Adaptive Melodic-Entropy–Guided Preprocessing

for Lightweight MIDI Generation Models

Authors: Huaiyu Zhang, Dr. Chengwei Lei

California State University, Bakersfield

Motivation & Problem Background

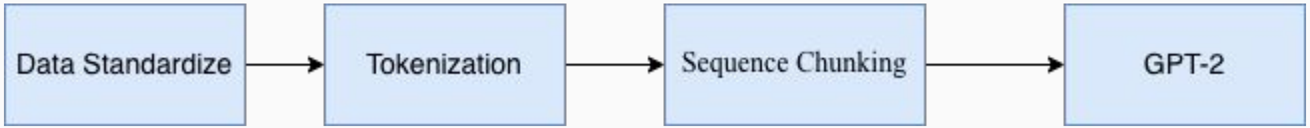
- **Independent musicians face increasing pressure**
  - High-cost cloud platforms
  - Data privacy concerns → **prefer local generation**
- **Large models cannot run locally**
  - GPU constraints (laptop / mobile / studio hardware)
- **Small models run fast — but generate poor-quality music**
  - Repetitive chords
  - Melodic collapse

- **Our Goal:**

  Enhance generation quality for lightweight models through structured data preprocessing.

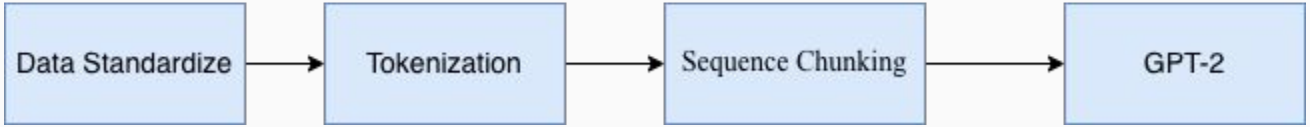
Overview

Traditional Pipeline



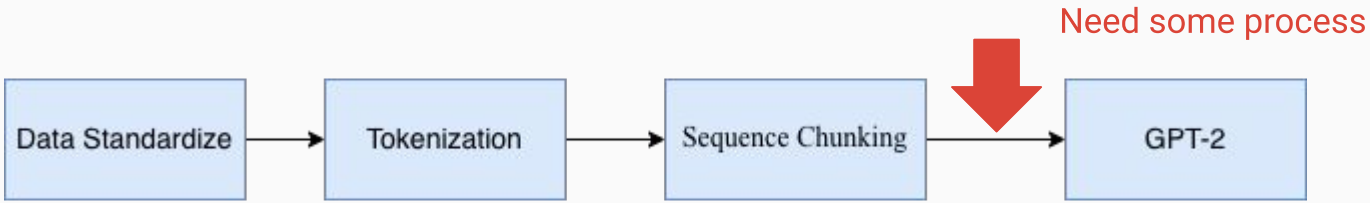
Overview

Disadvantage



The model's generative performance is largely determined by the scale of its parameters and the amount of training data. When both the dataset and model size are limited, the generation quality is noticeably degraded.

