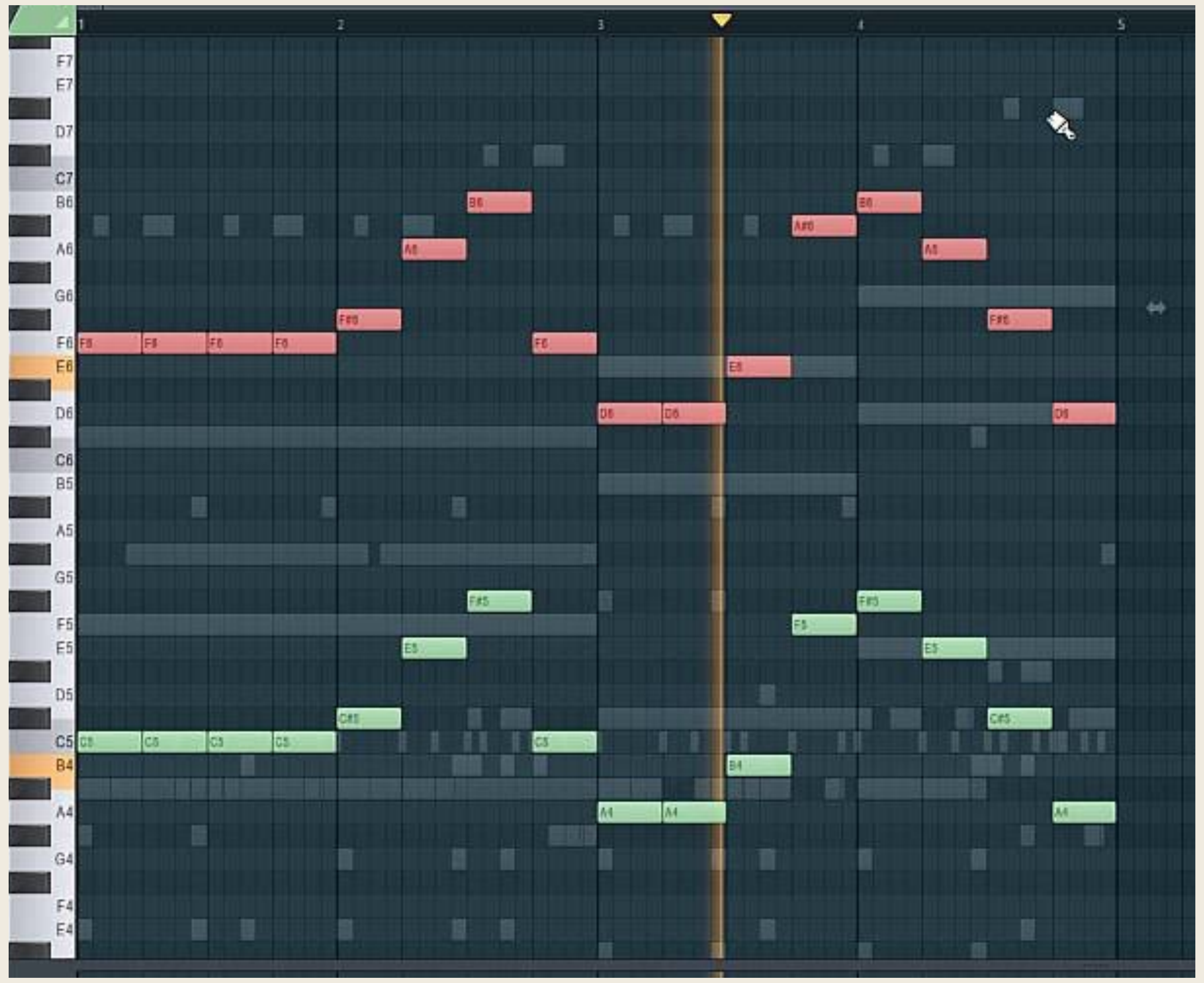
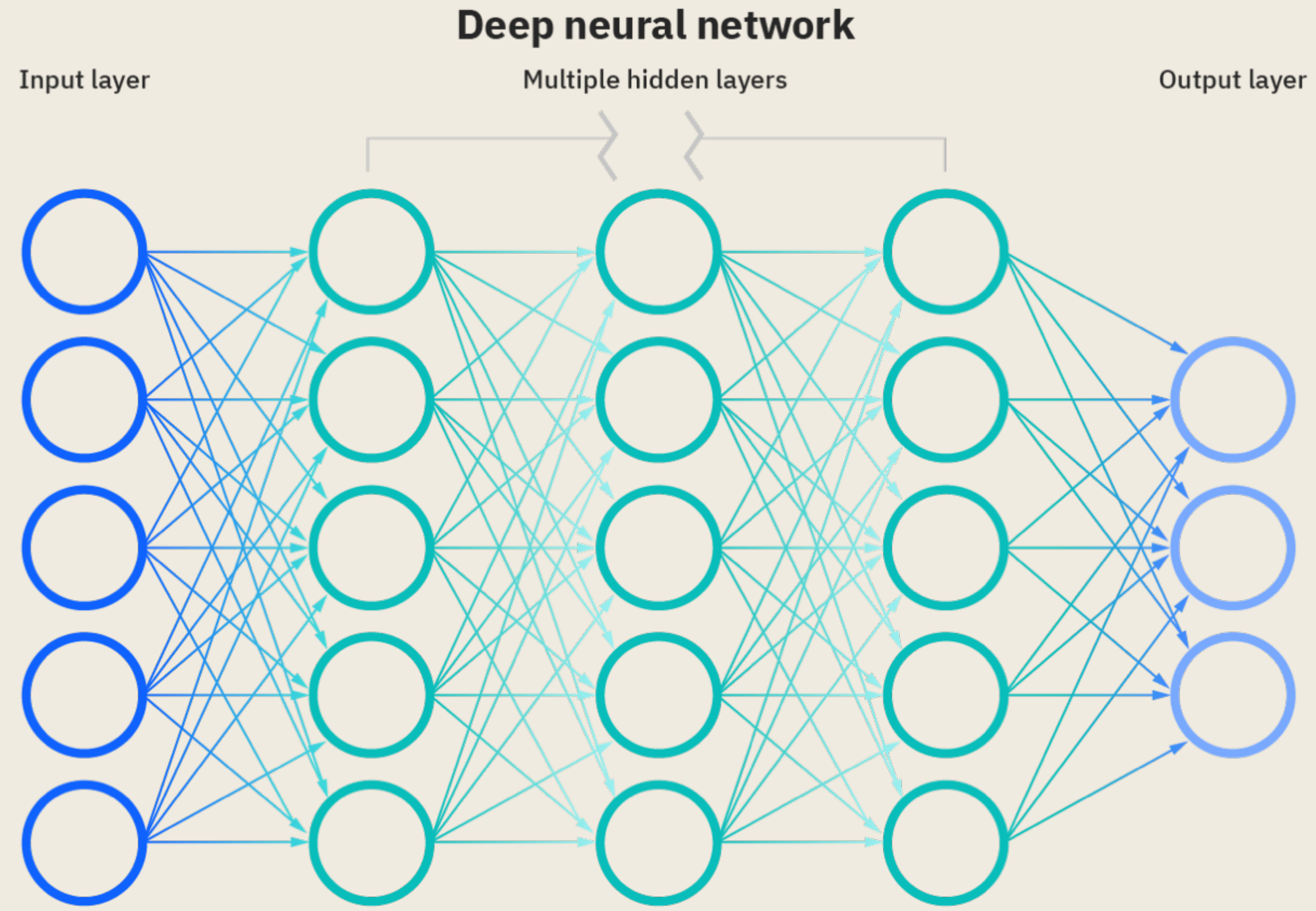
Deep Learning for Symbolic Music Modeling

