SCORING TIME INTERVALS USING NON-HIERARCHICAL TRANSFORMER FOR AUTOMATIC PIANO TRANSCRIPTION

Yujia Yan, Zhiyao Duan
University of Rochester, Rochester, New York, USA,
yujia.yan@rochester.edu, zhiyao.duan@rochester.edu

ABSTRACT

The neural semi-Markov Conditional Random Field (semi-CRF) framework has demonstrated promise for event-based piano transcription. In this framework, all events (notes or pedals) are represented as closed time intervals tied to specific event types. The neural semi-CRF approach requires an interval scoring matrix that assigns a score for every candidate interval. However, designing an efficient and expressive architecture for scoring intervals is not trivial. This paper introduces a simple method for scoring intervals using scaled inner product operations that resemble how attention scoring is done in transformers. We show theoretically that, due to the special structure from encoding the non-overlapping intervals, under a mild condition, the inner product operations are expressive enough to represent an ideal scoring matrix that can yield the correct transcription result. We then demonstrate that an encoder-only non-hierarchical transformer backbone, operating only on a low-time-resolution feature map, is capable of transcribing piano notes and pedals with high accuracy and time precision. The experiment shows that our approach achieves the new state-of-the-art performance across all subtasks in terms of the F1 measure on the Maestro dataset.

1 Introduction

Automatic Music Transcription (AMT) transforms the audio signal of music performances into symbolic representations [1]. In this work, we focus on transcribing piano performance audio into its piano roll representation. [1] The piano roll representation, as formulated in [2], can be abstracted as consisting of sets of non-overlapping time intervals of the form [onset, offset], with each set corresponding to one particular event type, e.g., a specific note or pedal.

Recent strategies to handle the problem of outputting this structured representation fall into three main categories: 1) Keypoint detection and assembly: This approach involves identifying the onsets, offsets, and frame-wise activations of notes and then assembling these elements together with a handcrafted post-processing step. Examples include [3–5]; 2) Structured prediction with a probabilistic model: Models in this category use a probabilistic model to ensure the structure of the output to be sets of non-overlapping intervals, e.g., [2, 6, 7]; 3) Sequence-to-sequence (Seq2Seq) methods [2] : These methods, such as [8], treat music transcription as a machine translation problem, which translates audio to tokens that encode the target symbolic representation.

Our study focuses on the neural semi-Markov Conditional Random Field (semi-CRF) framework [2] from the second category, which directly models each music event (note or pedal) as a closed time interval associated with a specific event type. The approach employs a neural network to score interval candidates and uses dynamic programming to decode non-overlapping intervals. This framework eliminates the need for separate keypoint detection and assembly steps in the first category but outputs the events (intervals) in a single stage. Compared to other methods in the second category, e.g. [6, 7], it does not need hand-crafted state definitions and state transitions. Additionally, it benefits from optimal decoding in a non-autoregressive fashion as opposed to the slow autoregressive and suboptimal decoding in Seq2Seq methods (the third category).

This paper builds upon, simplifies, and improves the neural semi-CRF framework [2] for piano transcription. Our major contributions are as follows. First, we replace the original scoring module that assigns a score for every possible interval with a simpler and more efficient pairwise inner product operation. Specifically, we prove that due to the special structure of encoding non-overlapping intervals, under a mild condition, the inner product operation is expressive enough to represent an ideal scoring matrix that can yield the correct transcription decoding. Second, inspired by the resemblance between the proposed inner product operation and the attention mechanism in the transformer [9], we use the transformer architecture to produce the interval representation for inner product scoring. We demonstrate that an encoder-only non-hierarchical transformer backbone, operating only on a low-time-resolution feature map, is capable of transcribing notes with high accuracy and time precision. Third, we compare our method against state-of-the-art piano transcription systems on the Maestro v3 dataset, showing that our method establishes the new state of the art across all subtasks in terms of the F1 score.

[1] Code: `https://github.com/Yujia-Yan/Transkun`

[2] Strictly speaking, the Seq2Seq approach can also be categorized as a probabilistic model for structrued prediction. We isolate it here for simplifying the discussion.

2 Related Work

2.1 Neural Semi-CRF for Piano Transcription

Previous work of [2] introduced a neural semi-Markov Conditional Random Field (semi-CRF) framework for event-based piano transcription, where each event (note or pedal) is represented as a closed interval associated with a specific event type. The approach employs a neural network to score interval candidates and uses dynamic programming to decode non-overlapping intervals. After interval decoding, interval-based features are used to estimate event attributes, such as *MIDI velocity* and *refined onset/offset positions* [3] .

The neural semi-CRF can be viewed as a general output layer, similar to a softmax layer, but tailored for handling non-overlapping intervals. For a sequence of T frames, let $\mathcal{Y}$ denote a set of non-overlapping closed intervals. The semi-CRF layer for $\mathcal{Y}$ takes two inputs for each event type:

1. $score(i, j)$: A $T \times T$ triangular matrix that scores every candidate interval $[i, j]$ for inclusion in $\mathcal{Y}$. The diagonal values $score(i, i)$ represent single-frame events.
2. $score_\epsilon(i-1, i)$: A $(T-1)$-dimensional vector that assigns a score to every interval $[i-1, i]$ not covered by any interval in $\mathcal{Y}$, serving as an inactivity score.