Overview

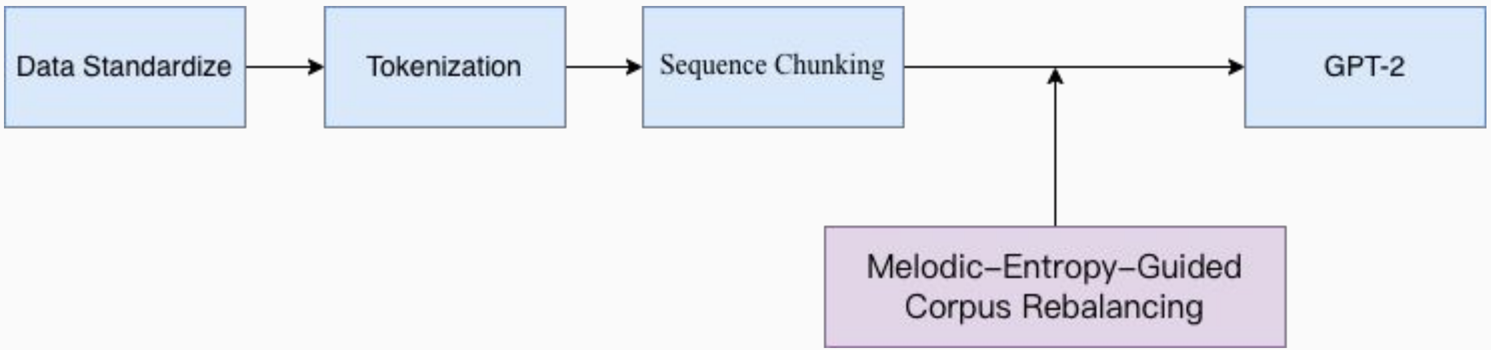


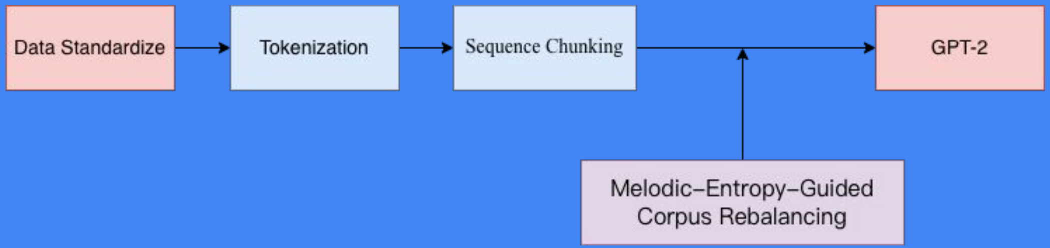
Observational insights:

- The model tends to persistently generate scalar patterns or repeat simple motifs.
- In many music corpora, scales and repetitive sequences occur at disproportionately high frequencies, which can obscure or overpower melodic information.
- Unlike text, the informational density in music is unevenly distributed.

Overview

Novel Method





Environment

Dataset: BitMidi (classic public MIDI collection)

- Large variety of genres
- Clean symbolic format (suitable for model alignment)

Transform Model Backbone: GPT-2 small

Component	Value
Hidden size	512
Layers	12
Heads	8
Head dim	64
Context length	256
Vocab size	3586
Dropout	0.1 (all)

About 39.8M parameters

Tokenization Method & Dimensionality Reduction

Comparison to REMI / REMI+

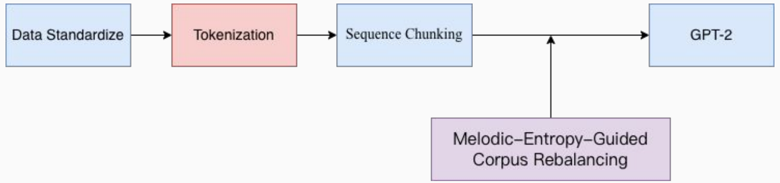
- **REMI:** bar-aware, beat-aware tokenization with rich MIDI events
- **Ours:** minimal melodic tokens → *pitch + position + duration*

Key reductions:

- Removed: velocity, tempo, chord, program, control, drums
- Vocabulary: **3,586** (compact)
- Shorter sequences → **small model learns better**

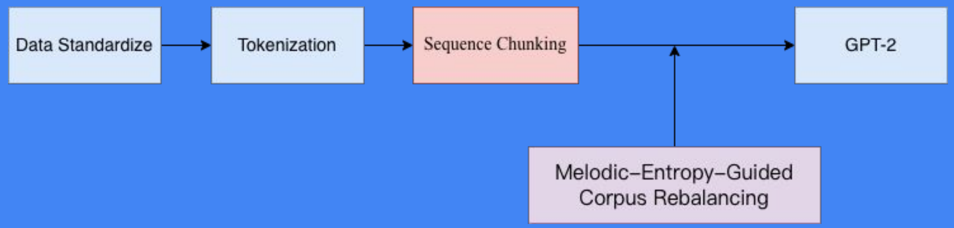
Further reduction: remove pitch extremes + cluster offset values → smaller vocabulary.

REMI: Huang & Yang, *Pop Music Transformer*, ACM MM 2020.