Séminaire Musique & IA - Université d'Angers

Background

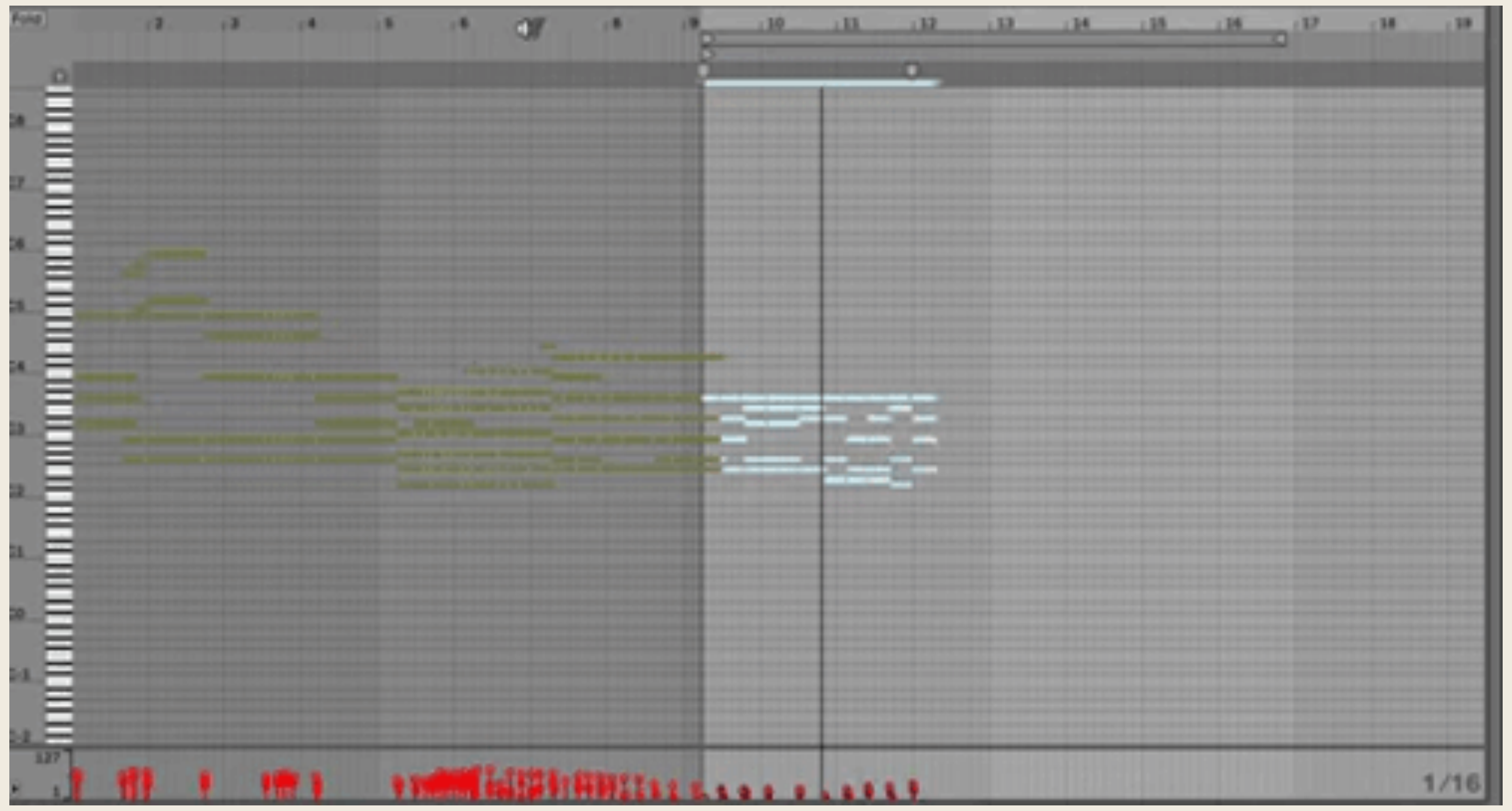
Symbolic music





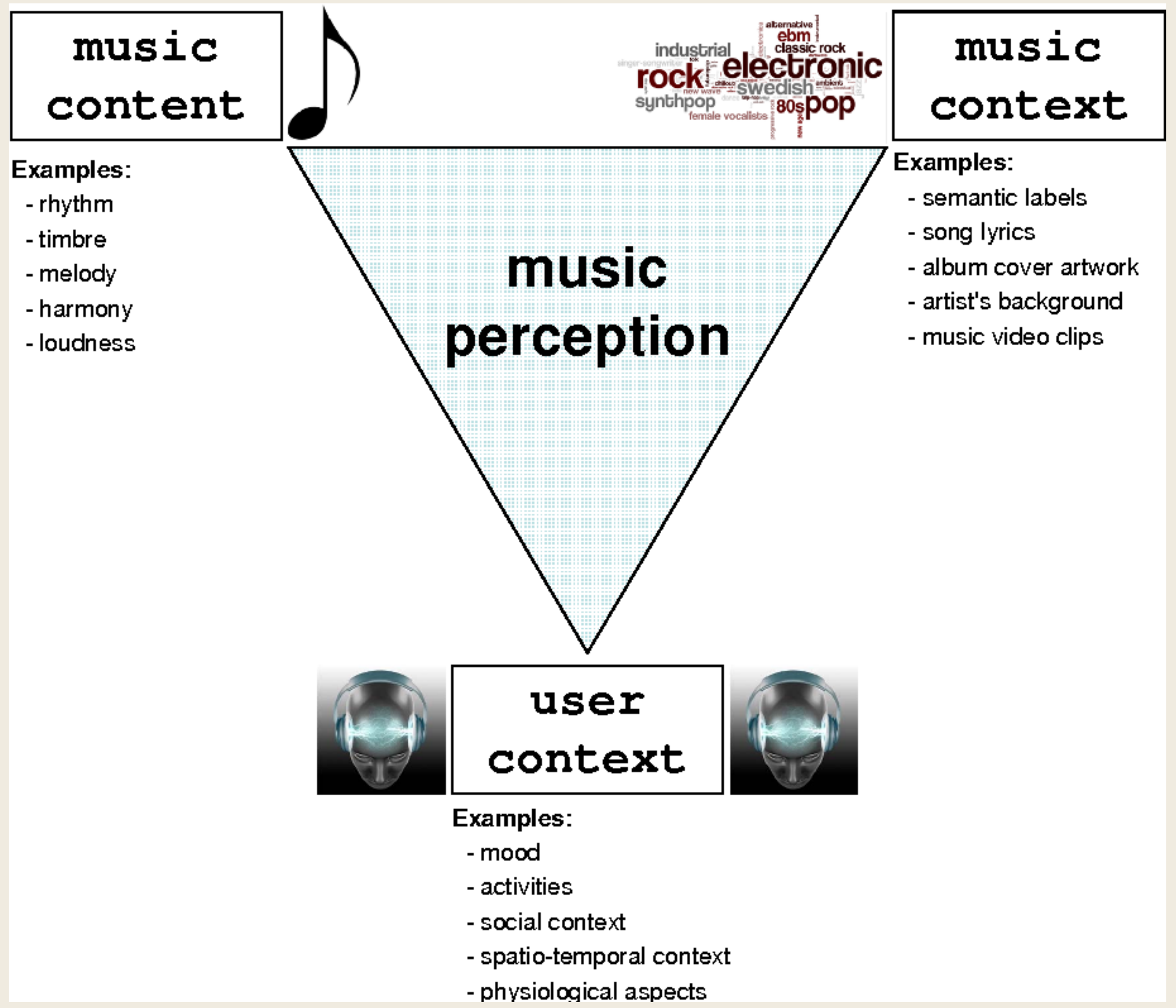
Tasks

MUSIC GENERATION



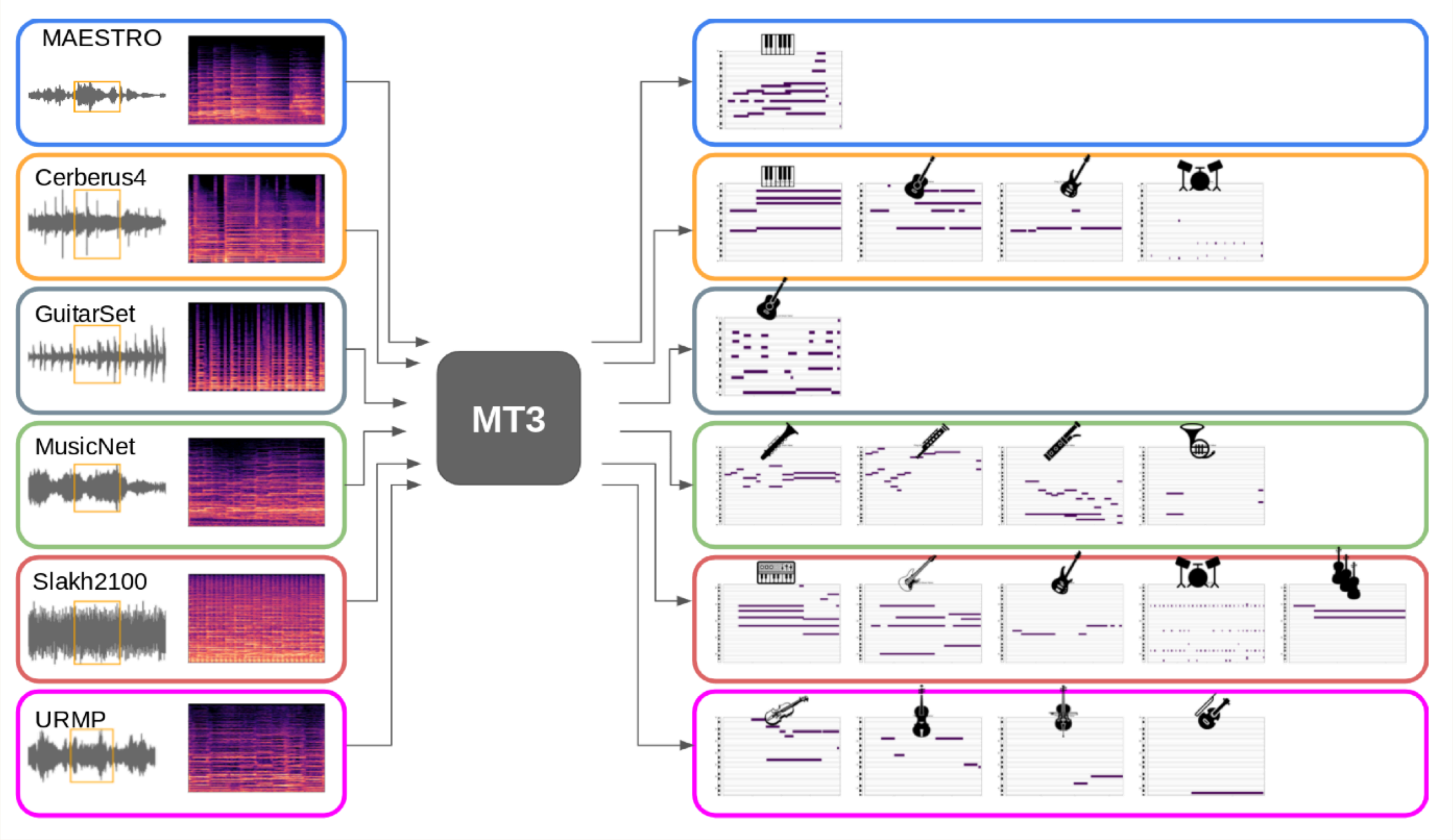
Piano Inpainting Application (Hadjeres et al.)

MUSIC INFORMATION RETRIEVAL (MIR)



(Schedl et al.)

MUSIC TRANSCRIPTION



MT3 (Gardner et al.)