Both $score(i, j)$ and $score_\epsilon(i-1, i)$ are computed using a neural network from the audio input $\mathcal{X}$. The total score for $\mathcal{Y}$, given $\mathcal{X}$, is:

$$\Phi(\mathcal{Y}|\mathcal{X}) = \sum_{[i,j]\in\mathcal{Y}} score(i, j) + \sum_{\substack{[i-1,i] \\ \text{not covered} \\ \text{in } \mathcal{Y}}} score_\epsilon(i-1, i). \tag{1}$$

For inference, maximum a posteriori (MAP) is used to infer the optimal set of non-overlapping intervals $\mathcal{Y}^*$:

$$\mathcal{Y}^* = \arg\max_{\mathcal{Y}} \Phi(\mathcal{Y}|\mathcal{X}). \tag{2}$$

For training, the maximum likelihood approach is used, with the conditional log-likelihood defined as:

$$\log p(\mathcal{Y}|\mathcal{X}) = \Phi(\mathcal{Y}|\mathcal{X}) - \log \sum_{\mathcal{Y}'} \exp \Phi(\mathcal{Y}'|\mathcal{X}). \tag{3}$$

Here, $\arg\max$ in Eq. (2), and the summation in the second term in Eq. (3) are over all possible sets of non-overlapping intervals. We refer the readers to [2] for algorithmic details.

To make predictions for all event types (88 keys + pedals), multiple instances of semi-CRF are used in parallel, each corresponding to a specific event type.

2.2 Vision Transformer and YOLOS

The Vision Transformer (ViT) [10] introduced a significant shift in computer vision, offering an alternative to traditional CNN models. ViT processes images as sequences of fixed-size patches using transformer layers [9], proving successful across various tasks. For end-to-end object detection, YOLOS [11] demonstrated a minimal, non-hierarchical encoder-only design that appends [DET] tokens (representing object slots) directly to image patch tokens as input to the transformer encoder. Our architecture adopts a similar encoder-only design for event-based music transcription.

[3] For dequantizing onset/offset positions from quantized positions.

3 Revisiting Interval Scoring for Semi-CRFs

The neural semi-CRF framework crucially relies on modeling the interval scoring matrix, $score(i, j)$, which assigns a score to each candidate interval. The size of the matrix, which grows quadratically with the sequence length, poses a challenge to designing an efficient and expressive model architecture. For this discussion, $score_\epsilon$ will be excluded due to its minimal impact on model performance from our observation and negligible modeling challenges.

3.1 Interval Scoring in [2]

In [2], a backbone model first transforms the input sequence $\mathcal{X} = [\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{T-1}]$ into a sequence of feature vectors $[\boldsymbol{h}_0, \ldots, \boldsymbol{h}_{T-1}]$. Each interval $[i, j]$ is scored by applying an MLP to features computed from the interval, with the output dimension being the number of event types. For simplicity, assuming only one event type to predict, the score is computed as

$$score(i, j) = \mathit{MLP}([\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_i \odot \boldsymbol{h}_j, \boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3]), \tag{4}$$

where $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ are feature vectors corresponding to the interval's onset and offset, $\odot$ denotes element-wise multiplication, and $\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3$ are the first, second, and third statistical moments over the interval $[i, j]$.

After producing the interval scoring matrices for all event types, a shallow CNN is applied, treating the interval endpoints as spatial coordinates and event types as channels. This refinement step slightly improves the result.

Directly computing Eq. (4) and the subsequent refinement step are memory intensive. The official implementation processes the scoring matrix in segments and applies gradient checkpointing during training, reducing peak memory usage at the cost of increased computational time. Consequently, the MLP and CNN layers' depth and width are constrained, potentially limiting the model's capacity and increasing susceptibility to local pattern overfitting.

3.2 Interval Scoring with Inner Product

We propose to use the following method for interval scoring:

$$score(i, j) = \frac{|j-i|}{\sqrt{D}} \langle \boldsymbol{q}_i, \boldsymbol{k}_j \rangle + b_i \delta(i, j), \tag{5}$$

where $\delta(i, j)$ is the Kronecker delta, which is 1 if $i = j$ and 0 otherwise. $\boldsymbol{q}_i \in \mathbb{R}^D$, $\boldsymbol{k}_i \in \mathbb{R}^D$ and $b_i \in \mathbb{R}$ are computed from the embedding vector $\boldsymbol{h}_i$ using a linear layer f:

$$[\boldsymbol{q}_i, \boldsymbol{k}_i, b_i] = f(\boldsymbol{h}_i). \tag{6}$$

The interval scoring matrix computed from Eq. (5) takes a low-rank plus diagonal structure. This method, termed **Scaled Inner Product Interval Scoring**, computes the score of an event as the scaled inner product between vectors $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$ representing the start and the end of the interval.

Despite its simplicity and resemblance to the attention mechanism in transformers, one question arises about the expressiveness of the inner product for capturing the transcription result. We answer this question by constructing a family of interval scoring matrices that can yield the correct decoded result, and then show that this family of matrices can be represented in the form of pairwise inner product under certain conditions.

Without loss of generality, we ignore the intervals of form $[i, i]$, which correspond to the diagonal values in the interval scoring matrix; they can be added back as diagonals as in Eq. (5). Additionally, since only the upper triangular part of the interval scoring matrix is used, we use the notation for a full matrix to simplify the derivation. We begin by defining a set of nonoverlapping closed intervals.