Traditional | Our Method

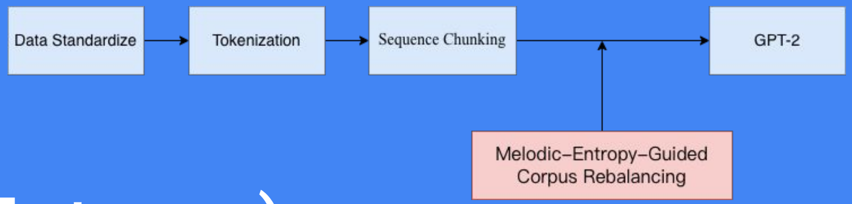
Feature	REMI / REMI+	Proposed Lightweight Tokenization
Representation Type	Event-based symbolic encoding	Event-based symbolic encoding
Bar Token	explicit	Implicit
Beat / Position Tokens	✅	✅
Pitch	✅	✅
Duration	✅	✅
Velocity	✅	❌
Tempo / Chord Tokens	✅	❌
Program / Instrument	✅	❌
Other MIDI Events	✅	❌
Vocabulary Size	3k–30k	3,586
Sequence Length	Longer and sparser due to richer event space	Shorter and denser, improving learnability for small models



BaseLine Method

Baseline: Bar-Level Slicing (Music Transformer)

- Detect bar onsets
- Cut fixed-length slices
- Uniform pitch-shift augmentation
- Ignores melodic complexity
- Fast and simple, but loses structural variation



Novel Method (Melody Entropy)

Pitch-Interval Change

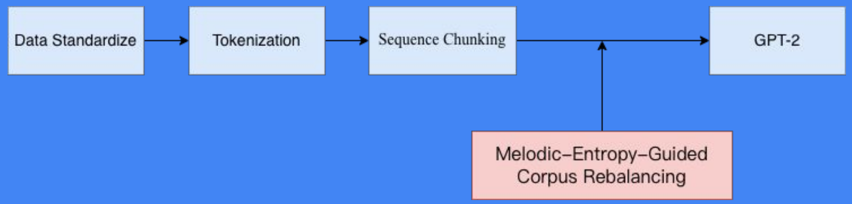
$$\Delta p_t = p_t - p_{t-1}$$

- Convert melody to a **beat-aligned pitch sequence**
- Compute **interval changes** between consecutive notes
- Removes dependence on absolute pitch → **transposition-invariant**

Entropy of Interval Distribution

$$H(W) = -\sum_{i \in \mathcal{I}} \widehat{p}_W(i) \log(\widehat{p}_W(i) + \varepsilon),$$

- W: sliding window of melodic events
- p_W(i): empirical probability of interval i within window W
- epsilon: small numerical constant for stability
- Measures local melodic unpredictability
- Low entropy: repetitive, stepwise, chordal regions
- High entropy: varied, expressive, melody-rich segments



Novel Method (Melodic-Entropy–Guided Preprocessing)

$$F_H = \text{CDF of all } H_f(u)$$

Purpose: Builds a global statistical distribution of melodic entropy across the entire dataset.
Input → Output: Takes all window entropy scores and produces their cumulative distribution.

$$q_k = F_H^{-1}(\beta_k)$$

Purpose: Quantile function Converts chosen percentiles into concrete entropy thresholds.
Input → Output: Takes a percentile and returns the corresponding entropy cutoff.

$$r_k = \frac{|\{(f,u) : b(f,u) = k\}|}{\sum_j |\{(f,u) : b(f,u) = j\}|}.$$

Purpose: Measures how frequent each entropy bucket is in the raw data.
Input → Output: Counts bucket assignments and produces the natural ratio for that bucket.

$$s_k = \frac{\alpha_k}{\max(\varepsilon, r_k)}, \qquad \varepsilon > 0.$$

Purpose: Computes how strongly each bucket should be sampled in the final dataset.
Input → Output: Compares the desired ratio with the raw ratio and outputs a sampling weight.