File formats

- .abc
- MusicXML
- MIDI
  - Tracks and instruments
  - Tempos, time Signatures
  - Effects (sustain pedal, pitch bend ...)

```
X:1
T:Speed the Plough
M:4/4
C:Trad.
K:G
|:GABc dedB|dedB dedB|c2ec B2dB|c2A2 A2BA|
  GABc dedB|dedB dedB|c2ec B2dB|A2F2 G4:|
|:g2gf gdBd|g2f2 e2d2|c2ec B2dB|c2A2 A2df|
  g2gf g2Bd|g2f2 e2d2|c2ec B2dB|A2F2 G4:|
```

Abc notation

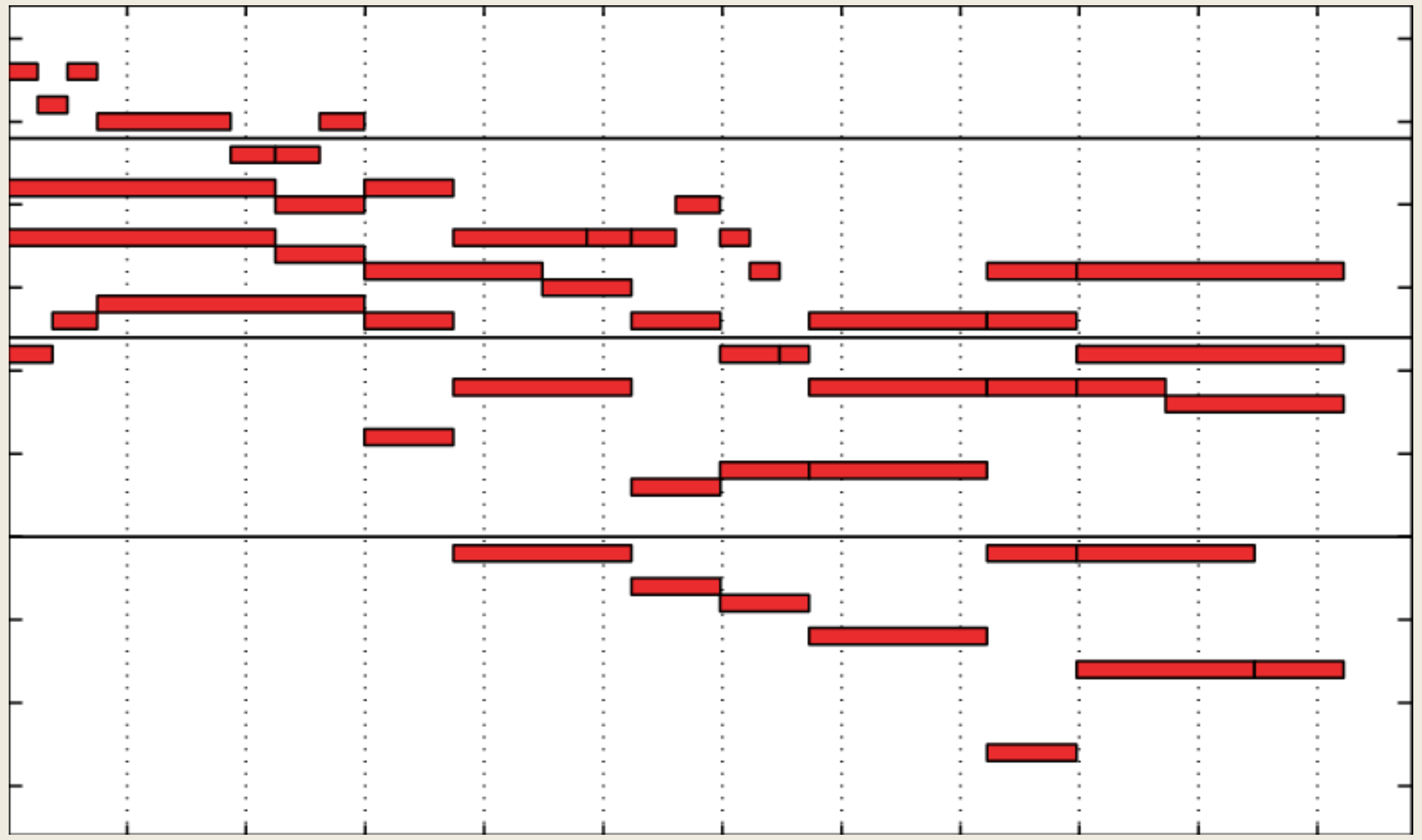
```
<note>
  <pitch>
    <step>E</step>
    <alter>-1</alter>
    <octave>4</octave>
  </pitch>
  <duration>2</duration>
  <type>half</type>
</note>
```

MusicXML

	STATUS		DATA 1 (if needed)	DATA 2 (if needed)
	1 t t t n n n n		0 x x x x x x x	0 x x x x x x x
HEX	Type of Message	Channel # 1-16 (Values 0-15)	Data Value 1 (0-127)	Data Value 2 (0-127)
0x8n	1 0 0 0	Note OFF		
0x9n	1 0 0 1	Note ON		
0xAn	1 0 1 0	Polyphonic Aftertouch		
0xBn	1 0 1 1	Control Change (CC)	CHANNEL VOICE MESSAGES (CC Controllers 120-127 reserved for CHANNEL MODE MESSAGES)	
0xCn	1 1 0 0	Program Change		
0xDn	1 1 0 1	Channel Aftertouch		
0xEn	1 1 1 0	Pitch Wheel		
0xFn	1 1 1 1	SYSTEM MESSAGE	Common, Real-Time, Exclusive	

Music as pianorolls

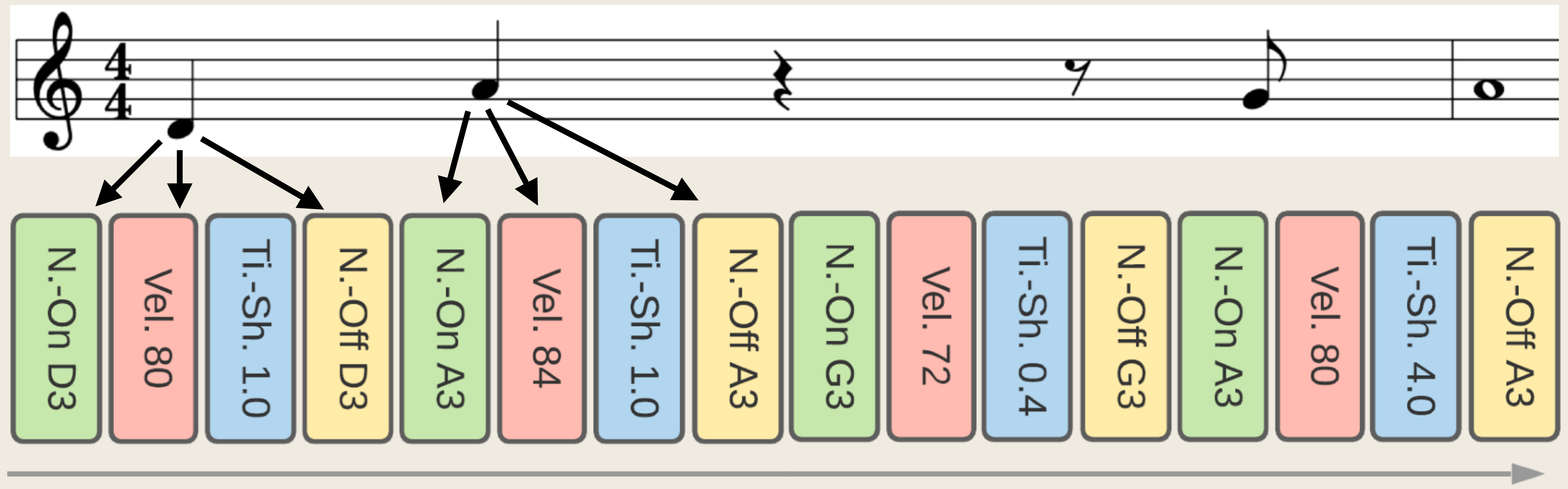
- Matrix with time and pitch dimensions / axis
- Used as an image with continuous models (CNN)
  - MuseGan (Dong et al.)
  - Coconet (Huang et al.)
- Arguably limited in terms of information represented
- And in results: continuous models doesn't perform well with discrete modalities



