MUSIB: Musical Score Inpainting Benchmark

Mauricio Araneda Hernandez
Universidad de Chile
26 de Abril

Profesor Guia:
Felipe Bravo, UCH
Denis Parra (Co-guia), PUC

Miembros del Comite:
Eduardo Graells, UCH
Nelson Baloian, UCH
Eliana Scheihing, UACH

Outline

Introduction

Motivation

- How to use AI to enhance human ability to create music?
- How can people control/interact with IA-based models to achieve this goal?

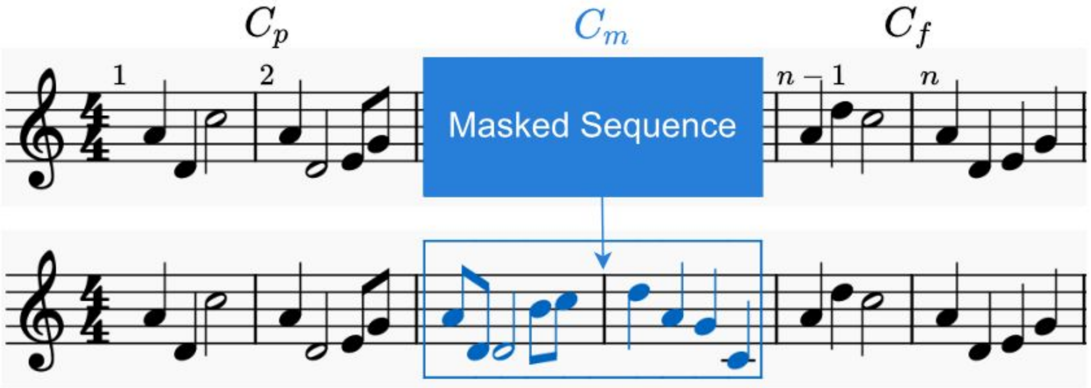
Preamble

- People tend to create music starting by creating small pieces and then assembling them into a larger piece. [1]
- Process is highly not sequential.
- Nature of the process contrasts with most common approaches to Music Generation in AI.
- **Music Inpainting Task**, a sub-task of Music Generation better models this procedure.

[1] B. Bogunović, Creative cognition in composing music, 2019

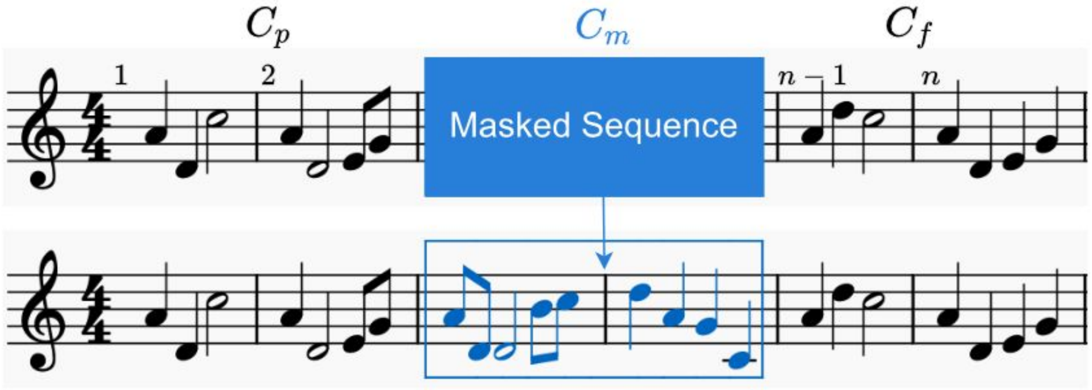
Music Inpainting Task Definition

- Given: a past musical context Cp, a future musical context Cf, the modeling task is to generate an inpainted sequence Cm which can connect Cp and Cf in a musically meaningful manner.



Issues in evaluation

- Proposed methods lack of standardized evaluation setups.
- Different data representation, datasets, metrics and baselines.
- We don't know the state of the art, and thus, we don't know if we are making progress.



Evaluation Challenges

- Metrics values differ when changing **representations** for the exact same data.
- The sets of **metrics for evaluation changes from paper to paper**, measuring different features.
- Training and evaluation of models done over **different datasets** that vary in characteristics such as: format, number of samples, style, notes distribution, etc.
- The output is generated through a random process.

Hypothesis:

It is possible to find a unifying pattern across several models of musical score inpainting that enables a direct comparison of approaches.

Additionally, we argue that it is possible to extend current evaluation procedures to measure the expected variability of a model.

Objective:

To develop an evaluation framework to properly compare different approaches for musical score inpainting, thus providing solid evidence to define the current progress of this task and its state of the art.

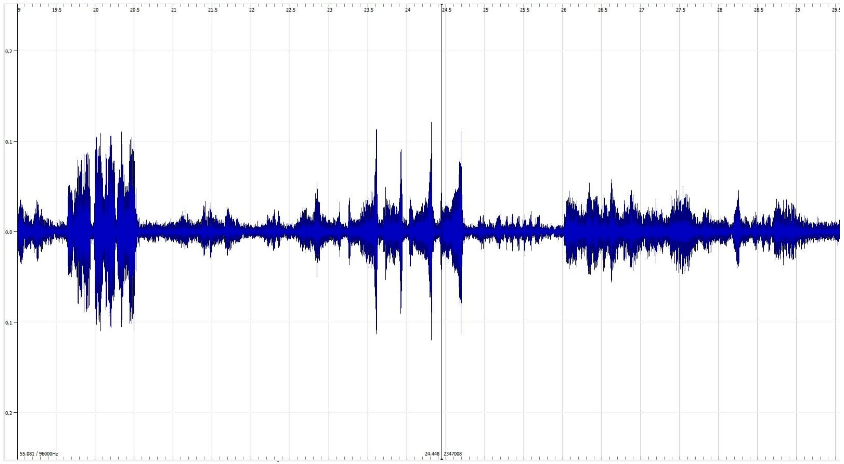
Background & Preliminary Concepts

Data representation

- Raw Audio vs Symbolic Music
- Data vectorization

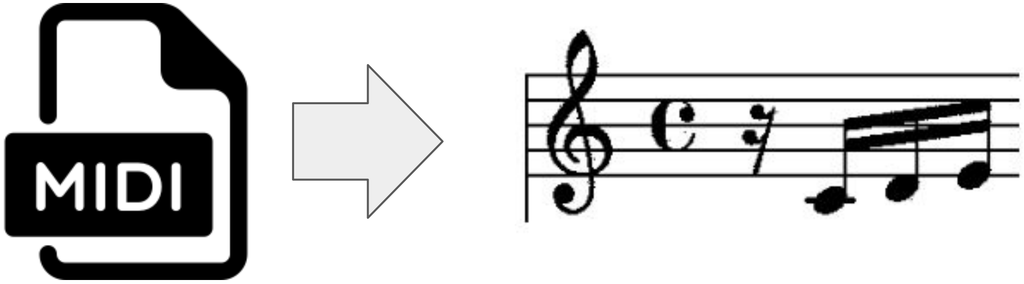
Data representation

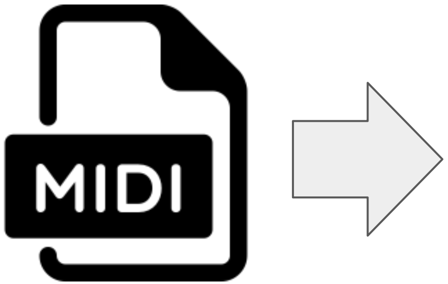
- **Raw Audio vs Symbolic Music**
- Data vectorization