Novel Method local optimize

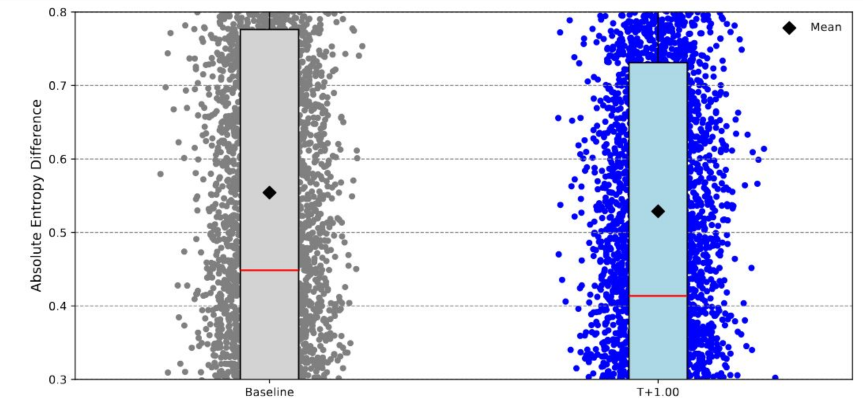
Different values of t change how much the dataset favors simple or complex melodic slices.
Negative t gives more simple patterns; positive t gives more complex, expressive ones.

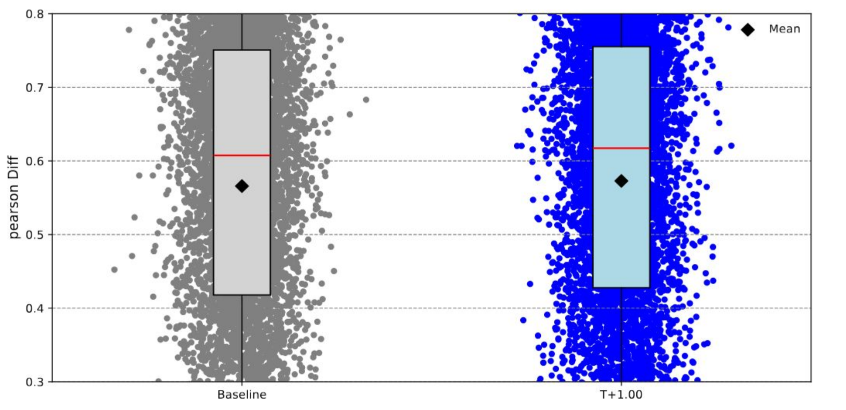
t	Mean $\lvert \Delta H \rvert$	Std.	Mean r	Std.
−1.00	0.5593	0.4651	0.5729	0.2342
−0.75	0.5502	0.4596	0.5719	0.2327
−0.50	0.5449	0.4603	0.5793	0.2341
−0.25	0.5531	0.4645	0.5757	0.2362
+0.00	0.5407	0.4654	0.5820	0.2333
+0.25	0.5337	0.4644	0.5831	0.2326
+0.50	0.5360	0.4671	0.5813	0.2325
+0.75	0.5307	0.4678	0.5840	0.2329
+1.00	0.5288	0.4618	0.5836	0.2307

Baseline vs Novel Method

Aspect	Baseline: Bar-Level Slicing	Proposed: Entropy-Guided Pipeline
Segmentation	Fixed bar-aligned slices	Entropy-conditioned variable windows
Musical Structure Awareness	Ignores melodic complexity	Uses melodic entropy as structural signal
Overlap Policy	standard	Entropy-dependent overlap
Augmentation	Uniform pitch-shift	Selective, entropy-conditioned augmentation
Data Efficiency	Low; heterogeneous slices	Higher; focuses on informative regions
Melody-Rich Coverage	Often misses melodic peaks	Explicitly targets melody-dense segments
Representative Method	Music Transformer	This work

Baseline vs Novel Method





Model	Mean $\lvert \Delta H \rvert$	Std.	Mean r	Std.
Entropy-guided	0.5288	0.4618	0.5836	0.2307
Music Transformer baseline	0.5542	0.4623	0.5660	0.2354

Our method outperforms the baseline in both average entropy deviation and Pearson similarity of pitch distributions.

Conclusions

Conclusion

- Entropy-aware preprocessing boosts melody coherence at no extra cost.
- Better than bar-level slicing on both metrics.
- Works even on small corpora.

Limitations

- Tune t more intelligently.
- Move beyond fixed quartiles.

Thanks!

Acknowledgment
This work is supported by the Office of GRaSP at California State University, Bakersfield, and the CV PATH Research Program.

Contact us:

California State University, Bakersfield

hzhang5@csub.edu