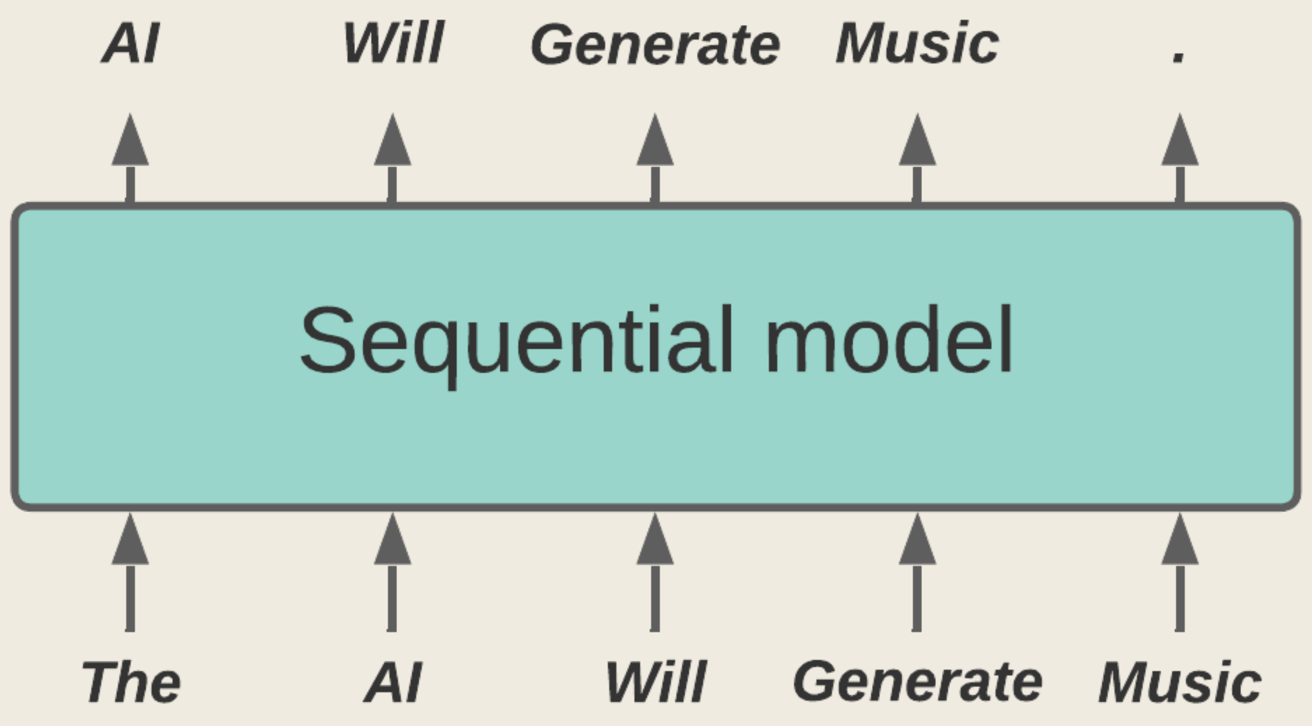
Pianoroll representation

Music as sequence of tokens

- The note attributes and time are serialized into tokens
  - Notes: pitch, velocity, duration or NoteOff
  - Time: TimeShift or Bar and Position
  - Additional information : Tempo, Time Signature…
- The set of all known tokens is called the **vocabulary**
- Used with discrete sequential models (RNN, Transformers)
  - Music Transformer (Anna Huang et al.)
  - Pop Music Transformer (Yu-Siang Huang et al.)
  - Figaro (Von Rütte et al.)

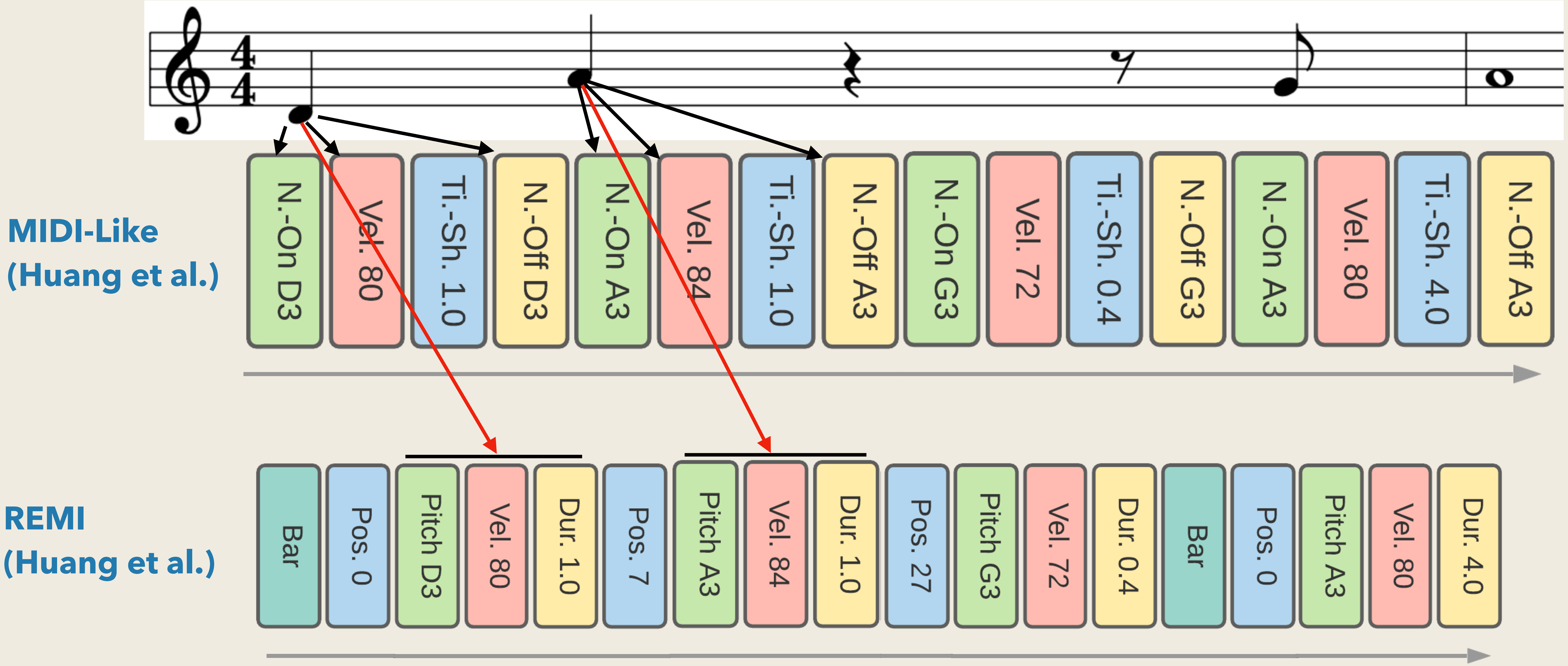


Sheet music and its « MIDI-Like » token sequence equivalent



How to tokenize Music

Several ways to tokenize music



- Unlike text, many ways —> more freedom but implies to make choices

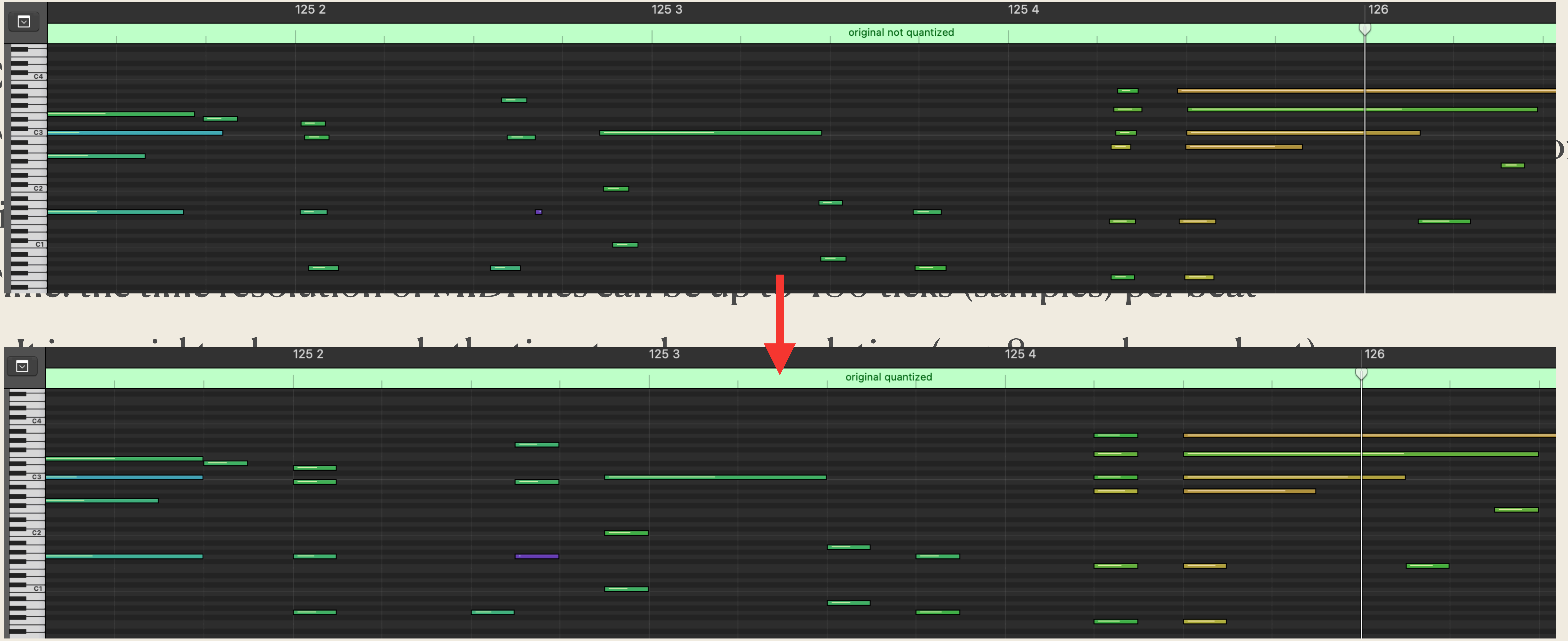
Decomposing music tokenization

- **Time:** using TimeShift to indicate time movements or Bar/Position to indicate new bars and the current time within the current one;
- **Note duration:** using explicit Duration tokens, or NoteOff tokens indicating when notes end;
- **Pitch:** using explicit Pitch tokens, or representing pitch intervals between consecutive notes;
- **Multitrack:** how to represent multiple instruments / tracks simultaneously;
- **Additional information:** tempo, time signature, effects…;
- **Downsampling:** how the information is « downsampled »;
- **Sequence compression:** any way to reduce the sequence length.