Definition 3.1. Let $\mathcal{Y}$ be a set of closed intervals defined on $\mathbb{N} \cap [0, T-1]$, i.e., T steps. It is a set of *non-overlapping* intervals if for any two intervals $[i_0, j_0] \in \mathcal{Y}$ and $[i_1, j_1] \in \mathcal{Y}$, $i_0 \geq j_1$ or $i_1 \geq j_0$, and, additionally, $\forall [i, j] \in \mathcal{Y}, i < j$.

Definition 3.2. An ideal interval scoring matrix for $\mathcal{Y}$ over T steps, i.e., $\boldsymbol{S}_{\mathcal{Y}} \in \mathbb{R}^{T \times T}$, is a matrix such that

$$\boldsymbol{S}_{\mathcal{Y}}(i, j) > 0, \qquad \forall [i, j] \in \mathcal{Y},$$
$$\boldsymbol{S}_{\mathcal{Y}}(i, j) = -\epsilon, \qquad \textit{otherwise}$$

where $\epsilon > 0$.

With an ideal scoring matrix $\boldsymbol{S}_{\mathcal{Y}}$, it is clear that the MAP decoding will yield $\mathcal{Y}$, since the exclusion of $\forall [i, j] \in \mathcal{Y}$ or the inclusion of $\forall [i, j] \notin \mathcal{Y}$ will decrease the total score.

Lemma 3.1. *The rank of an ideal interval scoring matrix $S_{\mathcal{Y}}$ for a set of non-overlapping intervals, $\mathcal{Y}$, is $M + 1$, where $M = |\mathcal{Y}|$, which is the number of intervals.*

Proof. By definition, the first column is $-\epsilon\mathbf{1}$, that is, $\forall i, \boldsymbol{S}_{\mathcal{Y}}(i, 0) = -\epsilon$. Subtracting the first column from all columns gives $\boldsymbol{S}'_{\mathcal{Y}}$ such that

$$\boldsymbol{S}'_{\mathcal{Y}}(i, j) > \epsilon, \qquad \forall [i, j] \in \mathcal{Y},$$
$$\boldsymbol{S}'_{\mathcal{Y}}(i, j) = 0, \qquad \textit{otherwise}$$

Given that no two non-zero entries in $\boldsymbol{S}'_{\mathcal{Y}}$ share a row or column (as per the definition of set of non-overlapping intervals), and there are M non-zero entries, the rank of $\boldsymbol{S}'_{\mathcal{Y}}$ is M. Since there are at most $T - 1$ non-overlapping intervals across T frames, we have $M \leq T - 1$, and the number of nonzero entries in $S'_{\mathcal{Y}}$ is smaller than or equal to $T-1$. As a result, $-\epsilon\mathbf{1}$ (T non-zeros) cannot be represented by a linear combination of other nonzero columns in $\boldsymbol{S}'_{\mathcal{Y}}$, therefore $rank(\boldsymbol{S}_{\mathcal{Y}}) = rank(\boldsymbol{S}'_{\mathcal{Y}}) + 1 = M + 1$. □

Theorem 3.2. *Let $\mathcal{Y}$ be a set of non-overlapping closed intervals over T steps, with cardinality M. An ideal interval scoring matrix $\boldsymbol{S}_{\mathcal{Y}}$ can be represented as pairwise inner products between two 1d sequences $(\boldsymbol{k}_i)_i$ and $(\boldsymbol{q}_i)_i$ of vectors:*

$$\boldsymbol{S}_{\mathcal{Y}}(i, j) = \langle \boldsymbol{q}_i, \boldsymbol{k}_j \rangle, \tag{7}$$

provided that $rank(\boldsymbol{Q}_{\mathcal{Y}}) > M$ and $rank(\boldsymbol{K}_{\mathcal{Y}}) > M$ where $\boldsymbol{Q}_{\mathcal{Y}} = [\boldsymbol{q}_0, \ldots, \boldsymbol{q}_{T-1}]$, and $\boldsymbol{K}_{\mathcal{Y}} = [\boldsymbol{k}_0, \ldots, \boldsymbol{k}_{T-1}]$.

Proof. By Lemma 3.1, the rank of $\boldsymbol{S}_{\mathcal{Y}}$ is $M + 1$. Then it directly follows the rank factorization of a matrix. □

Theorem 3.2 establishes a minimum rank requirement for $\boldsymbol{Q}_{\mathcal{Y}}$ and $\mathbf{K}_{\mathcal{Y}}$ to represent an ideal scoring matrix. This leads to two key observations:

1. The vector dimensions D of $\boldsymbol{k}_i$ and $\boldsymbol{q}_i$ must exceed the total number of intervals, $|\mathcal{Y}|$.
2. Consider a linear upsampling operator u_c, which is a special case of a 1-d transposed convolutional layer. It works by dividing each step of a vector sequence into c equal parts when the sequence is upsampled c times. Suppose we want to represent $\boldsymbol{Q}_{\mathcal{Y}}$ and $\boldsymbol{K}_{\mathcal{Y}}$ using low-resolution 1-d vector sequences: $\boldsymbol{Q}'_{\mathcal{Y}} = [\boldsymbol{q}'_0, \ldots, \boldsymbol{q}'_{T'-1}]$ and $\boldsymbol{K}'_{\mathcal{Y}} = [\boldsymbol{k}'_0, \ldots, \boldsymbol{k}'_{T'-1}]$ where $T' < T$, and this representation is achieved by applying u_c to $\boldsymbol{Q}'_{\mathcal{Y}}$ and $\boldsymbol{K}'_{\mathcal{Y}}$, resulting in $\boldsymbol{Q}_{\mathcal{Y}} = u_c(\boldsymbol{Q}'_{\mathcal{Y}})$, and $\boldsymbol{K}_{\mathcal{Y}} = u_c(\boldsymbol{K}'_{\mathcal{Y}})$, where $c = T/T'$ represents the upsampling factor. For this representation to be valid, the vector dimension D' for the low-resolution sequence, i.e., $\boldsymbol{q}'_i$ and $\boldsymbol{k}'_i$ should exceed $c|\mathcal{Y}|$.

