type of tokens* (not for notes) in the token sequence
 - Getting more complicated

Representing Symbolic Music: Lead Sheet

- How to represent a **lead sheet** sequence (i.e., melody+chord+lyrics)?

C E_m A_m D_m F
5 | 1 1 2 3 5 5 3 2 2 1 - - - | 0 2 2 3 4 3 2 |
已 過 二十世紀 以 來， 千 千萬 萬 寶貴

$\text{♩} = 120$

The musical score consists of a treble clef staff in 4/4 time. The lyrics are written below the staff, corresponding to the notes. A pink box highlights a rest in the eighth measure, labeled "Rest". The score includes the following lyrics:

Listen to the rhy thm of the fall ing rain Tel ling me

Representing Symbolic Music: Piano Music Score

- How to represent a sequence of **piano music score**?



- The ending of a note is **no longer** the beginning of the next note
 - concurrent notes, temporally overlapping notes, ...
- Need a way to mark the “flow of time”

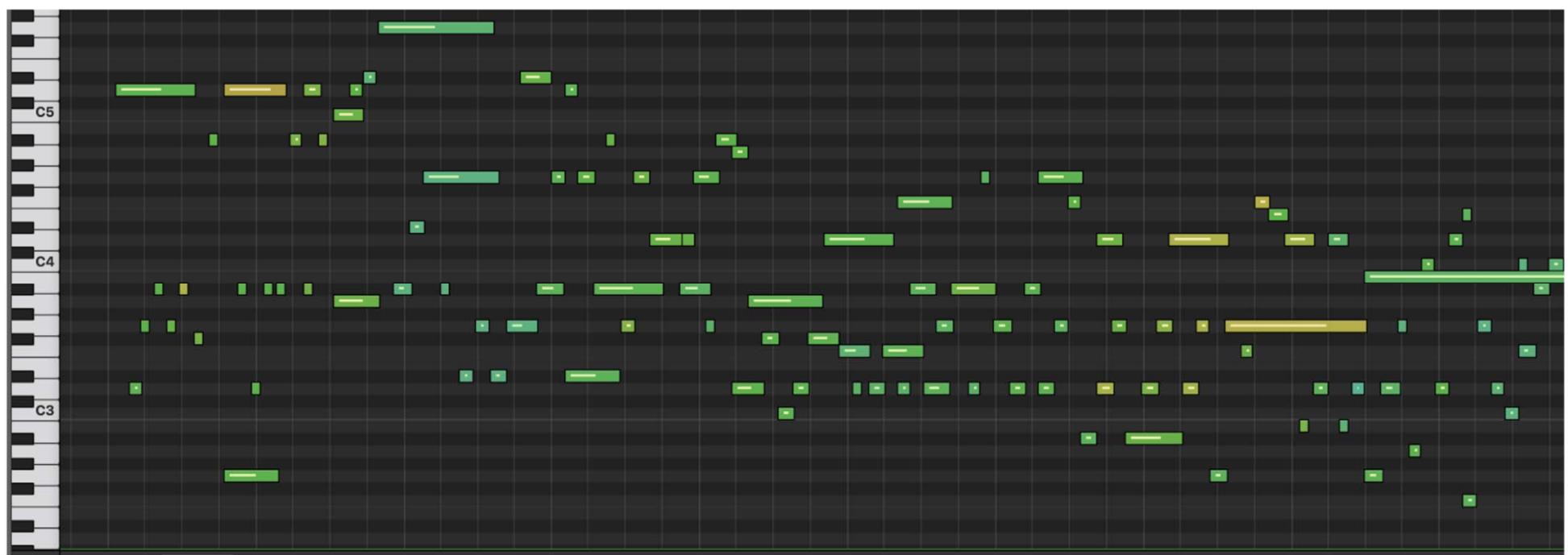
Representing Symbolic Music: Piano Music Score

- How to represent a sequence of piano music score?
- Add one additional token to mark “**time-shift**” (i.e., ΔT ; interval time)
- [pitch] [duration] [time-shift] [pitch] [duration] [time-shift] ...