Data representation

- Raw Audio vs Symbolic Music
- **Data vectorization**

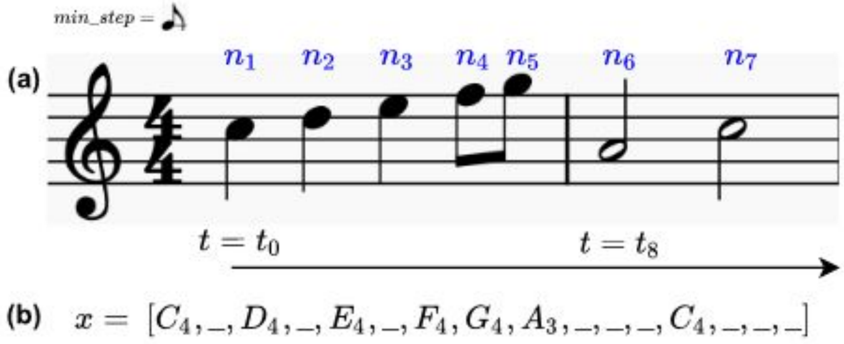




Track	Time	Event	Channel	Note	Velocity
2	96	Note_on	0	60	90
2	192	Note_off	0	60	0
2	192	Note_on	0	62	90
2	288	Note_off	0	62	0
2	288	Note_on	0	64	90
2	384	Note_off	0	64	0

Data representation

- Raw Audio vs Symbolic Music
- **Data vectorization**



Data representation

- Raw Audio vs Symbolic Music
- **Data vectorization**

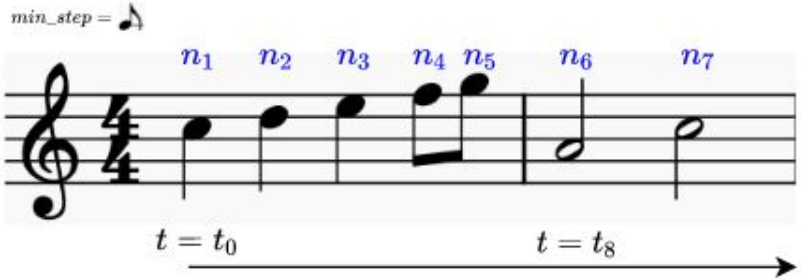
(c)

$$
\begin{aligned}
x &= (x_{pitch}, x_{rhythm}) \\
x_{pitch} &= [C_4, D_4, E_4, F_4, G_4, A_3, C_4] \\
x_{rhythm} &= [1,0,1,0,1,0,1,1,1,0,0,0,1,0,0,0]
\end{aligned}
$$

Data representation

- Raw Audio vs Symbolic Music
- **Data vectorization**

(a)



(d) $x = [\ [n_1, n_2, n_3, n_4, n_5], [n_6, n_7]\ ]$

n	Tempo	Bar Start	Position	Pitch	Velocity	Duration
n_1	120	1	0 | 8	C_4	90	♩
n_2	120	0	2 | 8	D_4	90	♩
n_3	120	0	4 | 8	E_4	90	♩
n_4	120	0	6 | 8	F_4	90	♪
n_5	120	0	7 | 8	G_4	90	♪
n_6	120	1	0 | 8	A_3	90	𝅗𝅥
n_7	120	0	4 | 8	C_4	90	𝅗𝅥

Our Proposal: MUSIB

Our proposal: MUSIB

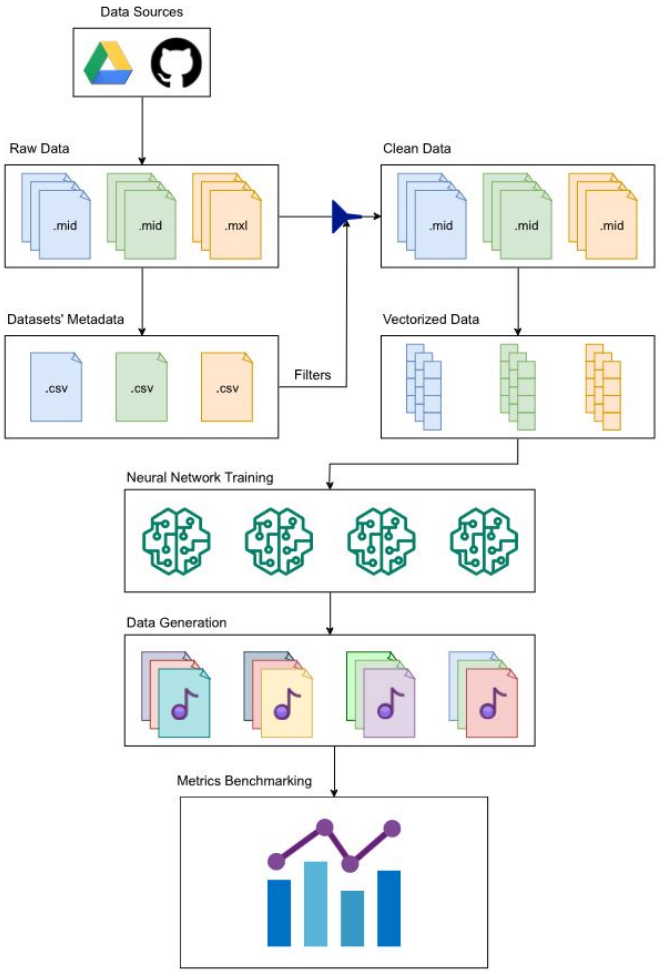


Figure 4.1: Diagram of the overall data pipeline in MUSIB.

Datasets

Raw datasets:

- IrishFolkSong (~45k songs)
- JSBChorales (~300 songs)

Context inputs:

- IrishFolkSong (~300k samples)
- JSB Chorales (~2.4k samples)

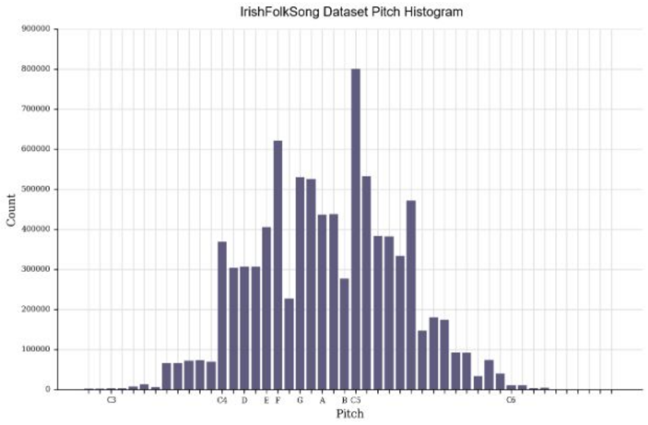


Figure 3.1: Pitch distribution for the IrishFolkSong dataset.

