Key Monkey

ML Model for Piano Notes

Bridging the gap in current methods of
Prediction of Piano Notes.

Problem and Context

Prediction of Piano Notes

The objective of this project is to develop a machine learning approach that can predict piano notes from an audio recording of a piano piece. The system should be able to:

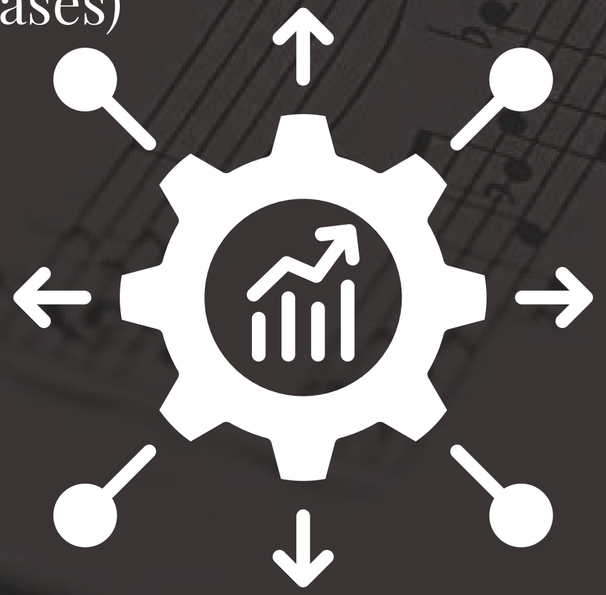
- Detect the next sequence of notes in the piano piece
- Identify note timing (onset and offset)
- Identify velocity.

Challenges in Piano Music Prediction

- Chords: multiple keys pressed simultaneously
- Sustain pedal: notes overlap and decay unpredictably
- Long-range dependencies: musical phrases span seconds to minutes

Applications

- Automatic music transcription (MIDI from audio)
- Music education tools (real-time feedback for learners)
- Composer assistance (continuation/completion of phrases)
- Digital archiving of live performances



Impact

A more accurate piano transcription system can:

- Reduce manual effort in creating sheet music
- Supports accessibility
- Enable better AI-driven music tools

Debate & Research Gaps

DEBATE & RESEARCH GAPS

FRAME BASED	SQUENCE BASED
• Breaking the music into millions of frames of milliseconds. For every frame the model must know what note is being played. • Relies on CNN based models to look for patterns in the Spectrograms. • Argues that it is more accurate to capture the exact timings of onset and offset of the notes. • Lacks Musicality.	• Treats Music as a language instead of partitioning it into frames. • Uses Transformers to predict the MIDI Files as if it was a sentence instead of predicting small parts seperately. • Argues that Transformers based models are better since they understand the music. • Faces memory issues during long songs (>15mins).

Limitations of Current Methods

Frame-Based Models uses CNN

- Good timing detection
- Poor long-term musical understanding
- May split sustained notes into smaller ones

Sequence-Based Models uses Transformers

- Good musical context understanding
- Struggles with dense chords and sustain pedal
- Memory issues for long music pieces

Current systems either:

- Focus on precise timing detection, or
- Focus on musical structure understanding

1. **Offset Problem:** The Piano notes don’t end instantly. There is some sound even after the Key is released which makes the Frame based models produce the wrong output.

Mi, J., Kim, S., & Toda, T. (2024). Improved architecture for high-resolution piano transcription to efficiently capture acoustic characteristics of music signals. 2024 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 1–6. https://doi.org/10.1109/apsipaasc63619.2025.10848780

2. **Hallucination Problem**: Sometimes due to overlapping Notes and Pedal Effect, Transformer based models hallucinate and produce notes which are not even present in the music.

Zhao, H., & Liu, S. (2025). Design and implementation of piano audio automatic music transcription algorithm based on convolutional neural network. EURASIP Journal on Audio, Speech, and Music Processing, 2025(1). https://doi.org/10.1109/isas64331.2024.10845524

Datasets

MAESTRO DATASET

Contains high-quality classical piano performances recorded as both audio and MIDI files (Yamaha Disklavier)

- Over 200 hours of performance audio files
- 1184 individual performances

WHY WAS THIS CHOSEN?

- Highly accurate
- Very large amount of data
- Rich with features

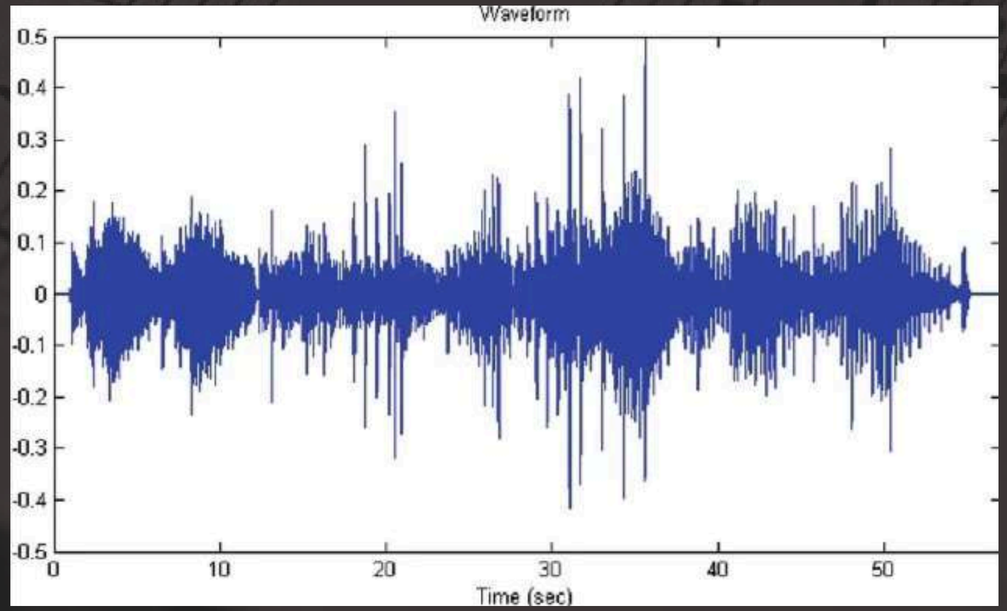
Feature	Description
Pitch	MIDI pitch value (0–127)
Velocity	Key strike intensity
Onset Time	When the note starts
Duration	How long the note is held
Meta Data	Composer, title, year