A piano music score is shown on a staff system. The top staff is in treble clef and the bottom staff is in bass clef, both in 4/4 time with a key signature of one sharp. Two specific notes are highlighted with red circles: note 'a' is a dotted half note in the treble staff, and note 'b' is a quarter note in the bass staff. Below the staff, a red box contains the text $\Delta T = 0$.

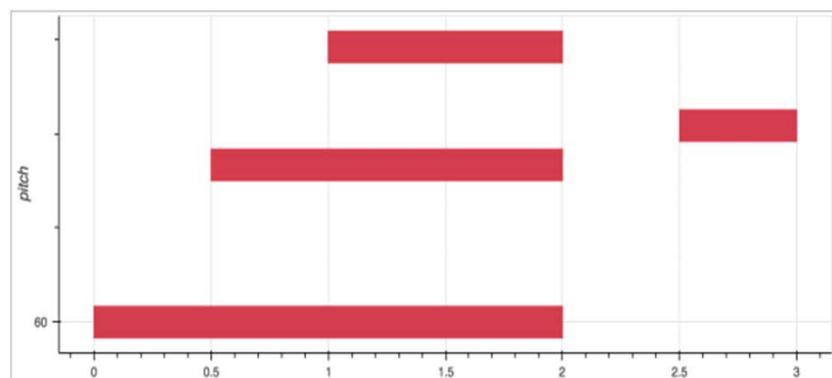
Representing Symbolic Music: Piano Music Performance

- How to represent a sequence of piano music performance?
 - Note **timings** are based on human performance rather than a score
 - Note **velocities** are based on human performance, i.e. with how much force did the performer strike each note



The MIDI-like Representation

- The **MIDI-like representation** for **piano music performance**
 - 128 **note-on** events + 128 **note-off** events: start or release a MIDI note
 - 100 **time-shift** events in increments of 10 ms up to 1 second; move forward in time to the next note event
 - 32 **velocity** events, corresponding to MIDI velocities quantized into 32 bins



SET_VELOCITY<80>, NOTE_ON<60>
TIME_SHIFT<500>, NOTE_ON<64>
TIME_SHIFT<500>, NOTE_ON<67>
TIME_SHIFT<1000>, NOTE_OFF<60>, NOTE_OFF<64>,
NOTE_OFF<67>
TIME_SHIFT<500>, SET_VELOCITY<100>, NOTE_ON<65>
TIME_SHIFT<500>, NOTE_OFF<65>

Ref: Simon & Oore, "Performance RNN: Generating music with expressive timing and dynamics," 2017

Representing Multi-track Music

OpenAI's MuseNet

<https://openai.com/index/musenet/>

v — **velocity** (volume), in [0, 127]
v0 — **offset** of a note
v72 — **onset** of a note
G1 — **pitch**
piano — **instrument**
wait — move the **time** marker

```
bach piano_strings start tempo90 piano:v72:G1
piano:v72:G2 piano:v72:B4 piano:v72:D4 violin:v80:G4
piano:v72:G4 piano:v72:B5 piano:v72:D5 wait:12
piano:v0:B5 wait:5 piano:v72:D5 wait:12 piano:v0:D5
wait:4 piano:v0:G1 piano:v0:G2 piano:v0:B4 piano:v0:D4
violin:v0:G4 piano:v0:G4 wait:1 piano:v72:G5 wait:12
piano:v0:G5 wait:5 piano:v72:D5 wait:12 piano:v0:D5
wait:5 piano:v72:B5 wait:12
```

(v72: onset of the piano note G1)

Representing Multi-track Music

OpenAI's MuseNet

<https://openai.com/index/musenet/>

v — **velocity** (volume), in [0, 127]
v0 — **offset** of a note
v72 — **onset** of a note
G1 — **pitch**
piano — **instrument**
wait — move the **time** marker

```
bach piano_strings start tempo90 piano:v72:G1
piano:v72:G2 piano:v72:B4 piano:v72:D4 violin:v80:G4
piano:v72:G4 piano:v72:B5 piano:v72:D5 wait:12
piano:v0:B5 wait:5 piano:v72:D5 wait:12 piano:v0:D5
wait:4 piano:v0:G1 piano:v0:G2 piano:v0:B4 piano:v0:D4
violin:v0:G4 piano:v0:G4 wait:1 piano:v72:G5 wait:12
piano:v0:G5 wait:5 piano:v72:D5 wait:12 piano:v0:D5
wait:5 piano:v72:B5 wait:12
```