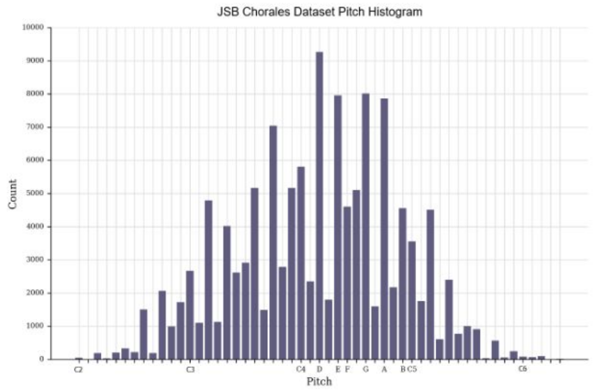


Figure 3.2: Pitch distribution for the JSB Chorales dataset.

Data Cleaning

Monophonic datasets:

- Empty files
- Repeated files
- 4/4 Time Signature
- Monophony
- Min Length (16 measures)

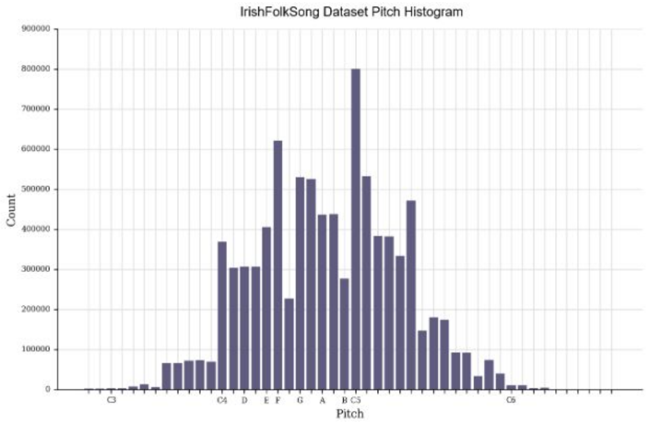


Figure 3.1: Pitch distribution for the IrishFolkSong dataset.

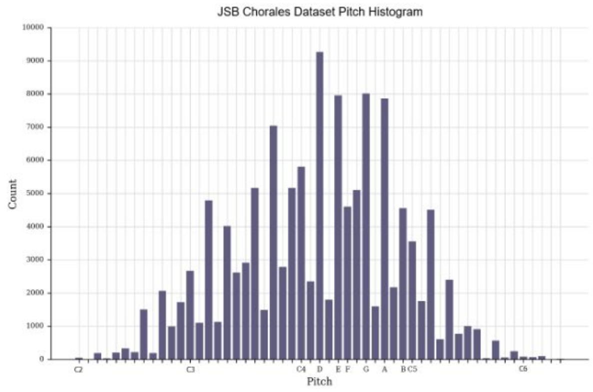


Figure 3.2: Pitch distribution for the JSB Chorales dataset.

Music Inpainting Models

- Out of 8 models, we selected 4 models based on on the feasibility of replicating their code in a single environment:
- InpaintNet
- SketchNet
- AnticipationRNN
- VLI

Model	Architecture	Year	Music Type	Base Framework
CocoNet	CNN	2017	Polyphony	TensorFlow
DeepBach	RNN	2017	Polyphony	Pytorch
InpaintNet	VAE + RNN	2019	Monophony	Pytorch
SketchNet	VAE + RNN	2020	Monophony	Pytorch
AnticipationRNN	RNN	2020	Monophony	Pytorch
VLI	XL-Net	2021	Polyphony	Pytorch
DiffModel	Diffusion models	2021	Monophony	Flax
MusIAC	Transformer	2022	Polyphony	Pytorch

Table 2.1: Existing models for music inpainting

InpaintNet

- Based in VAE encoding for each measure.
- Temporal modeling through RNN.

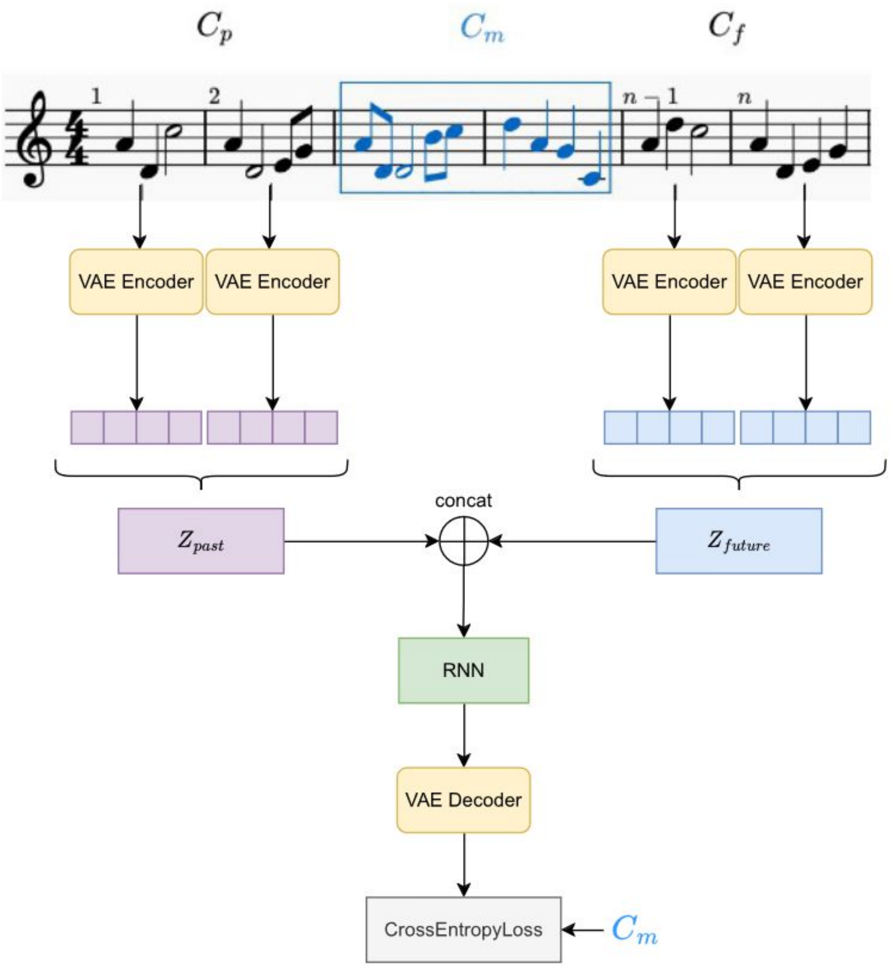


Figure 2.9: Diagram of the Music Inpaintnet architecture.