A zoom on downsampling

- Corresponds to the « level of detail » to represent the information
- Time, velocity, effects are « semi-continuous » in MIDI files —> we need to **discretize** the information
- Pitch, Velocity (128 possible values) —> can be downsampled to a reduced number of values
- Time: the time resolution of MIDI files can be up to 480 ticks (samples) per beat
  - It is crucial to downsample the time to a lower resolution (e.g. 8 samples per beat)

A zoom on downsampling



MidiTok

- Open source Python package to tokenize symbolic music
- Implements the most popular tokenizations, under a **unified API / workflow**
- Offers great flexibility over downsampling, additional tokens, BPE…
- Can be used with any model, for any task
- Introduced at ISMIR 2021, has since become established
- **GitHub:** github.com/Natooz/MidiTok
- **Documentation:** miditok.readthedocs.io
- **Installation:** *pip install miditok*

```
from miditok import REMI, TokenizerConfig
from miditoolkit import MidiFile

# Creating a multitrack tokenizer
config = TokenizerConfig(
    nb_velocities=16,
    use_chords=True,
    use_programs=True)
tokenizer = REMI(config)

# Loads a midi, converts to tokens, and back to a MIDI
midi = MidiFile('path/to/your_midi.mid')
# automatically detects MIDIs, paths and tokens
tokens = tokenizer(midi)
converted_back_midi = tokenizer(tokens)
```

Different tokenizations yield different results

Focus on time and note duration

	Duration	NoteOff
TimeShift	Pitch D3, Vel. 22, Dur. 7, Ti.-Sh. 7, Pitch A3, Vel. 24, Dur. 7	N.-On D3, Vel. 22, Ti.-Sh. 7, N.-Off D3, N.-On A3, Vel. 24, Ti.-Sh. 7, N.-Off A3
Bar + Position	Bar, Pos. 0, Pitch D3, Vel. 22, Dur. 7, Pos. 7, Pitch A3, Vel. 24, Dur. 7	Bar, Pos. 0, N.-On D3, Vel. 22, Pos. 7, N.-Off D3, N.-On A3, Vel. 24, Pos. 15, N.-Off A3

Generation: distribution of note features

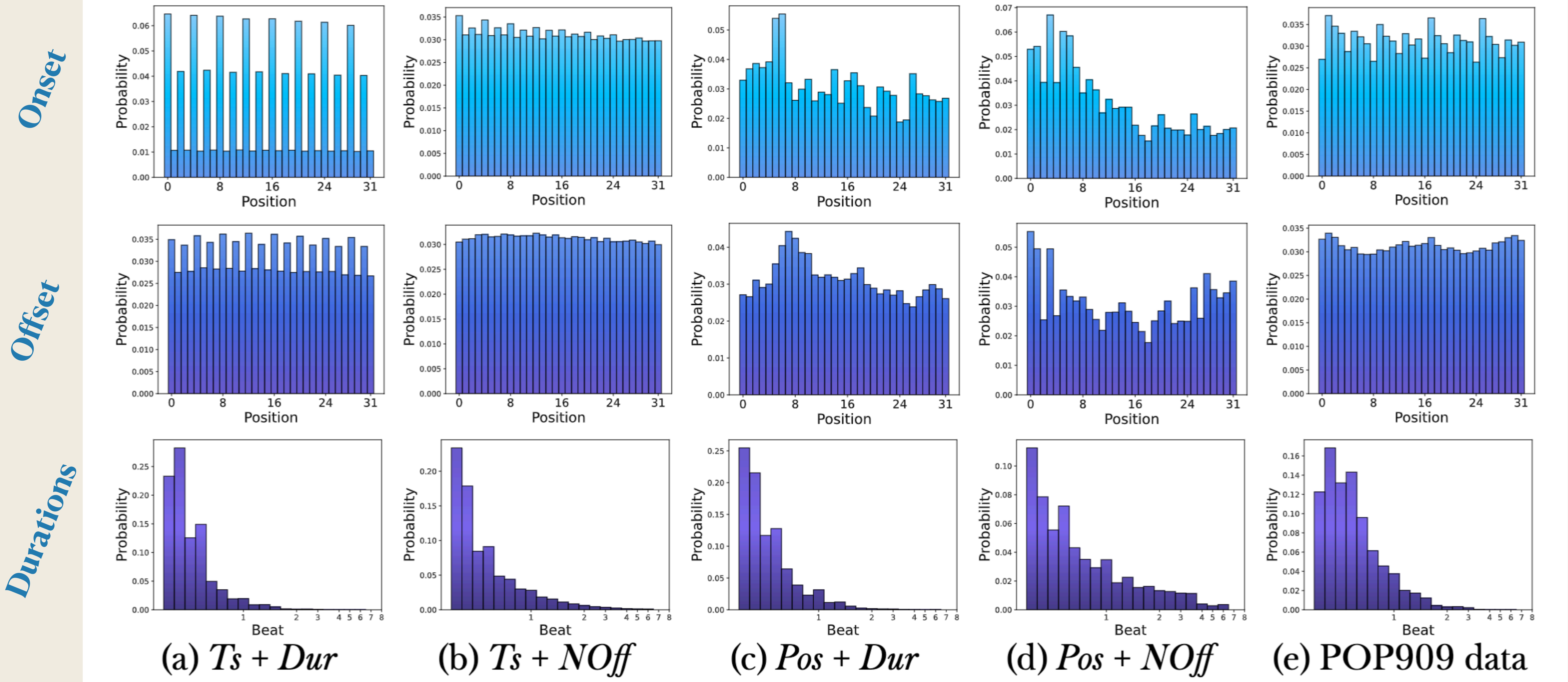
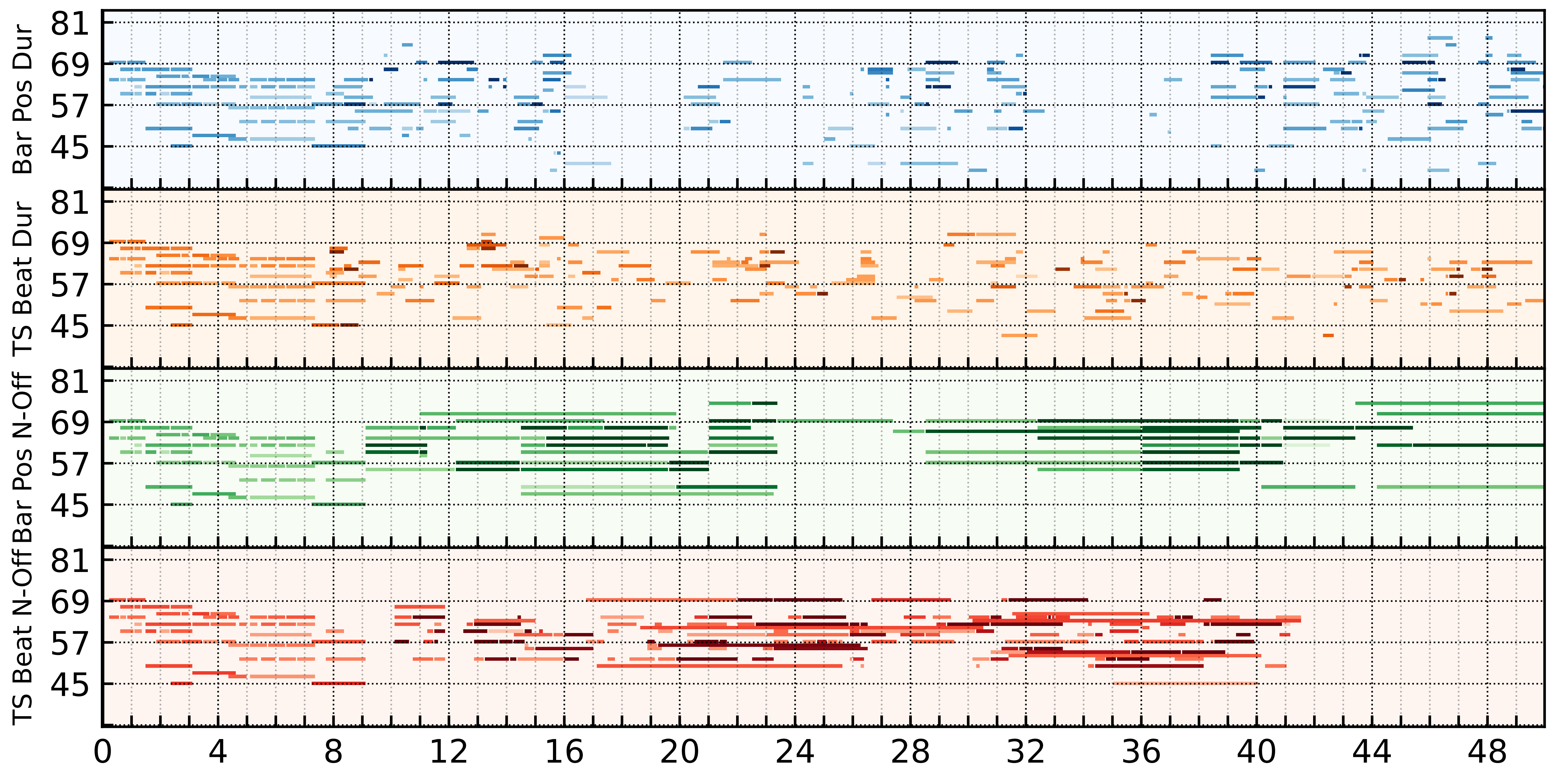