ETHICAL CONCERNS

- The recordings were from a public competition between 2004–2011
- The dataset's first version was released in 2018
- The dataset under a Creative Commons Attribution license, which allows for research use while giving proper credit to the performers.

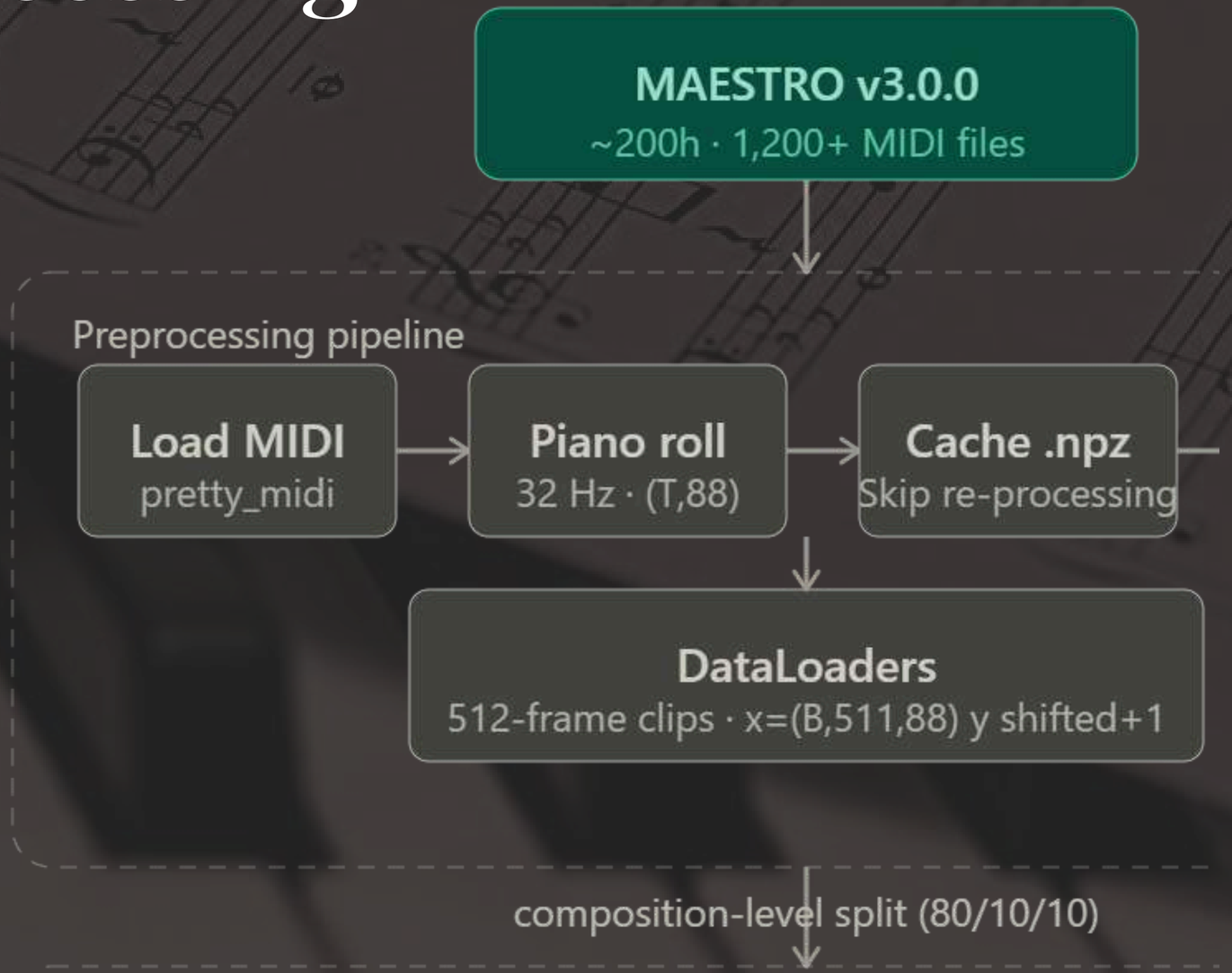
.MIDI OUTPUT VS .WAV OUTPUT:

```
PS C:\Users\samee> cd Downloads
PS C:\Users\samee\Downloads> py MLPR_Pro.py
Loaded successfully!
Instrument: 0
Pitch: 39 Start: 1.03125 End: 1.06640625 Velocity: 51
Pitch: 51 Start: 1.15234375 End: 1.1888020833333333 Velocity: 62
Pitch: 63 Start: 1.01953125 End: 1.296875 Velocity: 54
Pitch: 51 Start: 1.2903645833333333 End: 1.33203125 Velocity: 64
Pitch: 58 Start: 1.0143229166666665 End: 1.40234375 Velocity: 61
Pitch: 51 Start: 1.4453125 End: 1.4830729166666665 Velocity: 62
Pitch: 51 Start: 1.58203125 End: 1.61328125 Velocity: 59
Pitch: 51 Start: 1.7109375 End: 1.7486979166666665 Velocity: 63
Pitch: 51 Start: 1.8541666666666665 End: 1.88671875 Velocity: 55
Pitch: 67 Start: 1.0078125 End: 2.0143229166666665 Velocity: 71
D#2 Start: 1.03125 End: 1.06640625
D#3 Start: 1.15234375 End: 1.1888020833333333
D#4 Start: 1.01953125 End: 1.296875
D#3 Start: 1.2903645833333333 End: 1.33203125
A#3 Start: 1.0143229166666665 End: 1.40234375
D#3 Start: 1.4453125 End: 1.4830729166666665
D#3 Start: 1.58203125 End: 1.61328125
D#3 Start: 1.7109375 End: 1.7486979166666665
D#3 Start: 1.8541666666666665 End: 1.88671875
G4 Start: 1.0078125 End: 2.0143229166666665
```