These observations highlight that the dimensionality requirement depends solely on the count of intervals in $\mathcal{Y}$ and the downsampling (upsampling) factor $c = T/T'$ along the time axis. This analysis reveals sufficient conditions to guarantee the expressiveness of the inner product interval scoring method. From Theorem 3.2, by applying a scaling factor[4] and reintegrating diagonal terms, we can recover Eq. (5).

3.3 Comparison with Attention Mechanism

Comparing the neural semi-CRF with the inner product scoring to the attention mechanism reveals interesting parallels. Both of them have quadratic time complexity in the length of the input. The original score module, as in [2], resembles an additive attention mechanism, as introduced by [12]. However, attention mechanisms based on inner products [13] have become preferred for their simplicity and computational efficiency. Similarly, the proposed inner product scoring for neural semi-CRFs efficiently scores intervals. However, in contrast to attention mechanisms that score sequence positions and normalize posteriors for each position, neural semi-CRFs score intervals and normalize posteriors globally over sets of non-overlapping intervals.

The Transformer architecture can be viewed as inherently refining a sequential representation for inner product scoring. Inspired by these similarities, we utilize the transformer architecture to produce the 1-d sequence representations ($\boldsymbol{h}_i^{\text{eventType}}$) for each event type, termed **event tracks**, which will be used for inner product interval scoring.

4 Proposed System

Figure 1 summarizes the proposed system. The input is an oversampled log-mel spectrogram, as in [2]. The spectrogram is downsampled using 2-d strided convolutional layers, followed by the addition of spatial position embeddings (Section 4.2). Event tracks for all event types (notes and pedals) are initialized with their own spatial position embeddings and concatenated with the downsampled spectrogram representations. The concatenated features are processed by a transformer encoder. Subsequently, only the event track embeddings are upsampled using one 1-d transposed convolutional layer. The upsampled event tracks are used for inner product interval scoring (Eq. (5)) to generate

[4] Note that applying a length-dependent scaling on the ideal scoring matrix does not change the decoded result.

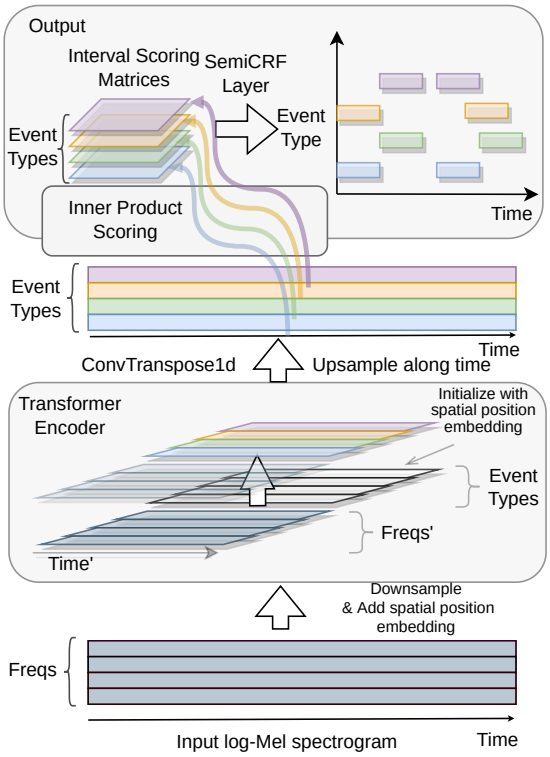


Figure 1: Overview of the proposed system. Inner product scoring follows Eq. (5).

interval scoring matrices, which are then fed to the neural semi-CRF layer for log-likelihood calculation or inference.

4.1 Rethinking Downsampling

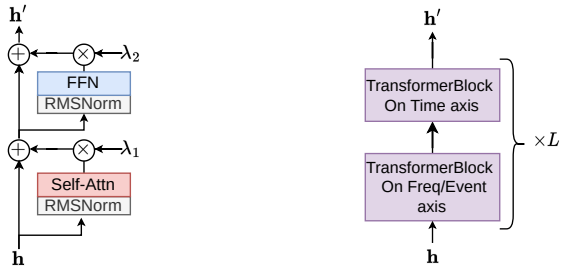
Existing studies on Vision Transformers (ViTs) demonstrate the effectiveness of a non-hierarchical design that uses highly downsampled, low-resolution feature maps even for tasks requiring dense predictions, e.g., [14], challenging the dominance of hierarchical models like UNET [15]. However, state-of-the-art (SOTA) piano transcription systems, including [2, 4, 5, 8], retain full resolution along the time axis. These approaches preserve the temporal detail of the input frames, but at the cost of increased training time and reduced model scalability.