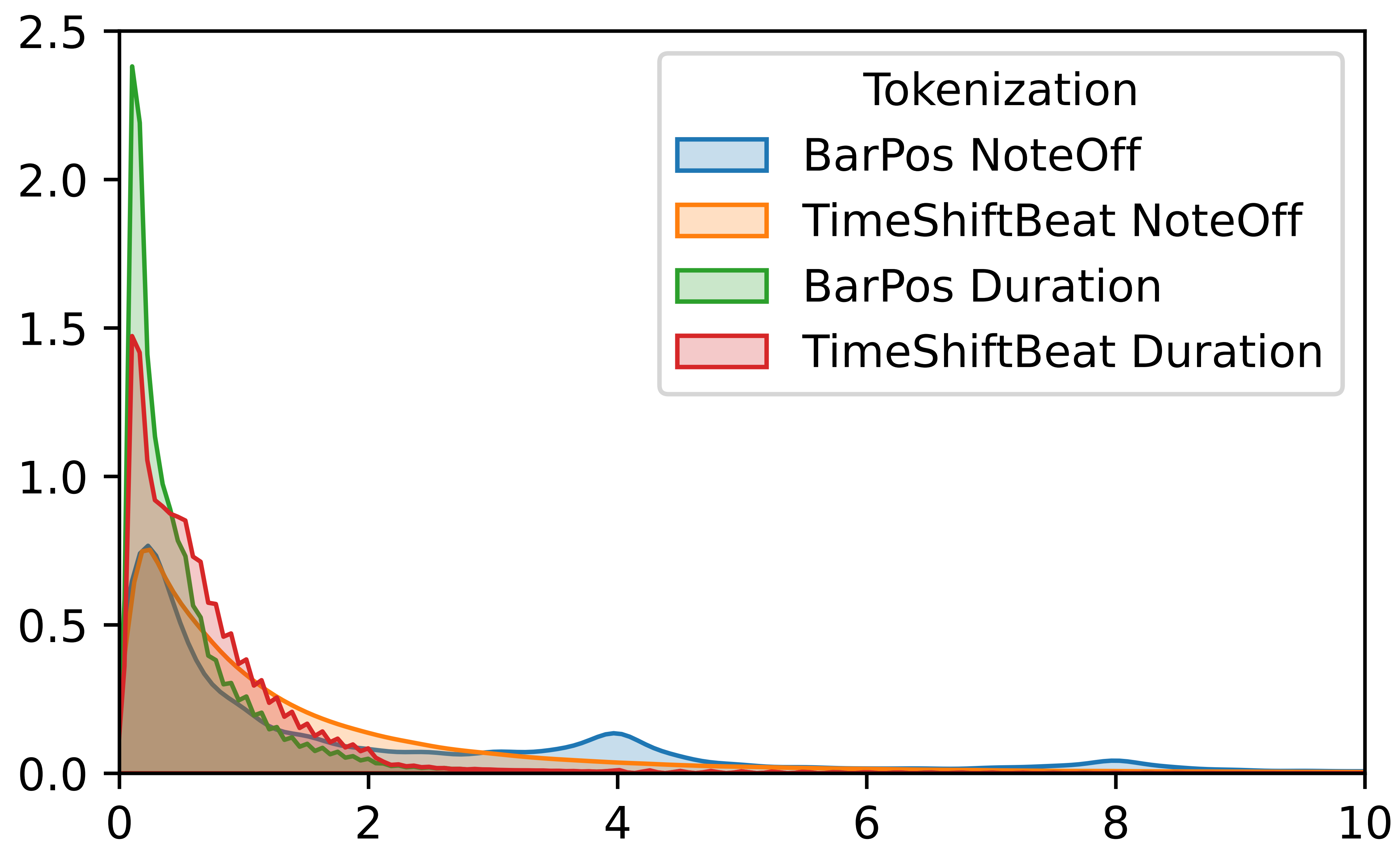
Figure 5.3: Histograms of the note onset positions within bars (top-row), note offset positions within bars (middle-row) and note durations (bottom-row) of the generated notes. There are 32 possible positions within a bar, numerated from 0 (beginning of bar) to 31 (last 32^{th} note). The durations are expressed in beats, ranging from a 32^{th} note to 8

- TimeShift + Duration —> even onset positions
- In all cases we see a decreasing density of high positions, which is accented with Bar / Position
- NoteOff —> Longer durations
  - Because the models can « forget » the notes previously being played

Unended notes with NoteOff tokens



Continuation of the same prompt with the four strategies during training

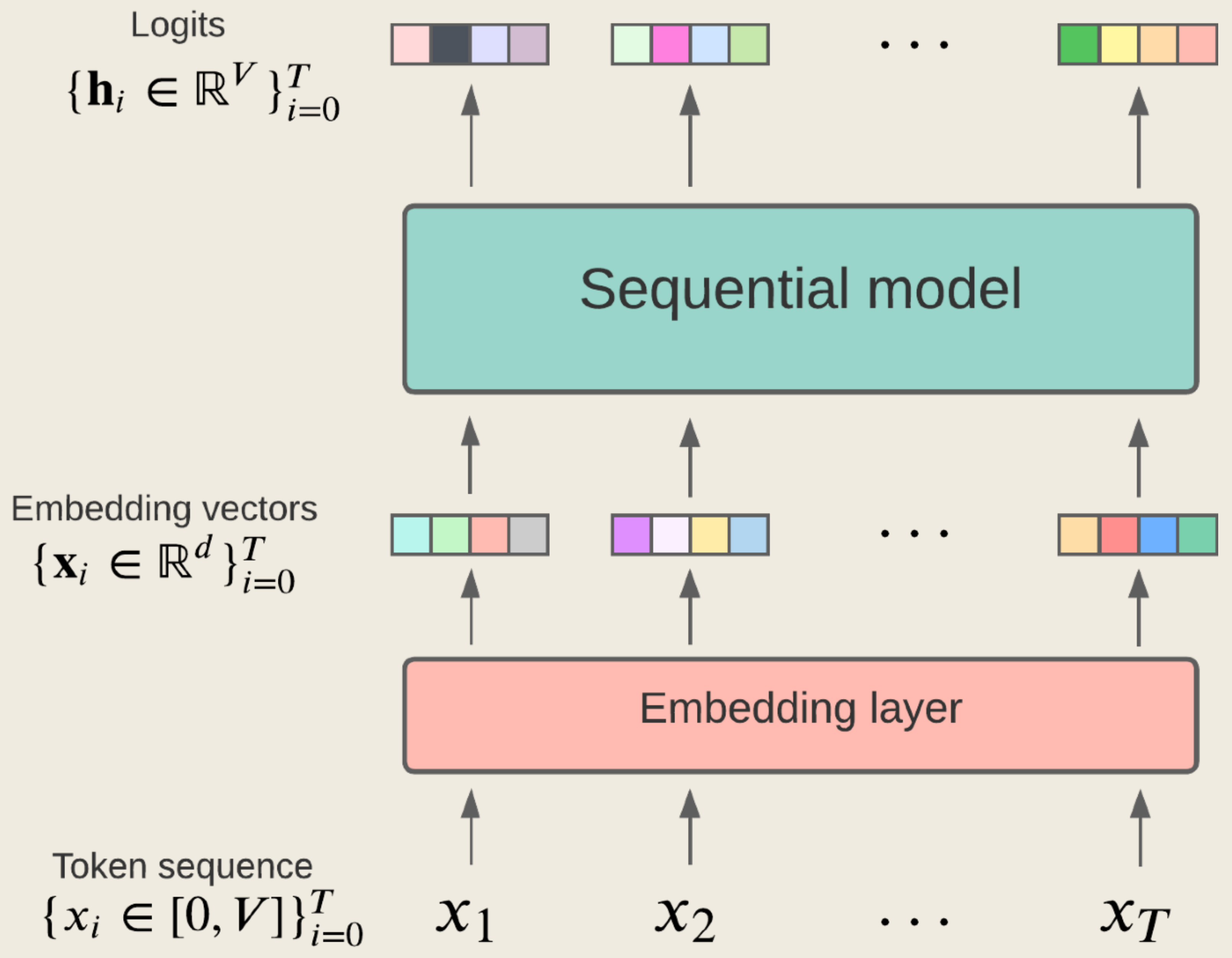


Distribution (density) of note durations in beat

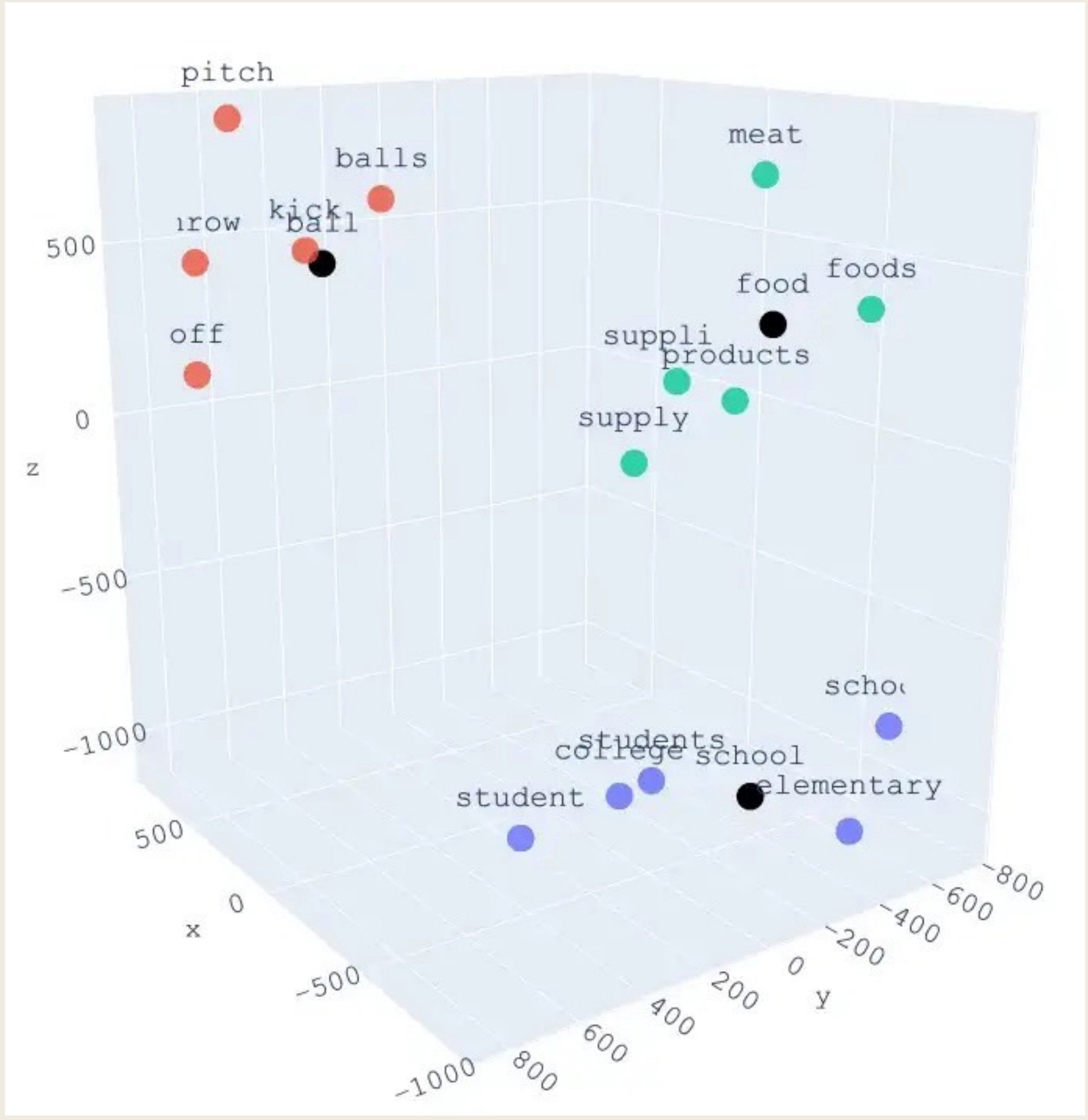
Byte Pair Encoding : improving model efficiency and results