Methodology

Preprocessing

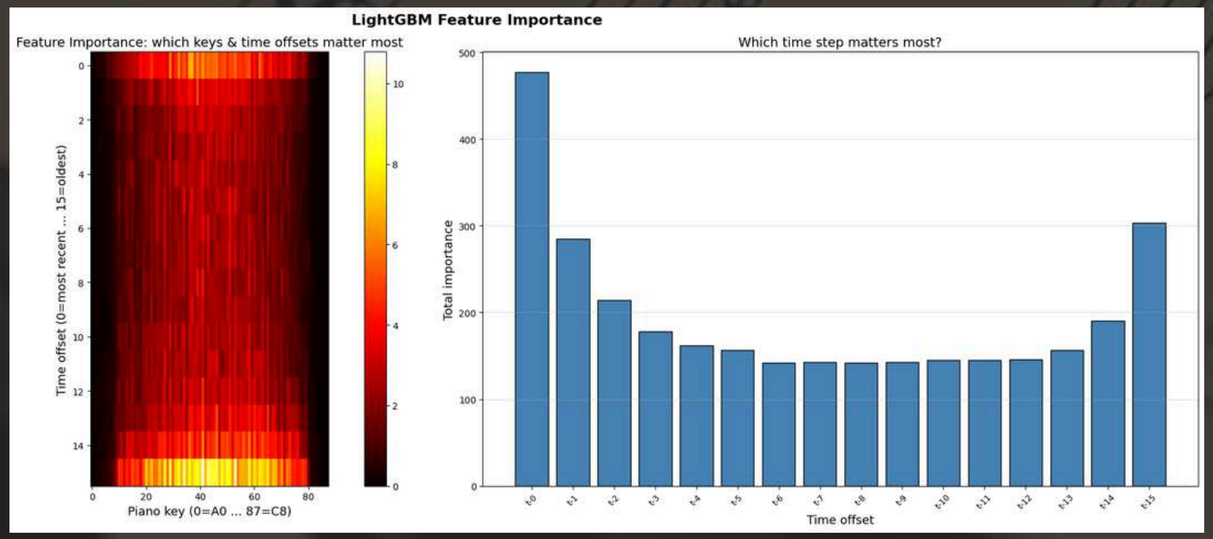


Naive Approach

Classical Machine Learning Models

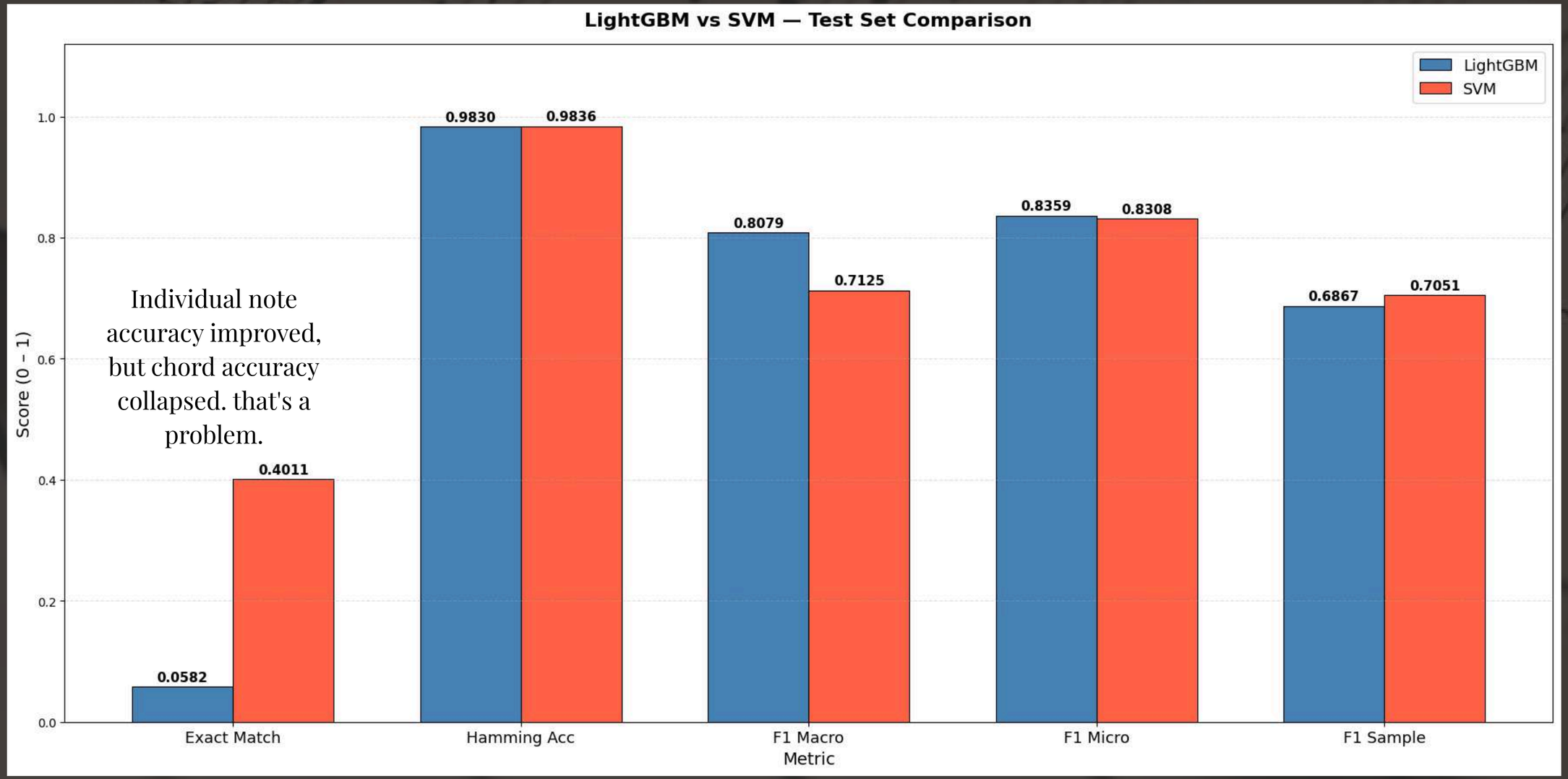
LightGBM

- Approach: 88 independent binary classifiers - one per piano key.
- Sliding window of 16 consecutive frames ↠ flatten ↠ 16 × 88 = 1,408 features
- ~4.6M windows



SGD-SVM

- Approach- Linear Support Vector Machine using: SGDClassifier
- 20,000 windows, 1408 features



Individual note accuracy improved, but chord accuracy collapsed. that's a problem.