This choice might be explained by concerns over losing temporal precision when locating events. However, we argue that the high dimensionality of the embeddings makes the low temporal resolution feature map still capable of processing with enough information.

In our approach, we use strided convolutional layers to downsample the input spectrogram, along both the time and frequency axes, transforming it from its original spatial dimensions (T, F) to a low-resolution feature map with dimensions $(T', F') = (\frac{T}{c_T}, \frac{F}{c_F})$. In line with the ViT literature, we refer to this reduced feature map as *patch embeddings for $c_T \times c_F$ patches*. The choice of patch size (c_T, c_F) may present a trade-off between computational efficiency and the model's capacity to capture dense events in the input spectrogram. As an initial exploration, we use a patch size of 8×4 to keep the training time within our expected range.

To upsample event tracks to the original temporal resolution of frames, we utilize a single transposed 1-d convolutional layer. We found that this simple upsampling layer efficiently prepares representations for inner product scoring at the desired resolution.

4.2 Transformer Encoder Architecture



(a) Transformer Block. (b) Encoder Layers

Figure 2: Building Blocks for the Transformer Encoder

Spatial Position Embedding. We use learnable Fourier features for spatial position embeddings [16] for both time-frequency representations with coordinates $(\mathit{frameIdx}, \mathit{freqIdx})$, and event tracks with coordinates $(\mathit{frameIdx}, \mathit{eventTypeIdx})$. This position embedding is chosen for its simplicity and broad compatibility with transformer architectures. Our formula differs slightly from [16] as we follow the formula in the original random Fourier features paper [17]. We compute the position embedding $\boldsymbol{y} \in \mathbb{R}^E$ from a multidimensional coordinate $\boldsymbol{x} \in \mathbb{R}^C$ as:

$$\boldsymbol{y} = g(\sqrt{\frac{2}{B}} \cos(\boldsymbol{W}_r \boldsymbol{x} + \boldsymbol{b})), \tag{8}$$

where $\boldsymbol{W}_r$ is a learnable matrix $\mathbb{R}^{B \times C}$, initialized from $\mathcal{N}(0, \gamma^{-2})$; B is the dimension for the Fourier features; γ is a hyperparameter; $\boldsymbol{b} \in \mathbb{R}^B$ is the learnable bias term, initialized from $\mathcal{U}(-\pi, +\pi)$; $g : \mathbb{R}^B \to \mathbb{R}^E$ is a two-layer perceptron. This position embedding functions like an MLP that takes coordinates as input, with the first nonlinearity being a scaled cosine function.

The Transformer Encoder Layer. Figure 2a illustrates the basic transformer block. This block first applies *RMSNorm* [18] before the self-attention and feed-forward layers. To enhance training stability, we use ReZero [19] which applies a learnable scaling factor λ, initially set to 0.01, before adding to the skip connection. As in Figure 2b, for reducing computational cost, we alternate attention within each transformer block along the time and frequency/eventType axes; similar ideas are often used for efficient transformer architectures [20–22].

4.3 Segment-Wise Processing

Longer audio is transcribed using segments with 50% overlap. Unlike [2], which discards events that exceed the segment boundary during training, we truncate such events to fit within the segment. We introduce two binary attributes, *hasOnset* and *hasOffset*, to indicate whether an event's onset or offset has been truncated.

For each event type within a segment, decoding starts from either: (1) the current segment's boundary, or (2) the offset of the last event in the result set with *hasOffset* = *true*, whichever is later. Events decoded in the current

segment are then processed as follows: (1) non-overlapping events with *hasOnset* = *true* are directly added to the result set; (2) for events overlapping with the last event of the same type in the result set: if the current event has *hasOnset* = *true*, it replaces the last event [5] ; otherwise, the two events are merged.

4.4 Attribute Prediction

Attributes associated with each event include *velocity*, *refined onset/offset positions* (for dequantizing frame positions), and the binary flags *hasOnset* and *hasOffset*. To predict these attributes for an event extracted from the event track $(\boldsymbol{h}_i^{\text{eventType}})_{i=0}^{T-1}$, e.g., $[a, b]$, we use a two-layer MLP that takes $\boldsymbol{h}_a^{\text{eventType}}$ and $\boldsymbol{h}_b^{\text{eventType}}$ as input. The MLP outputs the parameters of the probability distributions for each attribute. Specifically, *velocity* $\in \{0 \ldots, 127\}$ is modeled as a categorical distribution, *refined onset/offset positions* $\in (-0.5, 0.5)$ are modeled as continuous Bernoulli distributions [23] shifted by -0.5, and *hasOnset*/*hasOffset* $\in \{0, 1\}$ are modeled as Bernoulli distributions.

5 Experiment

5.1 Dataset

Maestro v3.0.0 [24]. This dataset contains about 200 hours of piano performances, including audio recordings and corresponding MIDI files captured using Yamaha Disklavier pianos. We use the standard train/validation/test splits.

MAPS [25]. The MAPS dataset includes both synthesized and real piano recordings, with the real recordings captured by MIDI playback on Yamaha Disklavier. We evaluate our model on the Disklavier subset (ENSTDkAm/MUS and ENSTDkCl/MUS) of the MAPS dataset, which consists of 60 recordings and is commonly used for cross-dataset evaluation. However, we discovered systematic alignment issues in the ground-truth annotations for both notes and pedals, affecting both onset and offset locations. Onset alignment issues have been previously reported in [26] but are not widely known in the community [6] .