The problem of unused embedding space

- Embedding vectors are contextually learned by the model to represent the information carried by the tokens
- Embedding of size d (512 to 2048)
- In music, vocabularies contain often below 500 tokens
  - More than one dimension per tok.
  - Sub-optimal use of space
- In NLP vocabulary sizes are between 30k and 50k tokens

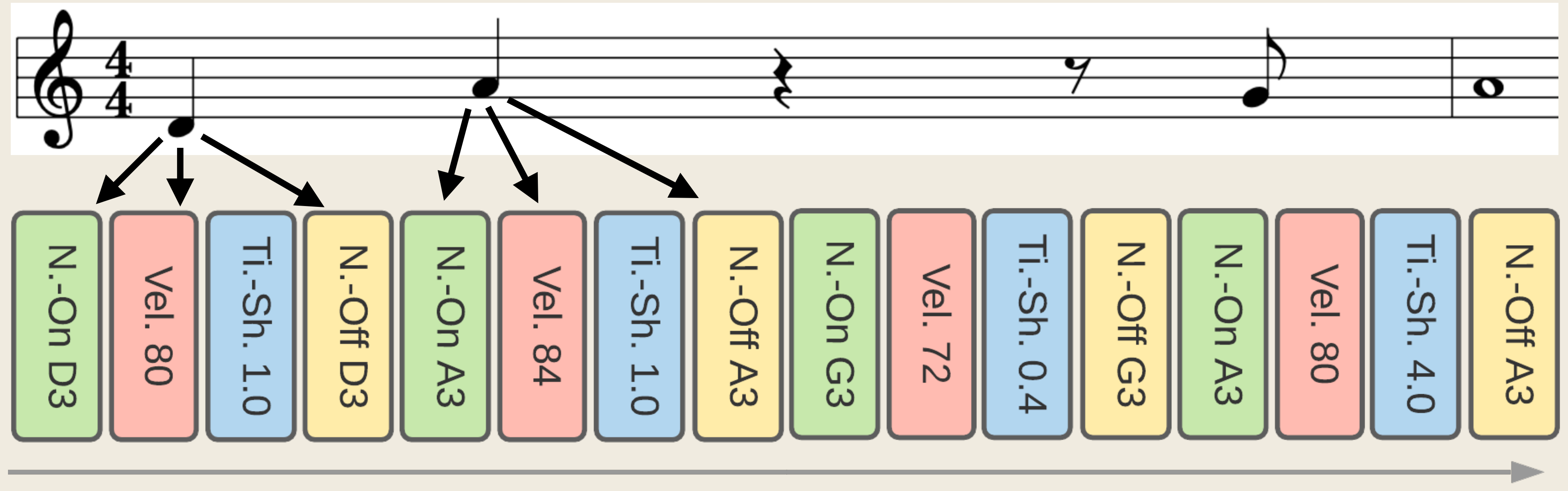


A T-SNE [1] representation of learned word embeddings

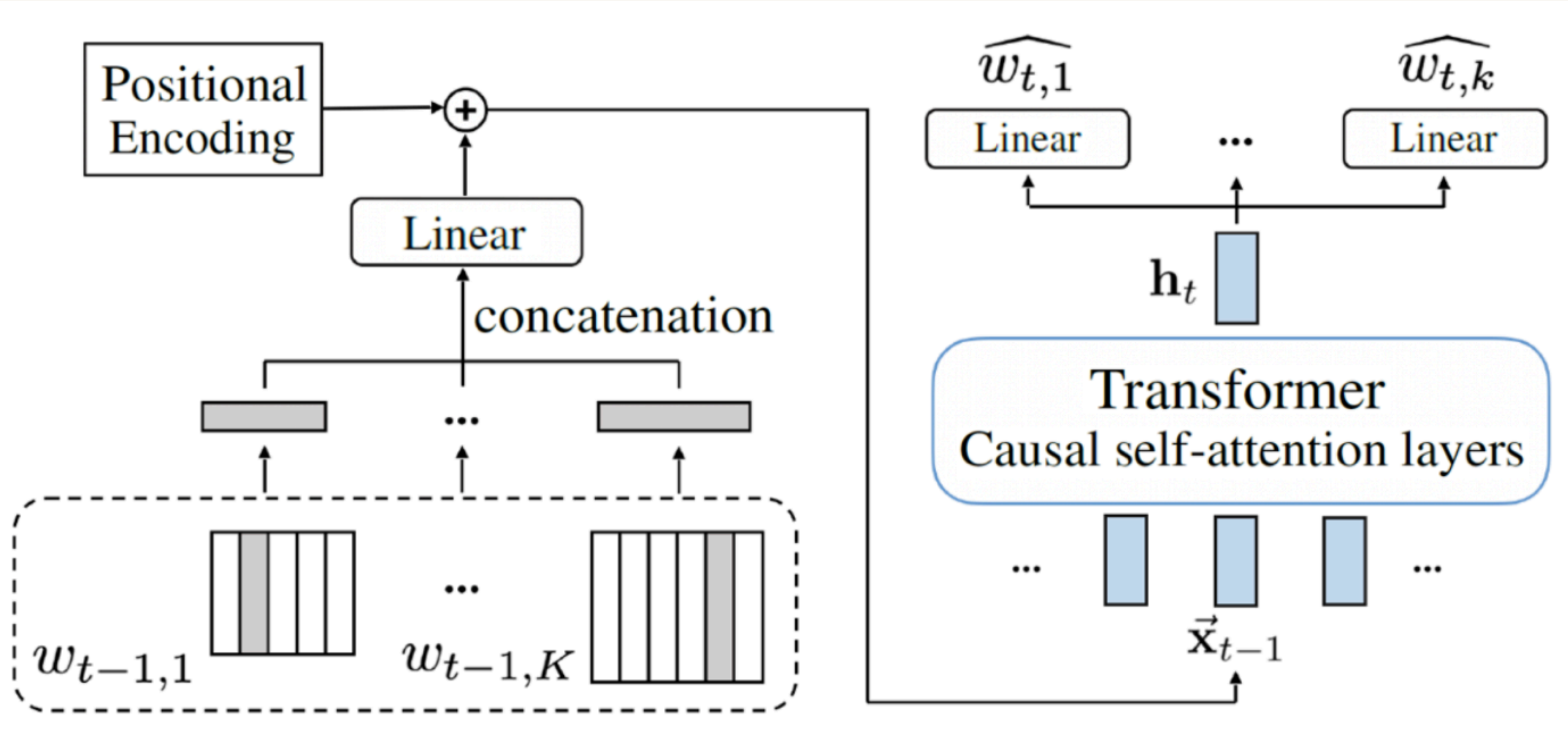


The problem of sequence length

- Music is tokenized into large sequence lengths
  - 2 or 3 tokens per note
- The complexity of Transformer models grows quadratically with the input sequence length
  - The « scope » of music is short
  - And / or model efficiency is bad
- The problem has been tackled by methods merging embeddings
  - Limited / constraining in practice



Sheet music and its « MIDI-Like » token sequence equivalent



Architecture of Compound Word Transformer (Hsiao et al.)

Byte Pair Encoding (BPE)

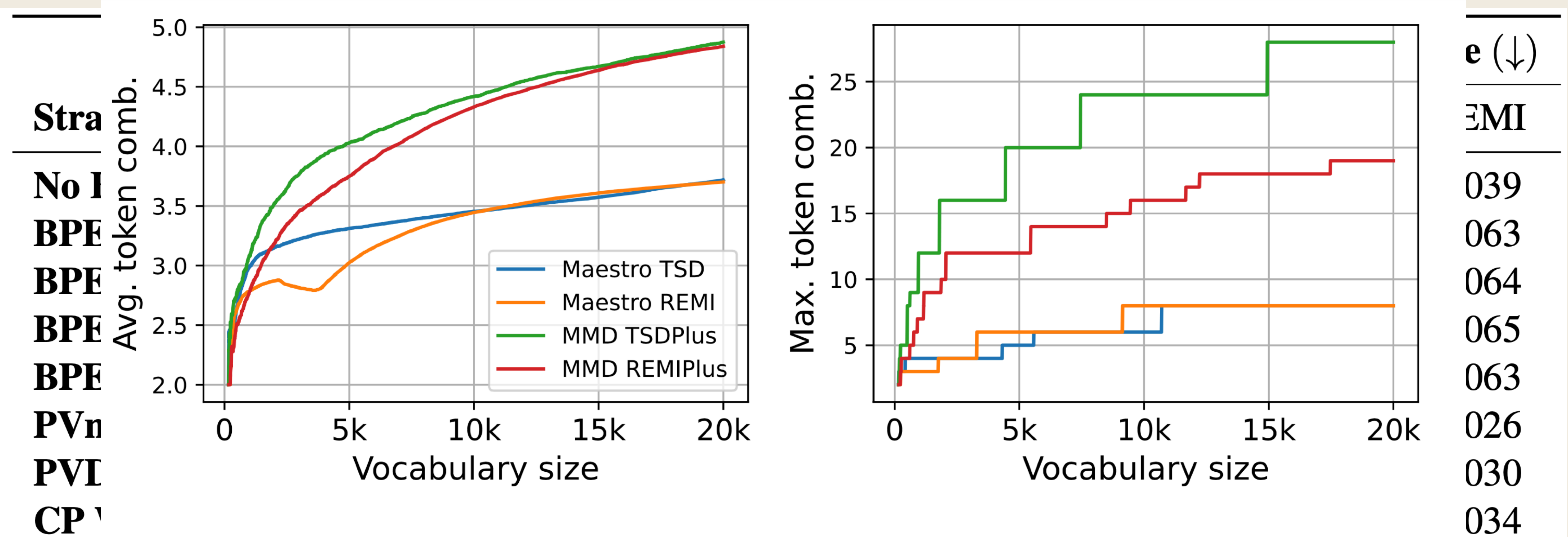
- Compression technique (Philip Gage) that iteratively replaces the most recurrent successive bytes of a corpus by newly created symbols
- Increase the vocabulary while reducing (compressing) the sequence length
- Widely used in NLP to build vocabularies of words from corpuses of characters (Sennrich et al.)
    - Words are the most recurrent byte successions
- Can tackle both the sequence length and embedding space usage problems