(special “wait” token to indicates the delta time between events)

Representing Multi-track Music

OpenAI's MuseNet

<https://openai.com/index/musenet/>

v — **velocity** (volume), in [0, 127]
v0 — **offset** of a note
v72 — **onset** of a note
G1 — **pitch**
piano — **instrument**
wait — move the **time** marker

```
bach piano_strings start tempo90 (1)
(2) piano:v72:G1 piano:v72:G2 piano:v72:B4 piano:v72:D4 violin:v80:G4
(3) (4) (5)
(6) piano:v72:G4 piano:v72:B5 piano:v72:D5 wait:12 (7) (8)
      piano:v0:B5 wait:5 piano:v72:D5 wait:12 piano:v0:D5
      wait:4 piano:v0:G1 piano:v0:G2 piano:v0:B4 piano:v0:D4
      violin:v0:G4 piano:v0:G4 wait:1 piano:v72:G5 wait:12
      piano:v0:G5 wait:5 piano:v72:D5 wait:12 piano:v0:D5
      wait:5 piano:v72:B5 wait:12
```

(all these eight notes are played at the same time)

Representing Multi-track Music

OpenAI's MuseNet

<https://openai.com/index/musenet/>

v — **velocity** (volume), in [0, 127]
v0 — **offset** of a note
v72 — **onset** of a note
G1 — **pitch**
piano — **instrument**
wait — move the **time** marker

```
bach piano_strings start tempo90 piano:v72:G1
piano:v72:G2 piano:v72:B4 piano:v72:D4 violin:v80:G4
piano:v72:G4 piano:v72:B5 piano:v72:D5 wait:12
piano:v0:B5 wait:5 piano:v72:D5 wait:12 piano:v0:D5
wait:4 piano:v0:G1 piano:v0:G2 piano:v0:B4 piano:v0:D4
violin:v0:G4 piano:v0:G4 wait:1 piano:v72:G5 wait:12
piano:v0:G5 wait:5 piano:v72:D5 wait:12 piano:v0:D5
wait:5 piano:v72:B5 wait:12
```

(duration of the note G1: $12+5+12+4=33$ ms)