SketchNet

- Based in VAE encoding for each measure.
- Separated encoding for Rhythm and Pitch.
- Temporal modeling through GRU

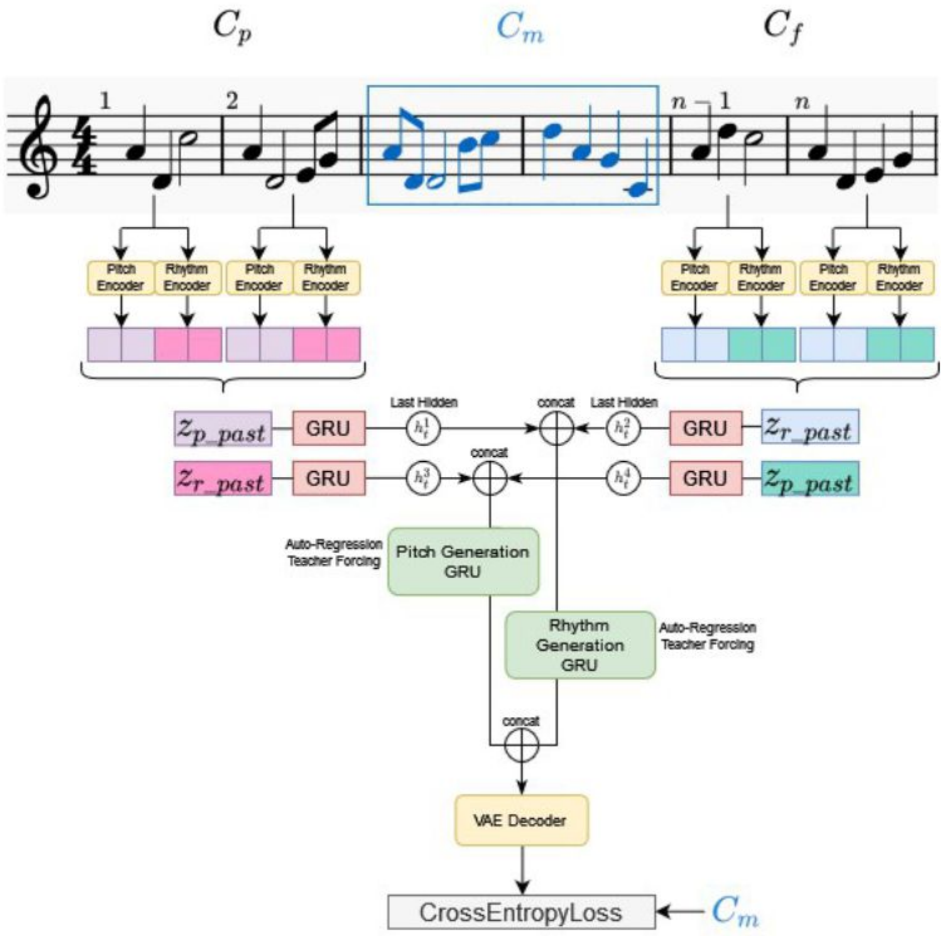


Figure 2.10: Diagram of the Music SketchNet architecture.

Anticipation RNN

- Each time-step is a token.
- Temporal modelling through RNN.

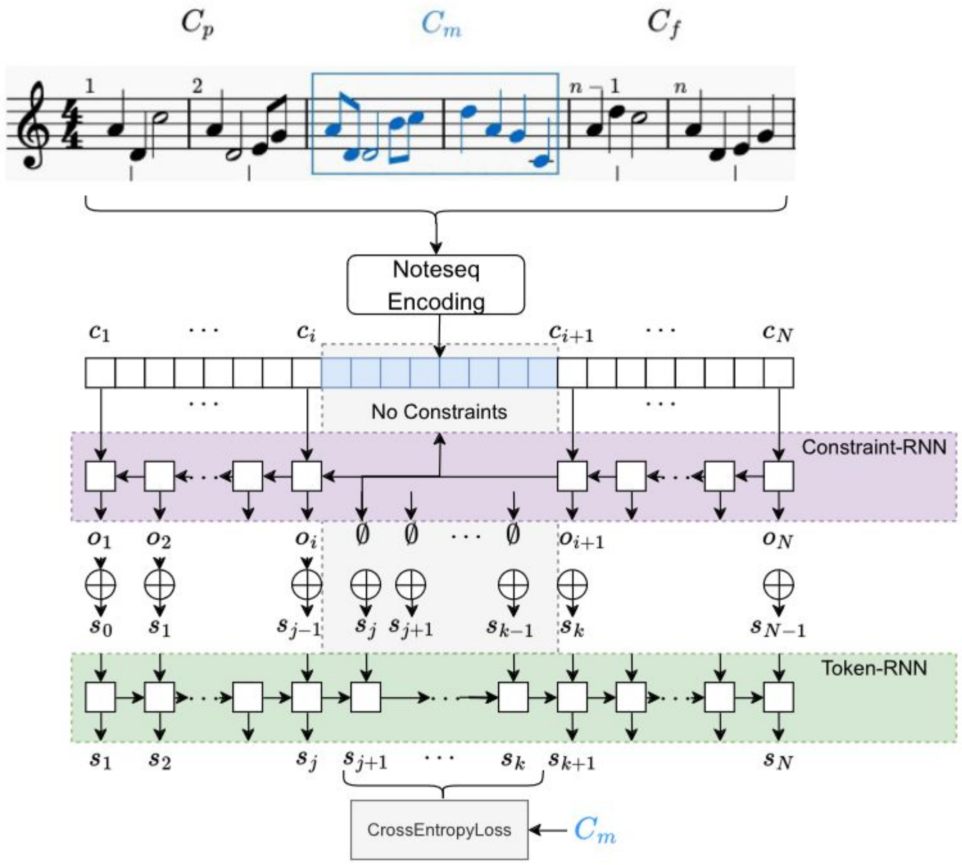
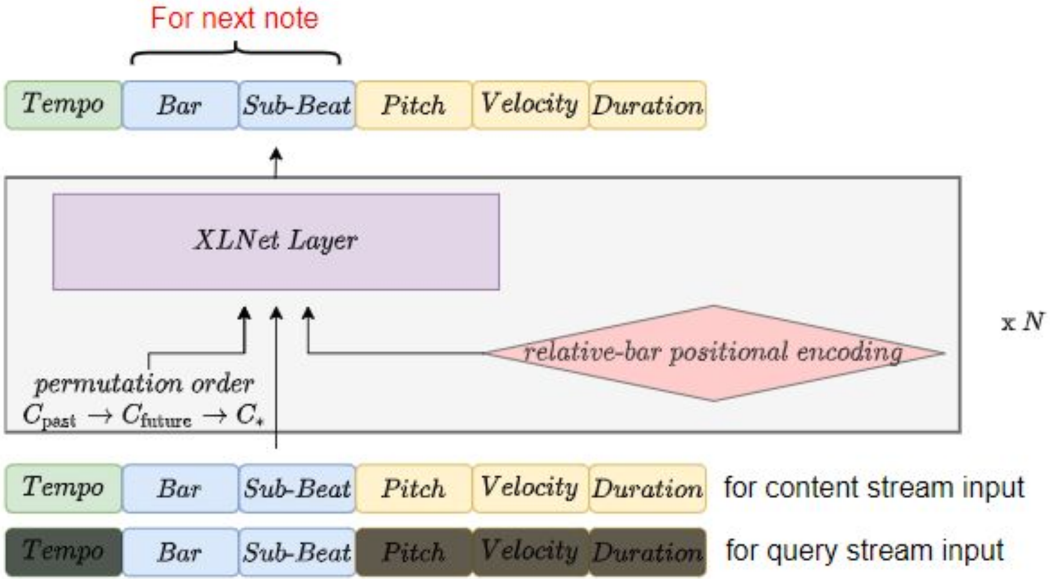


Figure 2.11: Diagram of the Anticipation RNN.