Iteration	Sequence	Vocabulary
0	a b a b c a b c	{a, b, c}
1	ab ab c ab c	{a, b, c, ab}
2	ab abc abc	{a, b, c, ab, abc}
3	ababc abc	{a, b, c, ab, abc, ababc}
4	ababcabc	{a, b, c, ab, abc, ababc, ababcabc}

Sequence length reduction

	Voc. size		tokens/beat (↓)		Tok. time (↓)		Detok. time (↓)	
Strategy	TSD	REMI	TSD	REMI	TSD	REMI	TSD	REMI
No BPE	149	162	18.5	19.1	0.174	0.151	0.031	0.039
BPE 1k	1k	1k	9.3 (-49.5%)	10.4 (-45.3%)	0.187	0.163	0.053	0.063
BPE 5k	5k	5k	7.0 (-62.2%)	8.5 (-55.2%)	0.181	0.165	0.053	0.064
BPE 10k	10k	10k	6.3 (-66.0%)	7.7 (-59.7%)	0.183	0.164	0.052	0.065
BPE 20k	20k	20k	5.8 (-68.9%)	6.9 (-63.9%)	0.184	0.163	0.052	0.063
PVm	1453	1466	13.4 (-27.8%)	13.8 (-27.4%)	0.134	0.123	0.024	0.026
PVDm	28185	28198	8.2 (-55.5%)	8.6 (-54.8%)	0.119	0.106	0.025	0.030
CP Word		188		8.6 (-54.8%)		0.169		0.034
Octuple		241		5.2 (-72.6%)		0.118		0.035

Sequence length reduction



(a) Avg. nb. of combinations (b) Max. nb. of combinations

Better and faster generation

	$\text{TSE}_{\text{type}}(\downarrow)$		$\text{TSE}_{\text{dupn}}(\downarrow)$		$\text{TSE}_{\text{time}}(\downarrow)$		Hum. Fidelity (↑)		Hum. Correctness (↑)		Hum. Diversity (↑)		Hum. Overall (↑)	
Strategy	TSD	REMI	TSD	REMI	TSD	REMI	TSD	REMI	TSD	REMI	TSD	REMI	TSD	REMI
No BPE	1.53	1.34	4.19	5.59	-	**28.93**	4.9%	4.0%	2.0%	2.0%	1.0%	0.0%	4.8%	0.0%
BPE 1k	1.59	0.62	3.60	4.16	-	34.65	13.6%	11.9%	11.8%	14.9%	10.8%	6.8%	8.6%	8.6%
BPE 5k	**0.31**	**0.38**	3.28	4.10	-	39.25	21.4%	**31.7%**	20.6%	21.8%	11.8%	11.7%	20.0%	18.1%
BPE 10k	0.49	1.04	3.83	6.39	-	48.16	23.3%	20.8%	**29.4%**	22.8%	18.6%	20.4%	22.9%	29.5%
BPE 20k	0.38	0.64	4.09	**3.60**	-	52.00	**29.1%**	19.8%	**29.4%**	**24.8%**	**36.3%**	**34.0%**	**30.5%**	**30.5%**
PVm	2.45	2.99	16.90	16.33	-	36.31	2.9%	2.0%	2.9%	0.0%	7.8%	2.9%	4.8%	1.0%
PVDm	0.63	6.32	**2.84**	10.64	-	46.75	4.9%	9.9%	3.9%	11.9%	13.7%	21.4%	8.6%	12.4%
CPWord		6.15		28.55		62.15		0.0%		2.0%		2.9%		0.0%
Octuple		-		244.11		305.43		0.0%		0.0%		0.0%		0.0%

Table 2: Metrics of generated results. TSE results are all scaled at e^{-3} for better readability. Hum stand for human, "-" for non-concerned (i.e. 0).

- Generated examples here (anonymized URL): ugtqphgirx.github.io/bpe-symbolic-music/

Better and faster generation

	tok/sec (↑)		beat/sec (↑)		note/sec (↑)		Voc. sampled (↑)	
Strategy	TSD	REMI	TSD	REMI	TSD	REMI	TSD	REMI
No BPE	40.2	43.8	4.5	9.9	10.6	10.9	100%	100%
BPE 1k	78.5	67.0	**13.0**	17.9	20.8	16.8	100%	99.9%
BPE 5k	99.1	83.9	12.8	**30.0**	26.7	20.7	100%	99.8%
BPE 10k	97.5	85.4	12.5	26.0	26.3	21.3	99.9%	99.9%
BPE 20k	**115.6**	**91.7**	12.9	24.9	**31.5**	22.7	99.4%	99.7%
PVm	59.3	58.1	8.2	12.2	15.9	14.9	99.3%	99.0%
PVDm	89.7	87.3	11.4	17.1	24.7	23.4	75.9%	74.3%
CPWord		75.8		15.2		19.0		76.7%
Octuple		-		14.3		**58.5**		57.4%

Table 3: Inference speeds (V100 GPU) and ratio of vocabulary sampled during generation. For tok/sec, the

	Genre (↑)		Artist (↑)	
Strategy	TSD	REMI	TSD	REMI
No BPE	0.836	0.796	0.907	0.876
BPE 1k	0.882	0.871	0.934	0.920
BPE 5k	0.901	0.875	0.933	0.925
BPE 10k	**0.904**	0.869	**0.937**	0.922
BPE 20k	0.851	0.877	0.909	0.923
PVm	0.853	0.810	0.905	0.886
PVDm	0.875	0.818	0.914	0.893
Octuple	-	**0.923**	-	**0.941**

Table 4: Average accuracy of classification models.

- Generated examples here (anonymized URL): ugtqphgirx.github.io/bpe-symbolic-music/

A better usage of embedding space

	Isoscore (↑)				PCA ID (↑)				FisherS ID (↑)			
	Gen / Maestro		Pt. / MMD		Gen / Maestro		Pt. / MMD		Gen / Maestro		Pt. / MMD	
Strategy	TSD	REMI	TSD	REMI	TSD	REMI	TSD	REMI	TSD	REMI	TSD	REMI
No BPE	0.899	0.883	0.925	0.730	62	66	44	45	5.4	5.2	8.1	7.9
BPE 1k	0.919	0.953	0.981	0.986	100	99	113	102	7.3	6.7	15.5	12.2
BPE 5k	0.965	0.962	0.989	0.989	131	119	145	119	9.0	8.6	16.7	13.7
BPE 10k	0.973	0.973	0.991	0.993	132	118	164	118	9.8	9.6	18.3	15.2
BPE 20k	0.976	0.981	**0.993**	**0.995**	**146**	**122**	**187**	**137**	**10.8**	**10.5**	21.4	16.9
PVm	**0.987**	**0.989**	0.961	0.961	71	67	52	52	7.1	6.8	13.9	14.7
PVDm	0.945	0.942	0.898	0.909	38	39	98	87	4.4	4.4	**24.1**	**22.8**

Table 6.5: Isoscore, and intrinsic dimension (ID) estimations. Gen. corresponds to the causal generative models, Pt. to the pretrained bidirectional models.

Use BPE!

- Faster training and generation
- Better results, better use of model space
- Already fully implemented in MidiTok, backed by 🤗tokenizers (superfast Rust implementation)
- Slightly longer tokenization / detokenization

Bibliography

- Hadjeres G., & Crestel, L. (2021). The Piano Inpainting Application.
- Schedl Markus, et al. Music Information Retrieval: Recent Developments and Applications, 2014.
- Gardner, Joshua P., et al. “MT3: Multi-Task Multitrack Music Transcription.” ICLR, 2022
- Dong H.-W., et al. “MuseGAN: Multi-Track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment”. *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018
- Huang Cheng-Zhi Anna, et al. “Counterpoint by Convolution.” ISMIR, 2017
- Huang Yu-Siang, and Yi-Hsuan Yang. “Pop Music Transformer: Beat-Based Modeling and Generation of Expressive Pop Piano Compositions.” Proceedings of the 28th ACM International Conference on Multimedia, 2020
- Von Rütte Dimitri, et al. “FIGARO: Controllable Music Generation Using Learned and Expert Features.” ICLR 2023
- Gao Tianyu, et al. “SimCSE: Simple Contrastive Learning of Sentence Embeddings.”, EMNLP 2021, pp. 6894–910
- Hsiao W.-Y., et al. “Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs”. *AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, May 2021, pp. 178-86
- Gage Philip. “A New Algorithm for Data Compression.” C Users J., vol. 12, no. 2, R & D Publications, Inc., Feb. 1994
- Sennrich, Rico, et al. “Neural Machine Translation of Rare Words with Subword Units.” ACL 2016, pp. 1715–25

Thank you

Any questions?