Representing Multi-track Music

OpenAI's MuseNet

<https://openai.com/index/musenet/>

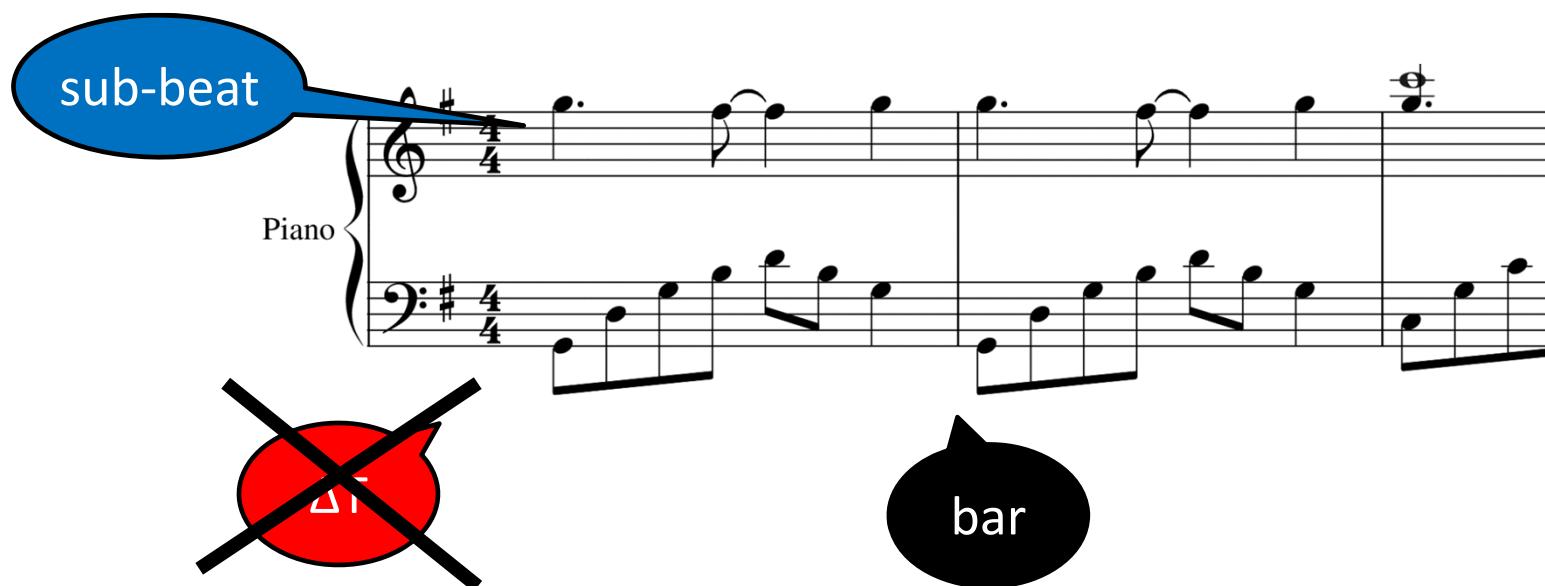
v — **velocity** (volume), in [0, 127]
v0 — **offset** of a note
v72 — **onset** of a note
G1 — **pitch**
piano — **instrument**
wait — move the **time** marker

```
bach piano_strings start tempo90 piano:v72:G1
piano:v72:G2 piano:v72:B4 piano:v72:D4 violin:v80:G4
piano:v72:G4 piano:v72:B5 piano:v72:D5 wait:12
piano:v0:B5 wait:5 piano:v72:D5 wait:12 piano:v0:D5
wait:4 piano:v0:G1 piano:v0:G2 piano:v0:B4 piano:v0:D4
violin:v0:G4 piano:v0:G4 wait:1 piano:v72:G5 wait:12
piano:v0:G5 wait:5 piano:v72:D5 wait:12 piano:v0:D5
wait:5 piano:v72:B5 wait:12
```

No bar lines!

REMI: Revamped MIDI Representation

- **REMI**: An alternative token representation of music
 - Mark the bar lines by “**bar**” tokens
 - Indicate the position of a note within a **bar** by “**sub-beat**” tokens (after timing quantization)



