Snapping Matters: Context-Aware Onset Refinement for Automatic Music Transcription

Abhirup Saha
International Audio Laboratories Erlangen
abhirup.saha@audiolabs-erlangen.de

Hans-Ulrich Berendes
International Audio Laboratories Erlangen
hans-ulrich.berendes@audiolabs-erlangen.de

Meinard Müller
International Audio Laboratories Erlangen
meinard.mueller@audiolabs-erlangen.de

Ben Maman
International Audio Laboratories Erlangen
ben.maman@audiolabs-erlangen.de

ABSTRACT

Precise note-level annotations are critical for training automatic music transcription (AMT) systems, in particular note-onset labels, which form a core component of many recent AMT systems. However, high-quality annotations for real-world recordings are scarce. Sequence-level score–audio alignment methods such as dynamic time warping provide only coarse correspondence, making a local refinement step necessary. This refinement step, known as snapping, adjusts aligned score onsets using peaks in a neural onset posteriorgram and often determines whether weakly aligned score–audio pairs become usable training data at all. Despite its practical importance, snapping is typically treated as a simple post-processing heuristic and implemented with greedy local decisions. We present a systematic analysis of snapping strategies for training instrument-agnostic transcribers, demonstrating that snapping is essential for learning from weakly aligned data. Building on this, we formulate snapping as a per-pitch assignment problem and solve it via bipartite graph matching, yielding context-aware onset decisions under overlapping refinement windows and uncertain initial alignments. Extensive cross-dataset experiments across piano, chamber, and orchestral recordings show improved onset alignment and transcription accuracy over greedy snapping, with gains increasing for wider snapping windows and coarser initial alignments. Qualitative examples are provided on our project page: `https://abhirupsaha8.github.io`

1. INTRODUCTION

Automatic music transcription (AMT), the task of converting audio recordings into symbolic note representations, remains a central problem in music information retrieval (MIR). While piano transcription has progressed rapidly, reliable onset and pitch estimation for non-piano instruments and multi-instrument mixtures is still difficult due to timbral variability, polyphony, and the limited availability of precisely annotated training data. In particular, modern data-driven AMT models rely on supervision with accurate note onsets, yet onset-level ground truth is hard to obtain for real recordings outside controlled acquisition settings.

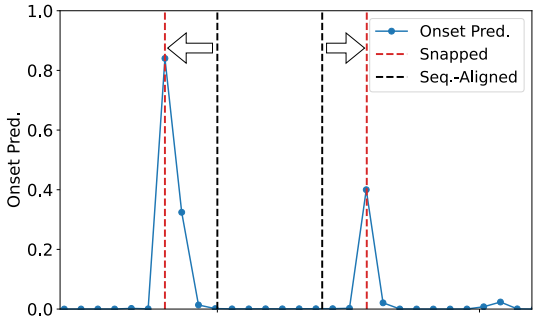

Figure 1. Snapping for a single pitch. The black dashed line ("Seq.-Aligned") shows onset timings from a sequence-level alignment (e.g., DTW), while the red dashed line ("Snapped") indicates the adjusted onsets after snapping. The blue line denotes activations in an onset posteriorgram.

A common workaround is to derive training labels by aligning a musical score (or a MIDI file) to the audio. Sequence-level alignment methods such as dynamic time warping (DTW) can provide robust coarse correspondences, for example based on chroma- or onset-related features [1, 2, 3]. However, even strong sequence-level alignments typically do not yield onset-accurate labels. Expressive timing deviations and local asynchronies (e.g., arpeggiation) can cause score events that are simultaneous in the score to be realized at different times in the audio, or vice versa. Consequently, directly transferring score onsets through a warping path often yields labels that are insufficiently precise for effective AMT training.

To bridge this gap, many recent pipelines adopt a two-stage strategy [7, 4, 14]: first compute a global, sequence-level score–audio alignment, then refine individual note onsets using neural onset activations, a step commonly referred to as snapping. In snapping, each score onset timing is refined within a local temporal window to a nearby peak in a learned onset posteriorgram, as depicted in Figure 1. Despite its practical success and growing adoption, snapping is typically implemented using local greedy decisions or simplified peak-picking heuristics. This is fragile when refinement windows overlap, because locally optimal choices can conflict, leading to duplicate assignments, missed peaks, or reduced global consistency. The problem becomes particularly pronounced when the initial alignment

is coarse and wider refinement windows are required.