Sequential model

Recurrent Neural Networks (shared architecture)

Model training

Recurrent neural networks

RNN	GRU	LSTM
Baseline	2 gates	4 gates

hidden=256 · layers=2 · dropout=0.3

Input: (batch, 512, 88)

↓

RNN Layer [hidden=256, layers=2, dropout=0.3]

↓

Linear FC: 256 → 88 (one logit per key)

↓

Output: (batch, 512, 88) - next-frame predictions

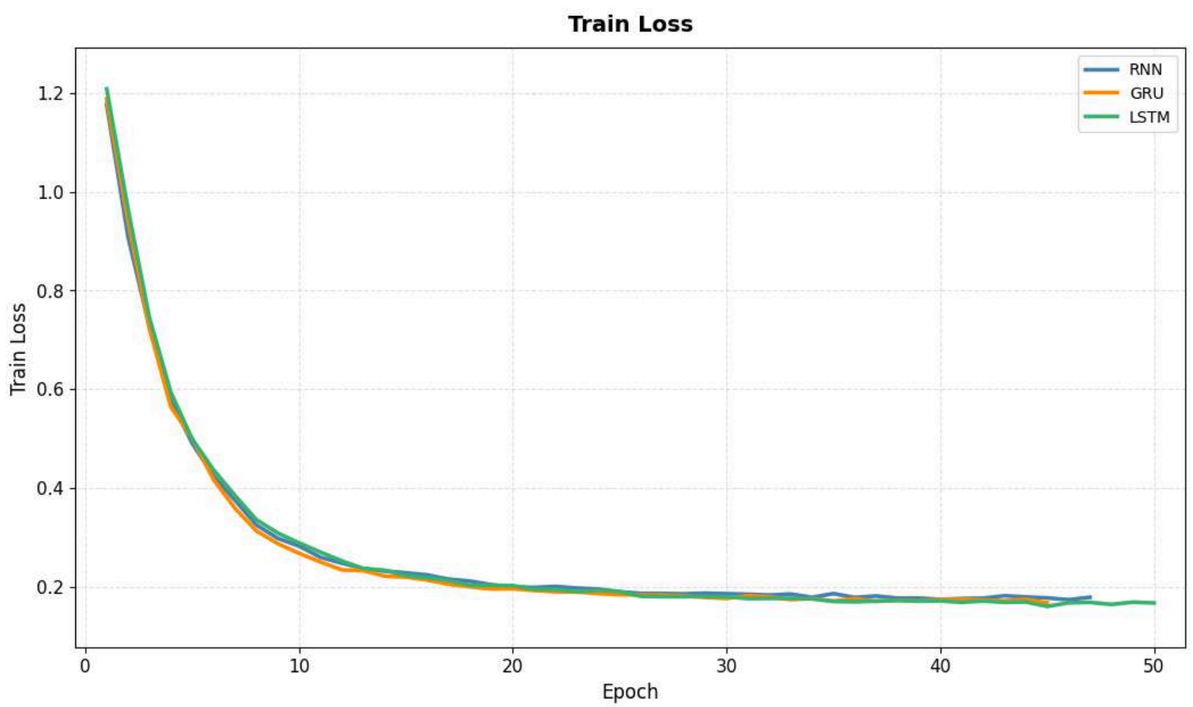
$$\text{pos\_weight}=\frac{\text{active cells}}{\text{silent cells}}=17{:}1$$