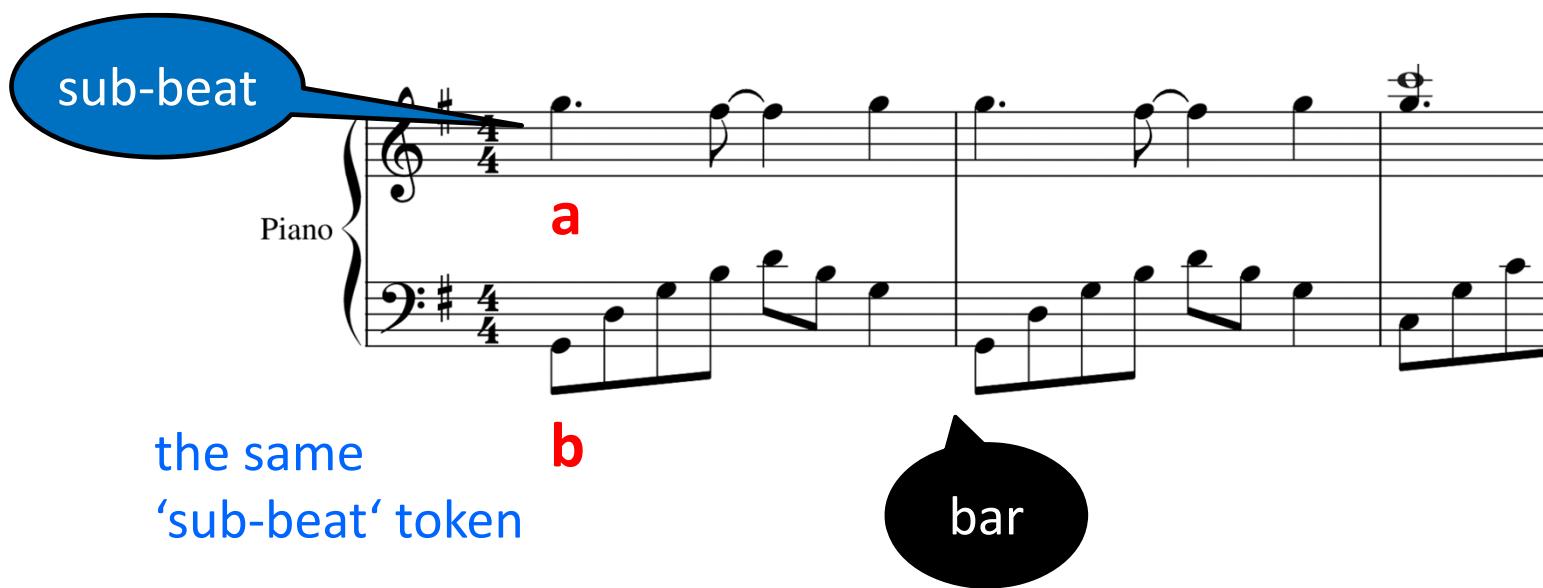
16th Notes

(images from the internet)



REMI: Revamped MIDI Representation

- **REMI**: An alternative token representation of music
 - Mark the bar lines by “**bar**” tokens
 - Indicate the position of a note within a **bar** by “**sub-beat**” tokens (after timing quantization)



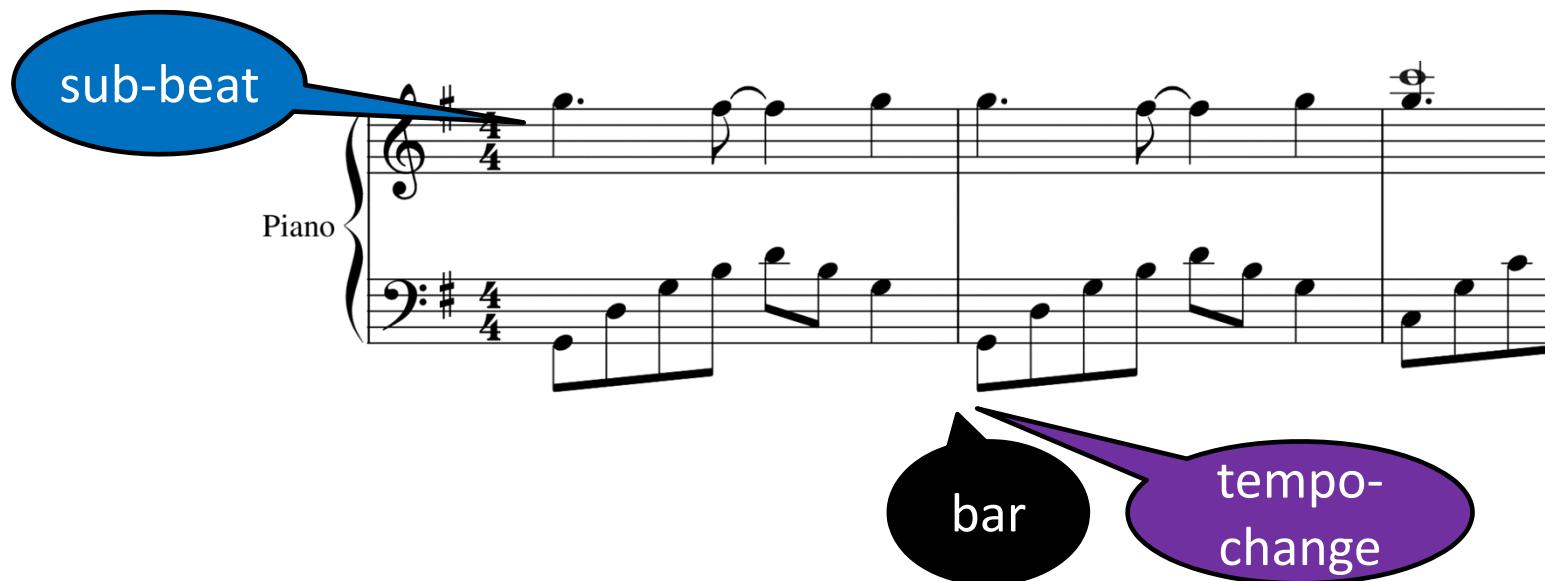
16th Notes

(images from the internet)