VLI

- Discretization based on relative position of each note.
- Temporal modelling through pretrained XL-Net.



Metrics

We propose two types of metrics:

- Note Metrics
- Divergence Metrics

Note Metrics

Directly compare note attributes in predicted data vs true data, one note at time.

We argue that for measuring the quality of notes predicted, we need to compare at least three dimensions:

- Position
- Pitch
- Rhythm.

Position Score

- Metric proposed in this work that measures the similarity of two musical sequences in terms of the position of their notes.
- We argue that to correctly measure notes’ position similarity, a metric needs to be able to:

1. Be equipped with a strategy to align the notes’ positions within gold and predicted sequences independently of the order in which they appear.

2. Handle sequences with potentially different number of notes.

3. Reward sequences that share the same positions for their notes.

4. Penalize sequences that do not share the same positions for their notes.

5. Penalize generated sequences with different number of notes than expected

Position Score

- We construct our metric as an F1 score calculated from gold and predicted note's positions whose internal variables (i.e., True Positives, False Positives, False Negatives) are computed as follows:

  • True Positives (TP): A note's position is present in both sequences.

  • False Positives (FP): A note's position is present in the generated sequence when it was not present in the gold sequence.

  • False Negatives (FN): A note's position is missing in the generated sequence when it was present in the gold sequence.

Note that True Negatives are not part of the F1 score function and thus its definition is not stated here.

Position Score

Next, we discuss how each of the the aforementioned requirements are satisfied by our F1 metric:

- 1. By defining the process of alignment based on checking the presence of a note within a given sequence we resolve the ordering problem between non-matching sequences.
- 2. Building the internal variables of the F1 Score based on the alignment of positions allows us to compare sequences with different number of notes since the match of positions for the i-th and j-th note may occur at arbitrary indexes in arbitrary long sequences.
- 3. Both values precision and recall will increase as the number of True Positives increases, increasing F1-Score performance, and thus rewarding sequences that share positions.
- 4. Both values precision and recall will decay as FP and FN increase. Note that metric functions such as Accuracy would not be able to penalize missing notes (FN). Additionally, there is no difference in cost for different types of mis-classifications in this task. Either adding or removing notes to the generated sequence with respect to the gold sequence would have the same impact in musicality. Due to this, both the recall and precision do not need particular weights when being evaluated, discarding alternatives such as $F\beta$ functions.
- 5. If the generated sequence contains more notes than the true sequence, the number of false positives will increase. Similarly, if the number of notes is smaller than the true sequence, the number of false negatives will increase. Both cases imply that F1-Score will decrease in performance, either by a worse Recall or Precision. This implies that Position Score penalizes sequences with a different number of notes than expected.

Pitch Accuracy

Firstly defined by Chen et Al. (2020), is the percent of pitches correctly predicted over the total of pitches in a sequence.

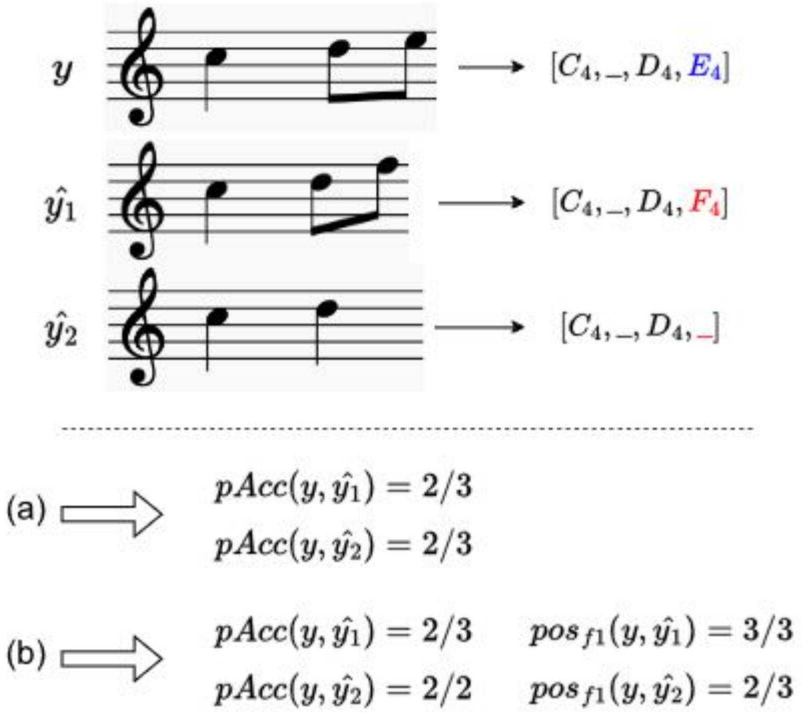
The metric is thought as a comparison of two musical sequences, where if a pitch is present at a given time index, the metric function checks the equality of this pitch in the same index for the other sequence.

For our evaluation procedure we slightly modified the application of the metric. We argue that the result of this metric may be misleading in explaining two fundamentally different musical phenomenons.

Pitch Accuracy edge-case

With this metric as is, a mismatch of pitch might represent either:

1. The first note and the note to be compared (both at time index i) do not share the same pitch (e.g. one note is F3 and the other one is D4), or
2. There is a note at time index i for the first sequence, but there is no matching note at the same time index in the sequence to compare because there is a silence or hold token.