SMD [27]. Similar to Maestro dataset, the SMD dataset was created by recording human performance on a Yamaha Disklavier. The dataset contains 50 recordings. We found that the onset annotations in SMD are better aligned compared to MAPS, but the offset annotations suffer from similar biases.

5.2 Model Specification

The key model specifications are summarized in Table 1. Training takes about 6 days on 2 *NVIDIA RTX 4090*.

[5] For overlapping events between segments: (1) The first event must have *hasOffset* = *false*. (2) A continuing second event must have *hasOnset* = *false*. (3) If the second event's *hasOnset* = *true*, the first event is replaced by the second event as it's not supported by the second.

[6] A piece-dependent onset latency around 15 ms has been previously discussed in [26]. Due to the electro-mechanical playback mechanism, this latency could also be note/pedal dependent. Offset deviation (up to approximately 70 ms) appears more complex and may be influenced by pedal-/note-dependent mechanical latency or undocumented specific piano model's response to non-binary pedal values.

Input Mel Spectrogram	sr: 44100 Hz, hop: 1024, window size: 4096, subwindows:5, mels: 229, freq: 30-8000 Hz, segment: 16s,
Patch	shape: 8×4, embeding size: 256
Strided Conv. Layers for Downsampling	initial proj. size: 64, added with freq. embeddings out: [128, 256, 256, 256], kernel size: 3, stride: [(2,1), (2,2), (2,2), (1,1)], Each followed by GroupNorm, groups = 4, and GELU (except for the last conv.)
Position Embedding	$\gamma = 1$, $\lvert B \rvert = 256$, MLP hidden size 1024
Transformer Encoder	8 heads, 6 layers (=12 blocks), FNN size: 1024
Upsampling	1d. transposed conv, out: 128, kernel size:8, stride:8
Attribute Prediction	two layer MLP, hidden size: 512, dropout 0.1
Batch Size	12
Optimizer	Adabelief [28], maximum learning rate: $4\mathrm{e}{-4}$
Weight Decay	$1\mathrm{e}{-2}$, excluding bias, norm., and pos. embedding
Learning Rate Schedule	500k iterations, 5% warm-up phase, cosine anneal.
Gradient Clipping	Clipping norms at 80% quantile of past 10,000 iterations

Table 1: Model Specification.

5.3 Evaluation Metrics

We compute precision, recall, and f1 score averaged over recordings for both activation level (from [2], equivalent to frame level with infinitesimal hop size), and note level metrics (*Note Onset*, *Note w/Offset*, and *Note w/Offset & Vel.*, using *mir_eval* [29], default settings). All metrics are directly computed from transcribed MIDIs. For details on these metrics, readers can refer to the supplementary material of [2], and the documentation of *mir_eval* [30].

For MAPS and SMD, we only report activation-level and onset-only note-level metrics due to the ground-truth alignment issues discussed in Section 5.1. These issues make offset-related metrics unreliable for these datasets.

5.4 Results

Our results on the Maestro v3 test set are presented in Table 2. The proposed model achieves state-of-the-art performance across all metrics in terms of f1 score, surpassing previous methods by a significant margin. We also report results for soft pedal transcription which has not been previously explored. The low event-level metrics suggest that accurately determining soft pedal onset and offset times is more challenging than for notes and sustain pedals. We conjecture this is because soft pedals are typically engaged for longer durations and appear significantly less frequently in the dataset than sustain pedals.

Scoring Methods Comparison. We conducted an ablation study to compare our proposed inner product scoring with the more complex scoring method from [2]. We trained a model with an identical architecture but replaced the inner product scoring with the scoring module from [2]. To ensure a fair comparison, we adjusted the hidden sizes of the scoring module to keep the training time for a single iteration within a factor of two of our proposed system. Specifically, all event tracks were projected to a single sequence with a dimension of 512, and the hidden size of the scoring module was set to 512. As shown in Table 2, our inner product scoring outperforms the more complex scoring method, demonstrating its effectiveness and efficiency.

Furthermore, we compared two variants of the inner product scoring: a linear layer and an MLP for computing the $\boldsymbol{k}/\boldsymbol{q}/b$ vectors (f in Eq. (6)). The results demonstrate that the linear layer yields better performance than the MLP.