REMI: Revamped MIDI Representation

- REMI: An alternative token representation of music
 - Add **bar-wise tempo-change** tokens when needed to account for **tempo variations within a song** (the “tempo-change” tokens are after “bar” tokens)



REMI: Revamped MIDI Representation

- REMI
 - Use `Sub-beat, Bar, and Tempo` instead of **Time-Shift**
 - Use `Note-Duration` instead of **Note-Off**
 - note-on and note-off are native MIDI events
 - but using note-duration seems to have better musical meaning
 - Add `Chord` tokens

	MIDI-like	REMI
Note onset	NOTE-ON (0-127)	NOTE-ON (0-127)
Note offset	NOTE-OFF (0-127)	NOTE DURATION (32th note multiples)
Note velocity	NOTE VELOCITY (32 bins)	NOTE VELOCITY (32 bins)
Time grid	TIME-SHIFT (10-1000ms)	POSITION (16 bins) & BAR (1)
Tempo changes	x	TEMPO (30-209 BPM)
Chord	x	CHORD (60 types)

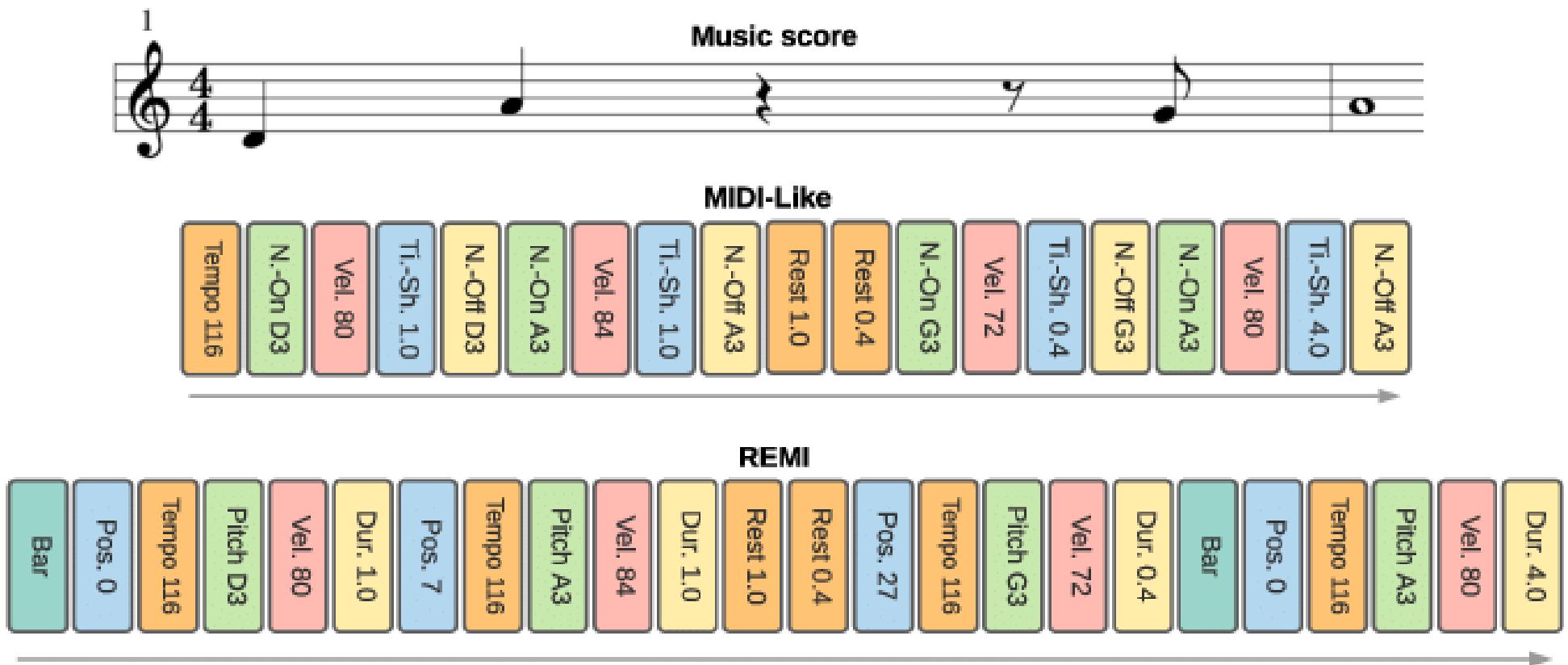
Table 1: A comparison of the commonly-used MIDI-like event representation with the proposed one, REMI. In the brackets, we show the corresponding ranges.