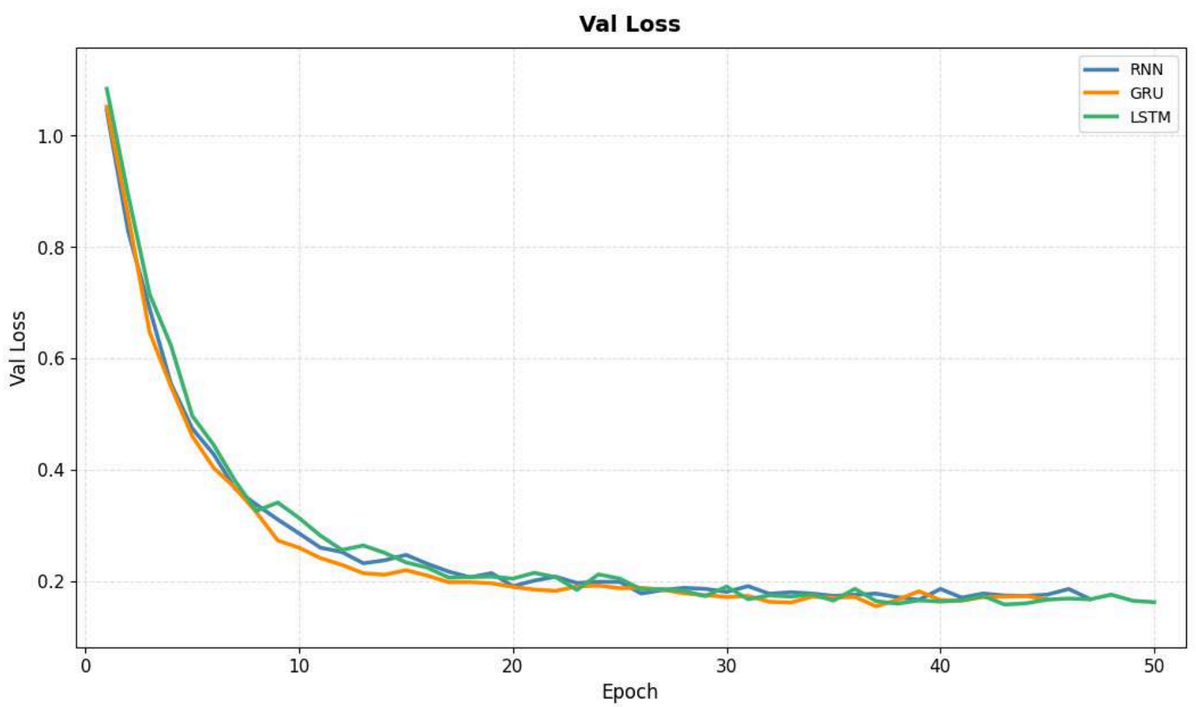
pos_weight
-17:1 imbalance

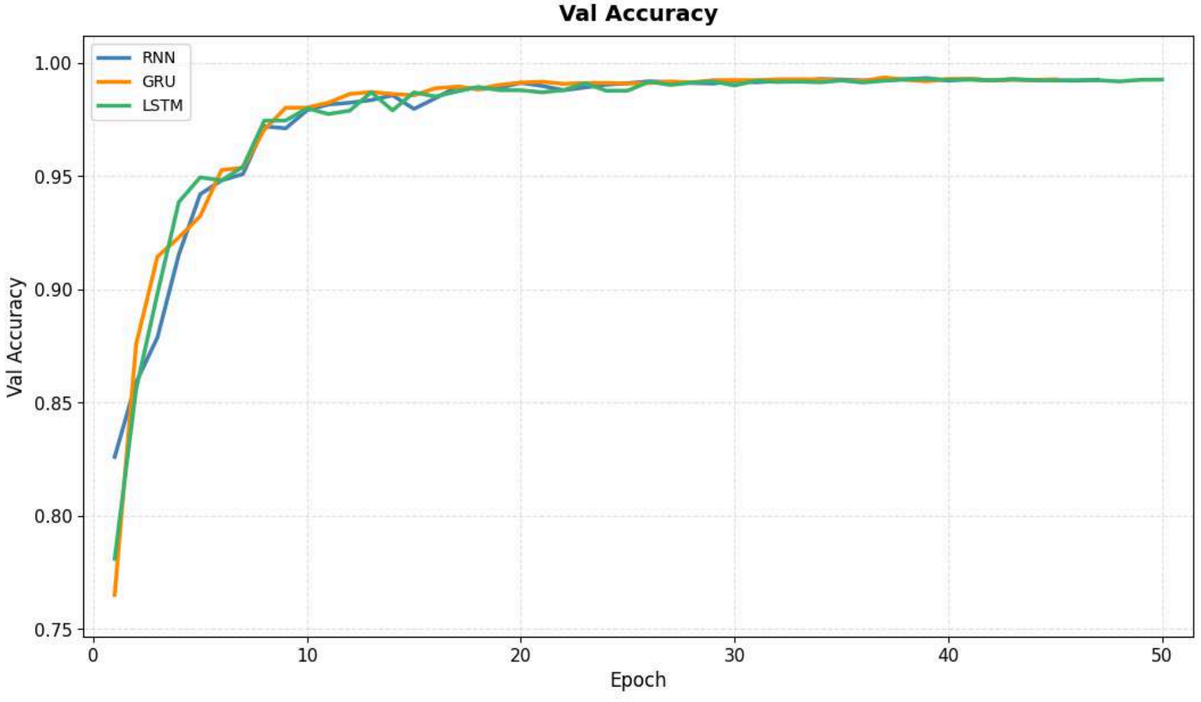
- prevent collapse into all-silence prediction
- upweights the rare active-key class in BCEWithLogitsLoss.