Method	# Param	Activation			Note Onset			Note w/ Offset			Note w/ Offset & Vel.		
		P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)
Notes													
SemiCRF [2]	9.8M	93.79	88.36	90.75	98.69	93.96	96.11	90.79	86.46	88.42	89.78	85.51	87.44
hFT, reported in [5]	5.5M	92.82	93.66	93.24	**99.64**	95.44	97.44	92.52	88.69	90.53	91.43	87.67	89.48
hFT [5] [8] .	5.5M	95.37	90.82	92.93	**99.62**	95.41	97.43	92.22	88.40	90.23	91.21	87.44	89.24
Ours with scoring method in [2]	11.0M	93.79	92.40	93.06	98.61	95.92	97.23	91.69	89.23	90.43	91.08	88.64	89.83
Ours with MLP $\boldsymbol{kqb}$ mapping	13.0M	95.66	94.79	95.20	99.54	96.91	98.19	94.39	91.92	93.12	93.84	91.40	92.59
Ours w/o incomplete events	12.9M	**93.76**	94.46	95.07	99.56	97.10	98.30	**94.66**	92.36	93.48	**94.12**	91.83	**92.95**
Ours	12.9M	95.75	**95.01**	**95.35**	99.53	**97.16**	**98.32**	94.61	**92.39**	**93.48**	94.07	**91.87**	92.94
Sustain Pedals													
Kong et al., reported in [4]	20.2M	94.30	94.42	94.25	91.59	92.41	91.86	86.36	87.02	86.58	-	-	-
Kong et al. [4][8] [9]	20.2M	94.14	94.29	94.11	77.43	78.19	77.71	73.56	74.21	73.81	-	-	-
SemiCRF [2]	9.8M	95.17	88.33	90.98	82.18	75.81	78.52	78.75	72.74	75.30	-	-	-
Ours w/o incomplete events	12.9M	**96.69**	92.92	94.47	**89.10**	83.96	86.28	**86.33**	81.40	83.63	-	-	-
Ours	12.9M	96.67	**94.46**	**95.40**	88.96	**84.22**	**86.37**	86.19	**81.66**	**83.71**	-	-	-
Soft Pedals													
Ours w/o incomplete events	12.9M	74.41	28.77	36.54	20.24	9.08	11.69	17.19	7.51	9.76	-	-	-
Ours	12.9M	86.42	83.12	84.09	24.32	17.39	19.46	18.51	13.40	15.06	-	-	-

Table 2: Transcription Result on Maestro v3.0.0 Dataset Test Split.

Interestingly, this aligns with how $\boldsymbol{k}$ and $\boldsymbol{q}$ are computed in transformers.

Effect of omitting incomplete events. We found that omitting steps of handling incomplete events at segment boundaries (Section 4.3) only cause noticeable performance impact for pedals, particularly the soft pedal (Table 2). This can be explained by the fact that pedal events, especially soft pedals, can often exceed the segment length, while notes are normally shorter than the segment length we choose.

Results on MAPS/SMD. We evaluated our model on the MAPS dataset using three different ground-truth annotations: (1) Original, (2) Ad hoc Align, where the median deviation from the initial evaluation is subtracted from all notes for each piece and then re-evaluated, and (3) Cogliati, which subtracted a latency value per recording for ENST-DkCL as provided by [26]. For the SMD dataset, only the original annotation is used. Table 3 presents the results.

All methods exhibit low activation-level F1 scores on both MAPS and SMD. Using the onset-corrected annotation (Cogliati) on MAPS increases the onset F1 score but degrades the activation-level F1 score due to the uncorrected offset biases. In fact, the Cogliati annotation achieves similar or lower activation-level F1 scores compared to all listed methods when evaluated against the original annotation.

For note onset, all methods on SMD achieve F1 scores close to those evaluated on Maestro. However, performance decreases significantly on MAPS, even with corrected annotations, suggesting that the alignment issue may be more complex than a simple global offset.

Notably, the corrected annotations can lead to different conclusions compared to the original annotation. For example, while the data-augmented Onsets&Frames model achieves a higher note onset F1 score than hFT using the original annotation, it scores lower than hFT when evaluated using the ad hoc correction and the Cogliati annotation.

These observations highlight the need for caution when evaluating models on datasets created using mechanisms that may involve systematic biases, e.g., electromechanical playback. Despite these complications, our proposed system, with or without data augmentation [7] , achieves the highest note onset F1 score among the compared methods on both SMD and MAPS with Ad hoc/Cogliati correction.

			Activation			Note Onset		
Method	Dataset	Groudtruth	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)
Onsets	MAPS	Original	90.27	80.33	84.87	87.40	85.56	86.41
&Frames [24]	MAPS	Ad hoc Align	90.50	80.53	85.08	88.79	86.93	87.78
w. Data Aug.[8]	MAPS	Cogliati	64.75	82.83	71.60	87.57	84.97	86.19
hFT [5].[8]	MAPS	Original	91.53	71.03	79.81	84.63	85.75	85.13
	MAPS	Ad hoc Align	91.77	71.25	80.04	87.32	88.48	87.84
	MAPS	Cogliati	68.83	74.07	70.24	89.94	90.10	89.97
	SMD	Original	54.01	93.35	66.23	98.71	95.58	97.09
Ours	MAPS	Original	88.41	82.29	85.08	84.31	88.10	86.10
	MAPS	Ad hoc Align	88.69	82.57	85.36	86.63	90.53	88.47
	MAPS	Cogliati	65.74	84.69	72.78	89.60	91.39	90.44
	SMD	Original	52.54	96.23	65.39	98.10	97.67	97.88
Ours	MAPS	Original	94.11	84.63	89.00	92.11	88.78	90.38
w. Data Aug.	MAPS	Ad hoc Align	94.35	84.84	89.22	94.21	90.76	92.41
	MAPS	Cogliati	67.77	87.39	75.03	94.66	91.43	92.98
	SMD	Original	52.83	96.72	65.86	99.75	97.71	98.71
Between Ground Truths								
Cogliati [26]	MAPS	Original	98.86	69.22	80.17	100	100	100

Table 3: Transcription Result on MAPS and SMD. See Text for discussion of dataset issues.