Ref: Huang et al, “Pop Music Transformer: Beat-based modeling and generation of expressive Pop piano compositions”, ACM Multimedia 2020

MIDI-like vs. REMI

- REMI
 - Mark the bar lines **bar** tokens and divide a bar into 16 discrete **sub-beats**
 - **Symbolic timing + explicit time grid**
 - More timing quantization
 - Good for **MIDI scores**
 - Good for MIDI performances with **steady beats** like the **Pop music**
 - Good for **multi-track music**
 - The bar tokens make it easy to synchronize
- MIDI-like
 - Use **time-shift** tokens
 - **Physical timing & relative time intervals**
 - **Less timing quantitation**
 - Good for expressive MIDI performances such as the **Classical music**

MIDI-like vs REMI



Ref: Fradet et al, "MidiTok: A Python package for MIDI file tokenization", ISMIR LBD 2021

piano

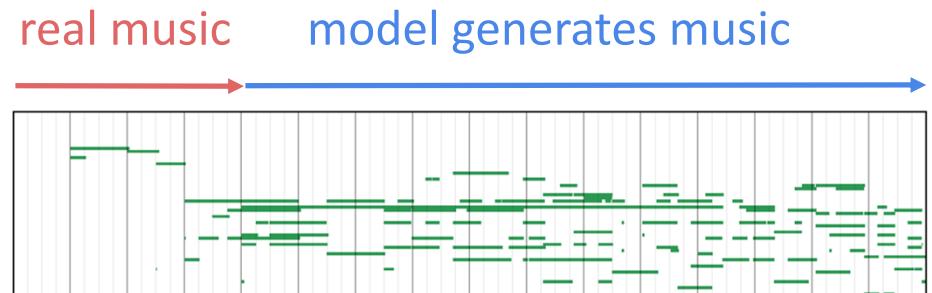


Pop Music Transformer [huang20mm]

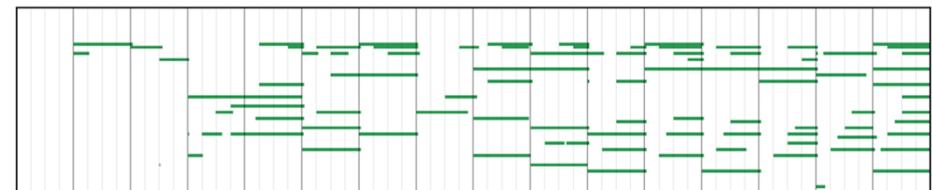
<https://github.com/YatingMusic/remi>

- Adopt the **REMI** representation
- Outperforms the Music Transformer (“MIDI-like”) for generating pop piano performances

Music
Transformer



Pop Music
Transformer



Ref: Huang et al, “Pop Music Transformer: Beat-based modeling and generation of expressive Pop piano compositions”, ACM Multimedia 2020

Pop Music Transformer [huang20mm]

Pop Music Transformer ("REMI") outscores Music Transformer ("Baselines 1 & 3") in a user study that involves 76 raters

Pairs		Wins	Losses	<i>p</i> -value
REMI	Baseline 1	103	49	5.623e-5
REMI	Baseline 3	92	60	0.0187
Baseline 3	Baseline 1	90	62	0.0440

Table 5: The result of pairwise comparison from the user study, with the *p*-value of the Wilcoxon signed-rank test.