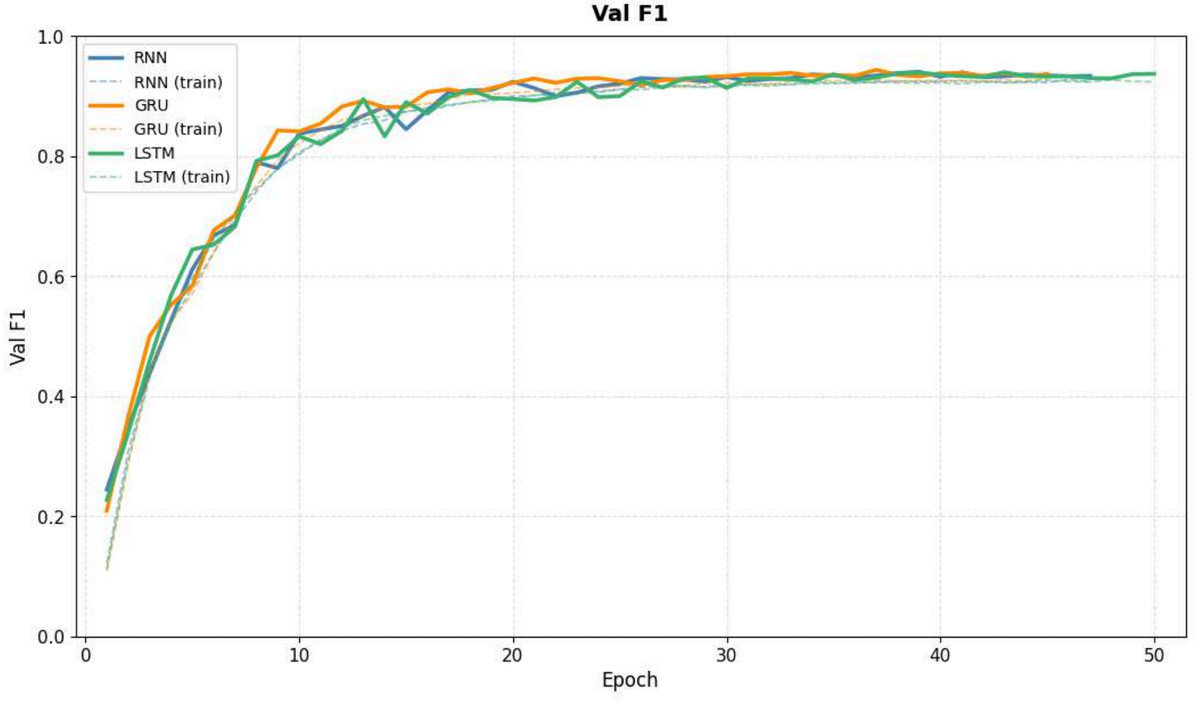
→

Training strategy
TBPTT chunk=64 · LR warmup+cosine · early stopping
auto threshold search · AdamW · checkpoint resume


Train Loss
RNN
GRU
LSTM
Train Loss
Epoch


Val Loss
RNN
GRU
LSTM
Val Loss
Epoch


Val Accuracy
RNN
GRU
LSTM
Val Accuracy
Epoch


Val F1
RNN
RNN (train)
GRU
GRU (train)
LSTM
LSTM (train)
Val F1
Epoch

Results

	Precision	Recall	F1-score
A#3	0.00	0.00	0.00
A3	0.50	1.00	0.67
A4	1.00	1.00	1.00
B3	1.00	1.00	1.00
C#4	0.71	1.00	0.83
C4	0.80	1.00	0.89
C5	0.00	0.00	0.00
D4	0.95	0.87	0.91
E4	1.00	0.72	0.84
F#4	1.00	1.00	1.00
F4	1.00	0.76	0.86
G#3	1.00	1.00	1.00
G4	1.00	0.53	0.69
Rest	0.00	0.00	0.00
Accuracy			**0.78**
Macro avg	**0.71**	**0.71**	**0.69**
Weighted avg	**0.96**	**0.78**	**0.85**

Approach	Precision	Recall	F-Measure
Mel cyclic spectrogram	91.07%	83.09%	86.90%
Mel STFT spectrogram	92.87%	83.89%	88.15%
Both	**94.27%**	**88.01%**	**91.03%**

Dai, J., Zheng, Q., Wang, Y., Shan, Q., Wan, J., & Zhang, W. (2025, November 29). Multi-feature fusion for automatic piano transcription based on Mel Cyclic and STFT spectrograms. MDPI. https://www.mdpi.com/2079-9292/14/23/4720

Model	Params	Onset, Offset & Velocity		
		P (%)	R (%)	F1 (%)
High-resolution [reproduced]	20M	81.68	79.28	80.44
HPPNet-sp [19]	1.2M	83.29	81.24	82.24
HRplus	2.7M	**84.31**	**81.69**	**82.96**
HRplus-hybrid	**0.9M**	83.91	80.79	82.30

Mi, J., Kim, S., & Toda, T. (2024b, September 29). Improved architecture for high-resolution piano transcription to efficiently capture acoustic characteristics of music signals. arXiv.org. https://arxiv.org/abs/2409.19614