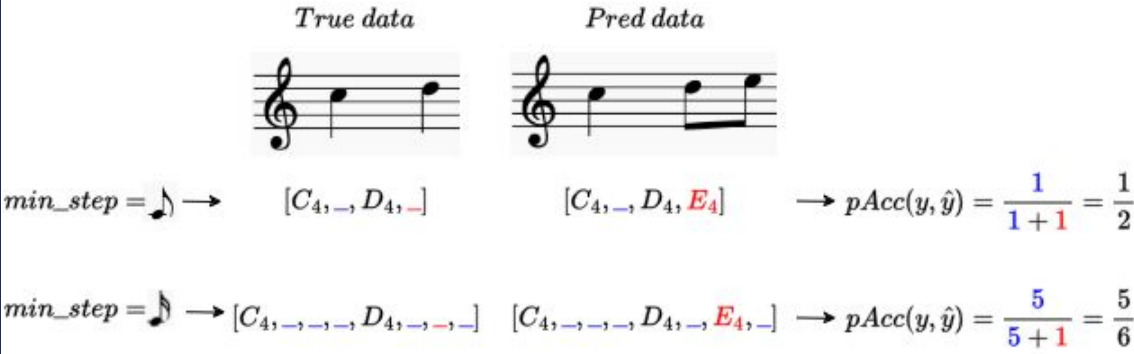
Rhythm Accuracy

Firstly defined by Chen et Al. [11], is the percent of notes' duration correctly predicted over the total of notes.

Rhythm Accuracy edge-case

We argue that this metric as is does not correctly measure the performance of the models due to differences in the results when it is applied to the same data with different notes' resolutions.

Note that the issue comes from the fact that the duration of a note is stored as multiple tokens, one per time-step. Changing the resolution of the sequence affects the representation of hold/silence classes while keeping intact the number pitch classes. This unbalances the overall distribution and raises errors where rhythm tokens are confused with pitch tokens.

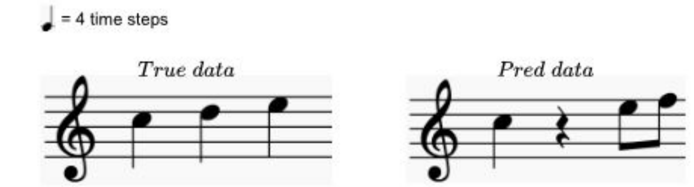


Rhythm Accuracy fix

In order to fix this behaviour we need to transform the input data before applying the metric such that the rhythm is a single value attached to a note instead of multiple values distributed among multiple time steps.

This can be done by representing each note as Note-based discretization including the number of time-steps that a note is held as the rhythm value.

The comparison then is applied similarly to Pitch Accuracy, where if two notes match in position, then the rhythm values of both notes are compared else the comparison is skipped and falls under Position Score evaluation.



	Position	*Pitch*	*Duration*
$n_{1\ true}$	0	60	4
$n_{2\ true}$	4	62	4
$n_{3\ true}$	8	64	4

	Position	*Pitch*	*Duration*
$n_{1\ pred}$	0	60	4
$n_{2\ pred}$	8	64	2
$n_{3\ pred}$	10	65	2

	TP	FP	FN
Position	2	1	1
Pitch	2	0	–
Duration	1	1	–

$TP = |\{x \mid x \in Pred \land x \in True\}|$

$FP = |\{x \mid x \in Pred \land x \notin True\}|$

$FN = |\{x \mid x \notin Pred \land x \in True\}|$

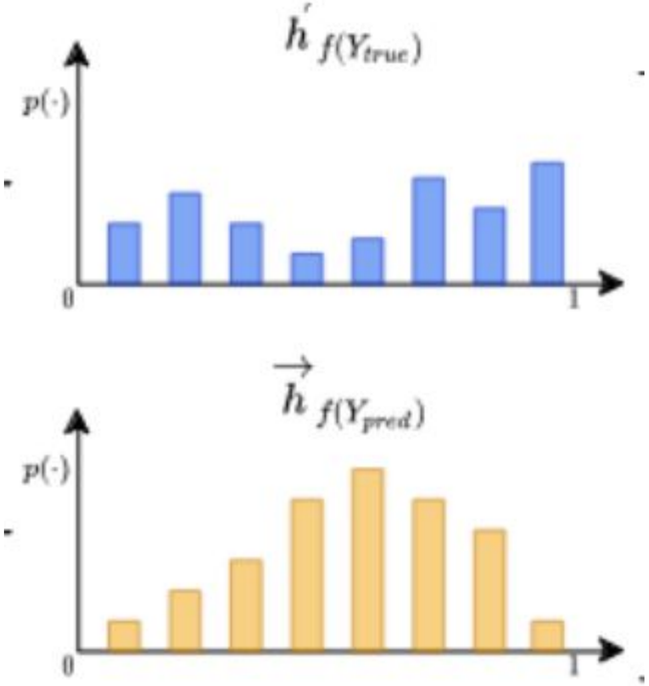
$$pos_{precision} = \frac{2}{2+1} = 0.67 \qquad pitch_{acc} = \frac{2}{2+0} = 1$$

$$pos_{recall} = \frac{2}{2+1} = 0.67 \qquad rhythm_{acc} = \frac{1}{1+1} = 0.5$$