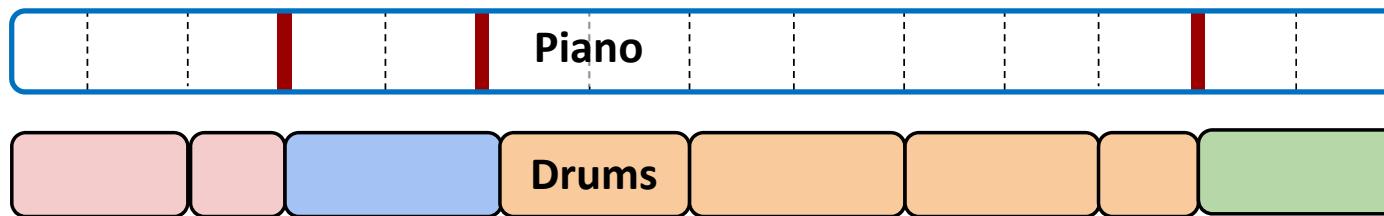
Method	Note offset	Time grid	TEMPO	CHORD	Beat STD	Downbeat STD	Downbeat salience
Baseline 1	NOTE-OFF	TIME-SHIFT (10-1000ms)			0.0968	0.3561	0.1033
Baseline 2	DURATION	TIME-SHIFT (10-1000ms)			0.0394	0.1372	0.1651
Baseline 3	DURATION	TIME-SHIFT (16th-note multiples)			0.0396	0.1383	0.1702
REMI	DURATION	POSITION & BAR	✓	✓	0.0386	0.1376	0.2279
	DURATION	POSITION & BAR	✓		0.0363	0.1265	0.1936
	DURATION	POSITION & BAR		✓	0.0292	0.0932	0.1742
	DURATION	POSITION & BAR			0.0199	0.0595	0.1880
Real data					0.0607	0.2163	0.2055

piano + drum guitar

Pop Music Transformer



- **Easier to add drums** (via structure analysis and grooving analysis)
 - https://soundcloud.com/yating_ai/sets/ai-piano-generation-demo-202004
 - https://soundcloud.com/yating_ai/sets/ai-pianodrum-generation-demo-202004



(Figure made by Wen-Yi Hsiao)

- Extensions can also generate **guitar tabs** [chen20ismir]
 - https://soundcloud.com/yating_ai/ai-guitar-tab-generation-202003/s-KHozfWOPTv5

Ref 1: Huang et al, “Pop Music Transformer: Beat-based modeling and generation of expressive Pop piano compositions”, ACM Multimedia 2020

Ref 2: Chen et al, “Automatic composition of guitar tabs by Transformers and groove modeling”, ISMIR 2020

Text vs Symbolic Music Tokens

- **Text tokenizers**
 - **Word**-based tokenizers
 - Large vocabulary size
 - OOV issue
 - **Character**-based tokenizers
 - Small vocabulary size (**tens**)
 - No OOV
 - Long token sequence
 - **Sub-word** tokenizers (e.g., byte-pair encoding)
 - Middle ground
- **Symbolic music tokenizers**
 - Need multiple tokens to describe the *various aspects* of a musical note
 - Need other auxiliary tokens to denote *time* or *musical structure*
 - It's like *character-level* tokenizers
 - Small vocabulary size (**hundreds**)
 - No OOV
 - Long token sequence
 - Can also do sub-word tokenization
 - Ref: Fradet et al, "Byte pair encoding for symbolic music," EMNLP 2023.

Library: **MidiTok**

<https://miditok.readthedocs.io/>

- “MidiTok can take care of converting (tokenizing) your symbolic music files (**MIDI, abc**) into **tokens**, ready to be fed to models such as **Transformer**, for any generation, transcription or MIR task”
 - “MidiTok features most known MIDI tokenizations (e.g. REMI, Compound Word...), and is built around the idea that they all share common parameters and methods”
 - “It supports Byte Pair Encoding (BPE) and data augmentation.”