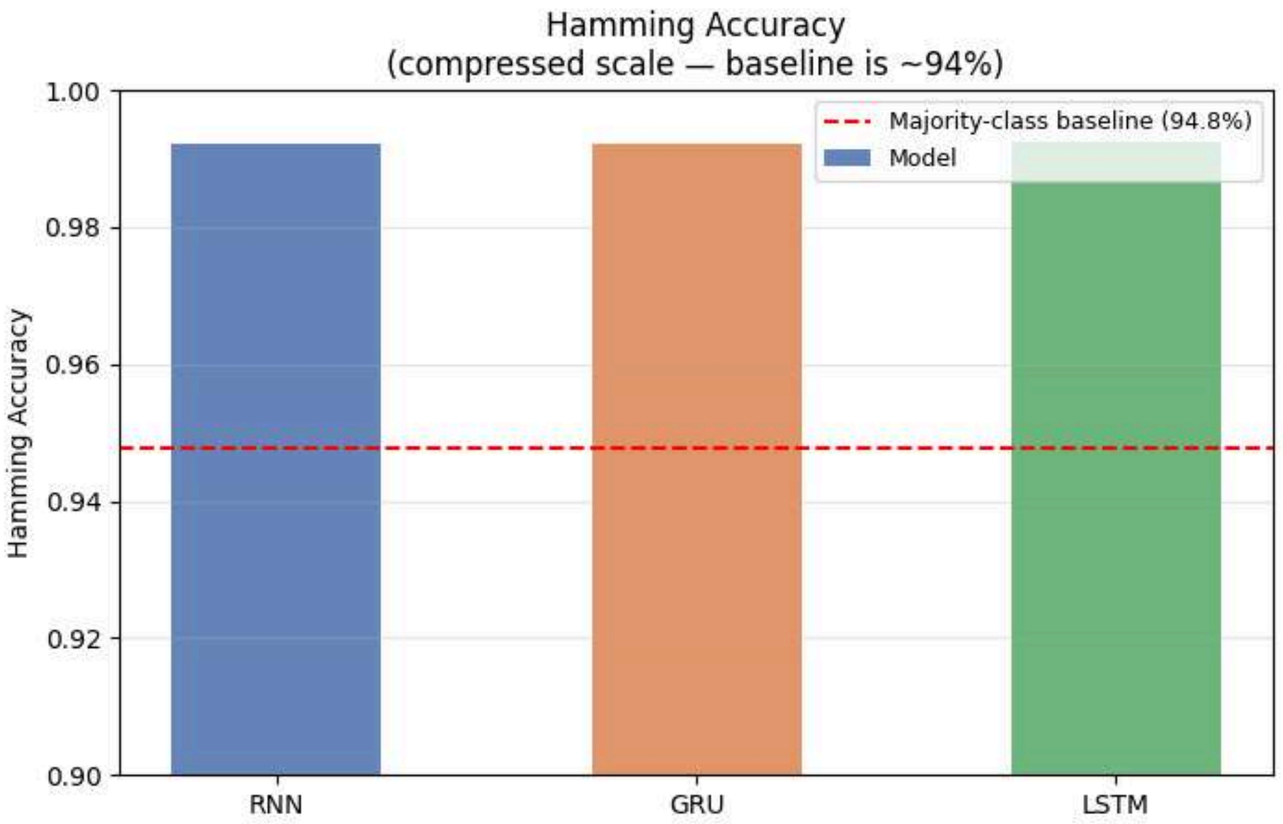
Metric design

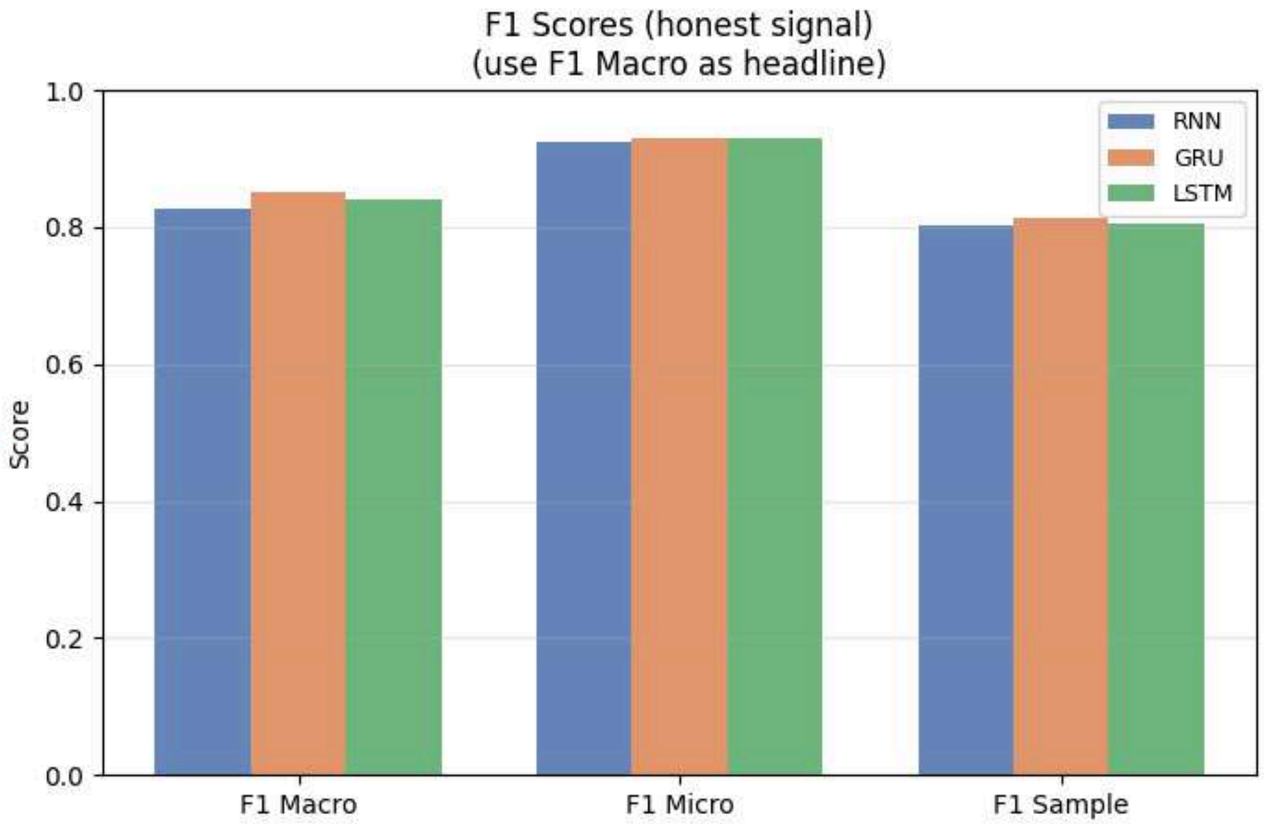
Metric	What It Measures	Why It Matters
Exact Match	All 88 keys correct per frame	Strictest; rare even for good models
Hamming Accuracy	Per-key accuracy averaged	Inflated by silence
F1 Macro	F1 equally weighted across all 88 keys	Sensitive to rare/extreme-register keys
F1 Micro	Global F1 over all key-frame pairs	Overall active-note detection quality
F1 Sample	F1 averaged per frame (chord-level)	Measures per-chord prediction quality