$$pos_{f1} = 0.67$$

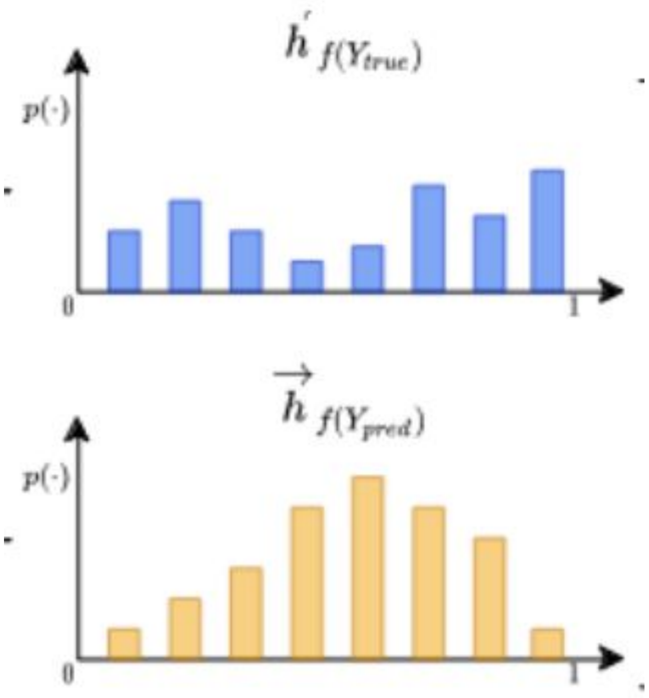
Divergence Metrics

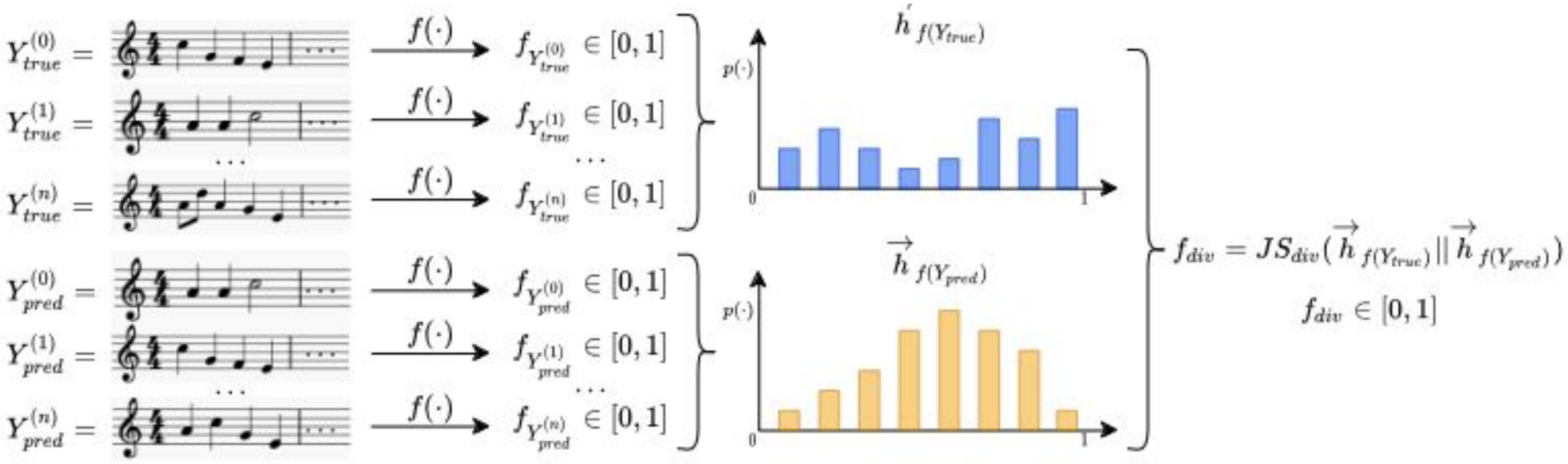
- Variability in music is common and even desirable. Note Metrics don’t capture that.
- How do we verify that a given musical attribute in a set of predicted songs is within the correct range of variability?
- Look at the distributions!



Divergence Metrics

- We can apply a function that maps a sequence to a given number
- The set of samples will transform into a distribution of values.
- If the training set and generated set have similar distributions, the models is doing a good job mimicking the musical properties of the dataset.





$$f_{div}(Y_{true}||Y_{pred}) = JS_{div}(\overrightarrow{h}_{f(Y_{true})}||\overrightarrow{h}_{f(Y_{pred})})$$

Note Metrics

Results

Results – IrishFolk

IrishFolk Dataset ($\approx$ 300K samples)

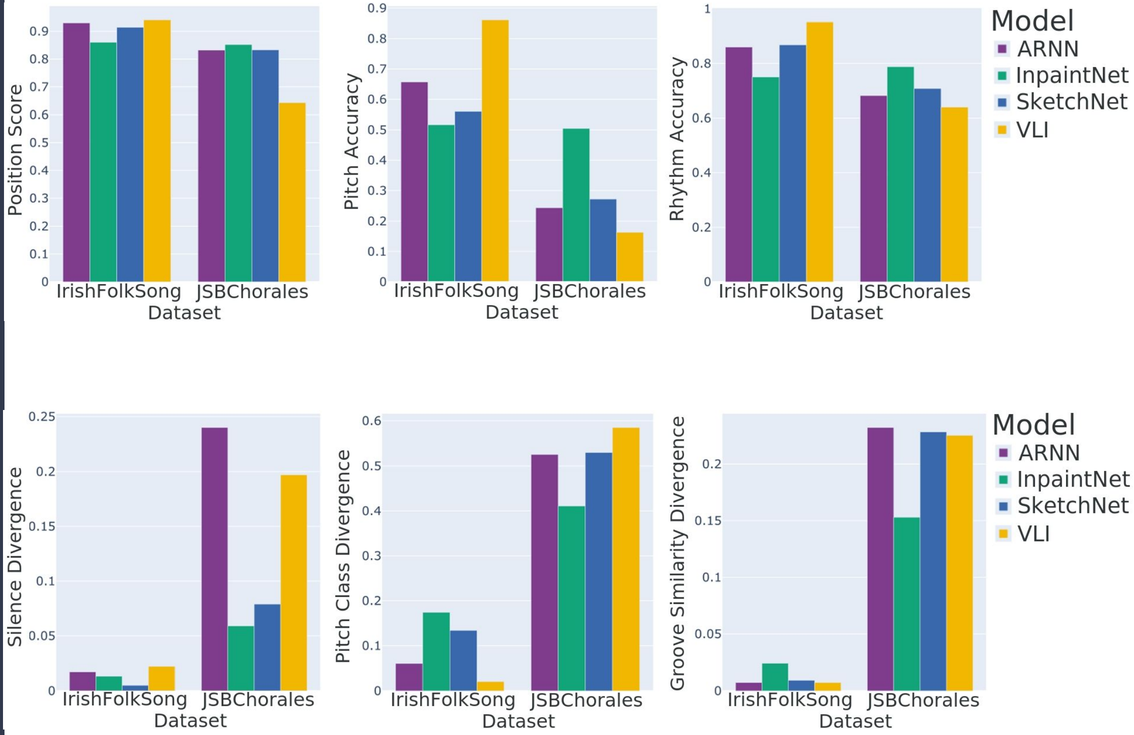
Model	$NLL \downarrow$	$pos_{F1} \uparrow$	$pAcc \uparrow$	$rAcc \uparrow$	$S_{div} \downarrow$	$H_{div} \downarrow$	$GS_{div} \downarrow$
Anticipation-RNN	0.453 (*0.662)	0.930	0.657	0.860	0.017	0.060	0.007
InpaintNet	0.487 (*0.662)	0.860	0.517	0.750	0.013	0.174	0.024
SketchNet	0.539 (*0.516)	0.914	0.560	0.868	**0.005**	0.134	0.009
VLI	0.059	**0.968**	**0.911**	**0.965**	0.015	**0.010**	**0.006**