6 Conclusion

This paper introduces a simple and efficient method for scoring time intervals using scaled inner product operations for the neural semi-CRF framework for piano transcription. We demonstrate that the proposed scoring method is not only simple and efficient but also theoretically expressive for yielding the correct transcription result. Inspired by the similarity between the proposed scoring method and the attention mechanism, we employ a non-hierarchical, encoder-only transformer backbone to produce event track representations. Our method achieves state-of-the-art performance on the Maestro dataset across all subtasks. Due to resource constraints, we have not evaluated the effect of patch and embedding sizes, which is left for future work. Additionally, future research could explore more advanced transformer architectures, investigate the interaction between transformer architecture and the neural semi-CRF layer, and extend the approach to other instruments and multi-instrument music transcription tasks.

[7] Data augmentation: pitch shifting ± 20 cents, adding noise from [31], applying randomized 8 band EQ and impulse response from [32].

[8] Use their provided code and pretrained weights. Recomputed from transcribed MIDIs.

[9] Previous SOTA for sustain pedals. Their released code indicates a 200 ms onset tolerance for pedal evaluation, contrary to the reported 50 ms in their paper. Here, we use a 50 ms onset tolerance, which explains the large discrepancy between the numbers here and their reported results.

7 Acknowledgement

This work is supported in part by National Science Foundation (NSF) grants 1846184 and 2222129.

8 References

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, pp. 20–30, 2019.

[2] Y. Yan, F. Cwitkowitz, and Z. Duan, "Skipping the frame-level: Event-based piano transcription with neural semi-CRFs," in *Advances in Neural Information Processing Systems*, 2021.

[3] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[4] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-resolution piano transcription with pedals by regressing onset and offset times," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3707–3717, 2020.

[5] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W. Liao, and Y. Mitsufuji, "Automatic piano transcription with hierarchical frequency-time transformer," in *International Society for Music Information Retrieval Conference*, 2023.

[6] R. Kelz, S. Böck, and G. Widmer, "Deep polyphonic adsr piano note transcription," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 246–250.

[7] T. Kwon, D. Jeong, and J. Nam, "Polyphonic piano transcription using autoregressive multi-state note model," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018.

[8] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, "Sequence-to-sequence piano transcription with transformers," in *International Society for Music Information Retrieval Conference*, 2021.

[9] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Neural Information Processing Systems*, 2017.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[11] Y. Fang, B. Liao, X. Wang, J. Fang, J. Qi, R. Wu, J. Niu, and W. Liu, "You only look at one sequence: Rethinking transformer in vision through object detection," in *Neural Information Processing Systems*, 2021.

[12] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[13] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[14] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *European conference on computer vision*, 2022, pp. 280–296.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of the 18th International Conference on Medical Image Computing and Computer-assisted Intervention (MICCAI)*, 2015, pp. 234–241.

[16] Y. Li, S. Si, G. Li, C.-J. Hsieh, and S. Bengio, "Learnable fourier features for multi-dimensional spatial positional encoding," in *Advances in Neural Information Processing Systems*, 2021.

[17] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Neural Information Processing Systems*, 2007.

[18] B. Zhang and R. Sennrich, "Root Mean Square Layer Normalization," in *Advances in Neural Information Processing Systems*, 2019.

[19] T. Bachlechner, B. P. Majumder, H. Mao, G. Cottrell, and J. McAuley, "Rezero is all you need: fast convergence at large depth," in *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, vol. 161, 2021, pp. 1352–1361.

[20] J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans, "Axial attention in multidimensional transformers," *ArXiv*, vol. abs/1912.12180, 2019.

[21] N.-C. Ristea, R. T. Ionescu, and F. S. Khan, "Septr: Separable transformer for audio spectrogram processing," in *Proceedings of INTERSPEECH*, 2022, pp. 4103–4107.

[22] W.-T. Lu, J.-C. Wang, Q. Kong, and Y.-N. Hung, "Music source separation with band-split rope transformer," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 481–485.

[23] G. Loaiza-Ganem and J. P. Cunningham, "The continuous bernoulli: fixing a pervasive error in variational autoencoders," in *Advances in Neural Information Processing Systems*, 2019.

[24] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *International Conference on Learning Representations*, 2019.

[25] V. Emiya, N. Bertin, B. David, and R. Badeau, "Maps - a piano database for multipitch estimation and automatic transcription of music," 2010.

[26] A. Cogliati, Z. Duan, and B. Wohlberg, "Context-dependent piano music transcription with convolutional sparse coding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2218–2230, 2016.

[27] M. Müller, V. Konz, W. Bogler, and V. Arifi-Müller, "Saarland music data (SMD)," in *Late-Breaking and Demo Session of the 12th International Conference on Music Information Retrieval (ISMIR)*, 2011.

[28] J. Zhuang, T. Tang, Y. Ding, S. Tatikonda, N. Dvornek, X. Papademetris, and J. Duncan, "Adabelief optimizer: Adapting stepsizes by the belief in observed gradients," *Conference on Neural Information Processing Systems*, 2020.

[29] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "Mir_eval: A transparent implementation of common mir metrics," in *International Society for Music Information Retrieval Conference*, 2014.

[30] C. Raffel, "mir_eval documentation on transcription metrics," https://craffel.github.io/mir_eval/#id46, Accessed: 10-April-2024.

[31] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, 2015, pp. 1015–1018.

[32] "Echo thief," http://www.echothief.com/, accessed: 2023-05-07.