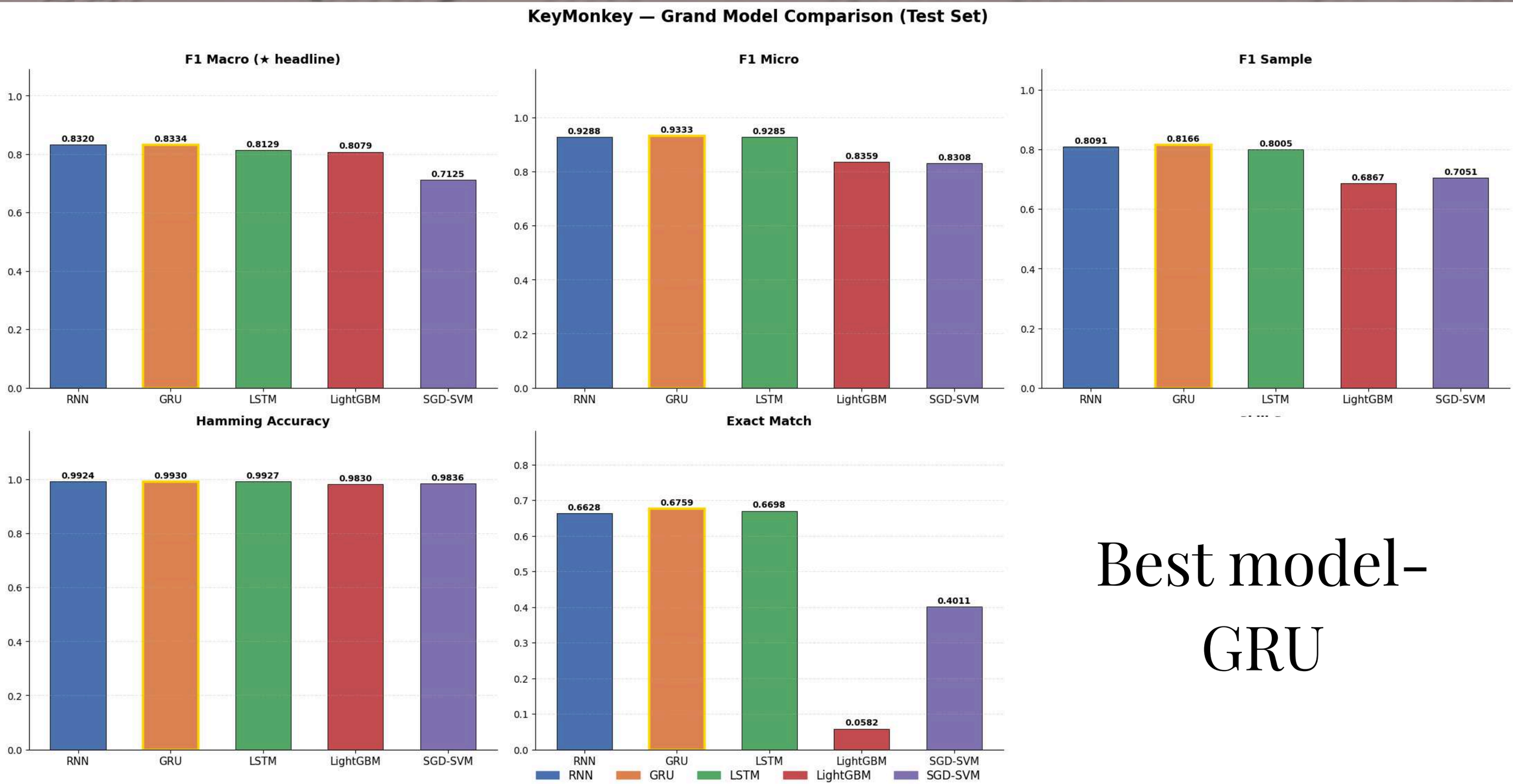
```
================================================================================
Baseline                                HammAcc   F1Micro   F1Macro    F1Samp
================================================================================
Majority-class (silence)                 0.9472    0.0000    0.0000    0.0000
Copy-forward (x_t -> x_{t+1})            0.9943    0.9460    0.8853    0.8305
================================================================================

NOTE: Majority-class Hamming accuracy will be ≈ 0.94 because 94% of
      cells are silent. This is why Hamming accuracy is a useless headline metric.
      Copy-forward F1 is the real bar your models must beat.
```

Metric Choice: Hamming Accuracy vs. F1 Scores







Best model- GRU

Challenges

1. Correctly identifying the “Release of the note” while resonating decay of the piano.
2. Ensuring that the model does not run into Hallucination Problem.
3. Identifying if there was data leakage and overfitting
4. Compute Constraints — Training 3 RNNs on 200h of audio required TBPTT, checkpointing, and Google Drive caching to fit within **Colab's session limits.**


Thank You