Results – JSB Chorales

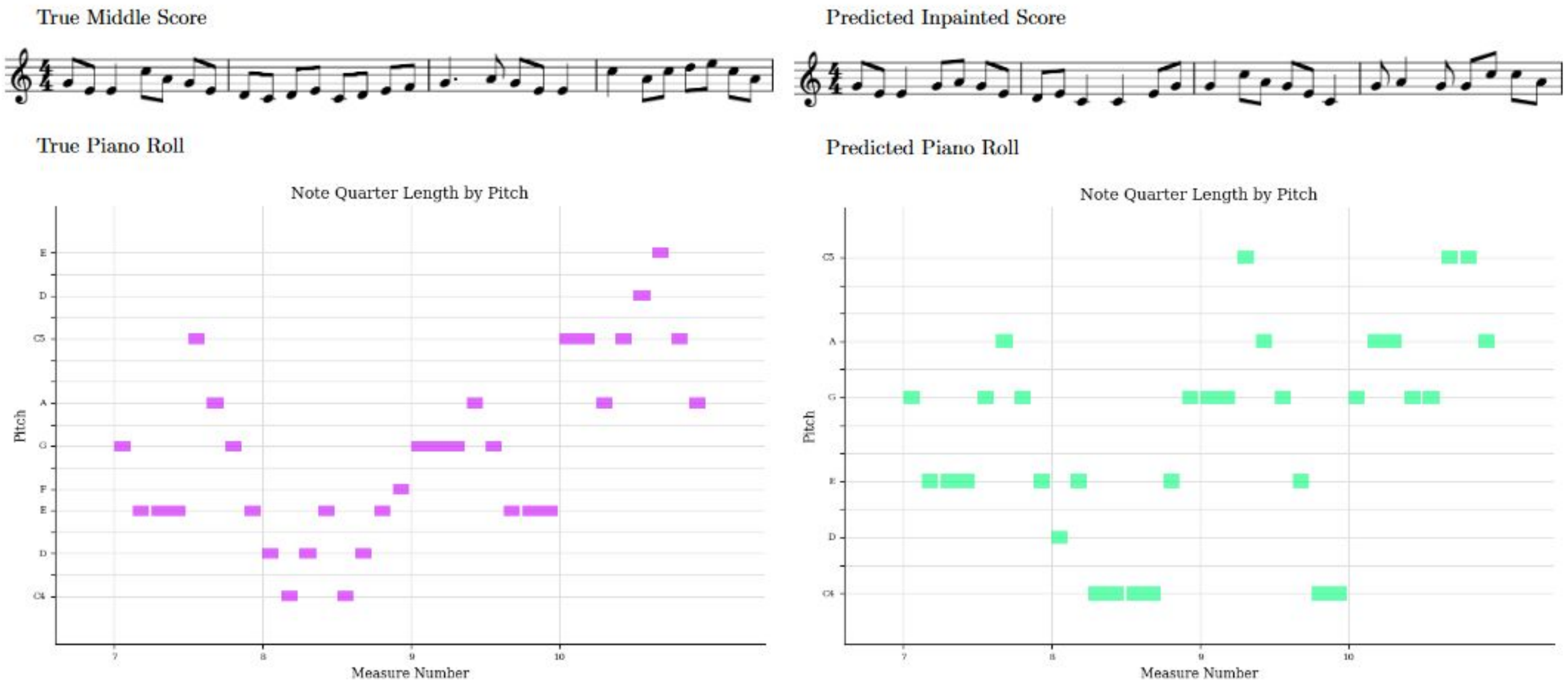
JSB Chorales Dataset ($\approx$ 2.4K samples)

Model	$NLL \downarrow$	$pos_{F1} \uparrow$	$pAcc \uparrow$	$rAcc \uparrow$	$S_{div} \downarrow$	$H_{div} \downarrow$	$GS_{div} \downarrow$
Anticipation-RNN	0.459	0.832	0.243	0.682	0.240	0.525	0.232
InpaintNet	0.327	**0.852**	**0.505**	**0.788**	**0.059**	0.411	**0.153**
SketchNet	0.605	0.833	0.272	0.708	0.079	0.529	0.228
VLI	1.053	0.827	0.283	0.747	0.087	**0.286**	0.306

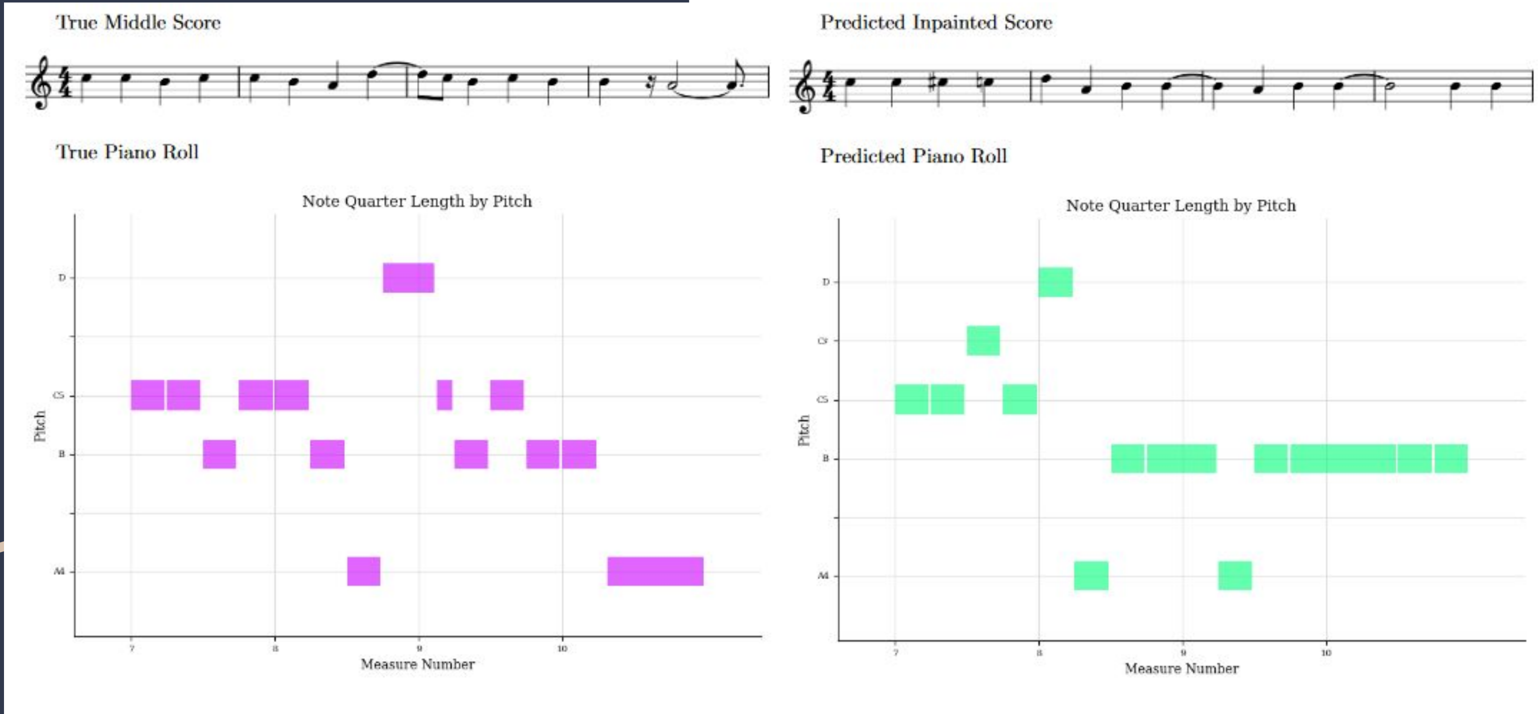
Results


Position Score
IrishFolkSong
JSBChorales
Dataset
Pitch Accuracy
Rhythm Accuracy
Model
ARNN
InpaintNet
SketchNet
VLI
Silence Divergence
Pitch Class Divergence
Groove Similarity Divergence

Results – IrishFolk



Results – JSB Chorales



Conclusions

Conclusions

- We proposed MUSIB, a new standardization framework and benchmark for musical score inpainting evaluation.
- We compiled, standardized and extended metrics to measure meaningful musical attributes.

Future Work

- Polyphonic music inpainting models
- Variable length infilling task
- Data augmentation strategies