This paper revisits *snapping* as an essential component of AMT label generation and asks how onset refinement should be formulated and solved to remain reliable under realistic alignment uncertainty. Our key idea is to treat snapping not as independent peak picking, but as a structured decision problem. For each pitch, we assign note events to candidate audio frames within admissible time windows, maximizing onset-posterior evidence while enforcing one-to-one consistency. This yields a principled and scalable refinement step that remains robust in the presence of overlapping windows and coarse initial alignments. The main contributions of this paper are as follows:

- We clearly distinguish *sequence-level alignment*, which defines a warping relation between timelines, from *note-onset-level alignment*, which maps discrete events to precise times, and we position snapping as the refinement step that connects the two.
- We formulate snapping as a per-pitch assignment problem based on bipartite graph matching, explicitly handling overlapping windows and global consistency beyond greedy heuristics.
- We provide a systematic cross-dataset evaluation analyzing how solver choice, window size, and initial alignment quality affect both alignment accuracy and downstream transcription performance, including robustness under very coarse initial alignments.
- We show that graph-based snapping yields consistent improvements in transcription accuracy across diverse datasets and becomes increasingly beneficial as refinement windows grow, which is precisely the regime required when labels are only weakly aligned.

This establishes snapping as a principled and tunable refinement step that bridges robust sequence alignment and onset-accurate supervision, enabling more reliable training for instrument-agnostic AMT under realistic conditions.

The remainder of this paper is organized as follows. Section 2 reviews related work on AMT and audio–score alignment. Section 3 introduces our method, including formal definitions of sequence- and onset-level alignment and a mathematical formulation of graph-based snapping. Section 4 defines the transcription and alignment tasks considered in this work. Section 5 presents experiments and evaluation procedures. Section 6 summarizes the findings and outlines directions for future work.

2. RELATED WORK

For an overview of automatic music transcription, see [8]. Early work focused on piano transcription, initially using non-negative matrix factorization (NMF) [9], before giving way to data-driven approaches. Key advances such as Onsets and Frames [10] and the MAESTRO dataset [11] enabled high-quality polyphonic piano transcription [12] based on note-onset detection.

For multi-instrument datasets [13], score–audio alignment via DTW [2] is common but prone to errors. Recent work [4] shows that onset features from pre-trained neural transcription models enable more precise score–audio alignment, improving sequence-level robustness [3] and allowing accurate note-onset annotations via snapping. A similar strategy was later used to construct guitar datasets [5, 14]. More recent work further simplifies alignment using histogram-based top-K peak picking [6].

While initially applied to piano and other instruments with sharp onsets such as guitar, onset-based transcription has been extended in recent work to strings, winds, and multi-instrument settings, both instrument-sensitive [16, 17, 4] and instrument-agnostic [18, 6].

3. METHOD

In this section, we describe our proposed method in detail. In Section 3.1, we formally define the notions of sequence-level and note-onset-level alignment, highlighting the differences between them. Then, in Section 3.2, we formulate how snapping—the refinement of a sequence-level alignment into an onset-level alignment—can be performed optimally using bipartite graph matching based on a learned note-onset posteriorgram.

3.1 Sequence-level vs. Note-Onset-level Alignment

Given an audio recording of a musical piece and a corresponding digital representation of the score, such as a piano roll or a MIDI file, we define two discrete timelines: the score timeline

$$\mathcal{T}_{\mathrm{s}} = [1 : T_{\mathrm{s}}] \coloneqq \{1, 2, \ldots, T_{\mathrm{s}}\} \tag{1}$$

of length $T_{\mathrm{s}} \in \mathbb{N}$, and the audio (physical) timeline

$$\mathcal{T}_{\mathrm{a}} = [1 : T_{\mathrm{a}}] \coloneqq \{1, 2, \ldots, T_{\mathrm{a}}\} \tag{2}$$

of length $T_{\mathrm{a}} \in \mathbb{N}$. We further define a set of possible pitches $\mathcal{P}$ of size P, and a set of possible instruments $\mathcal{I}$ of size I:

$$\mathcal{P} = \{p_1, \ldots, p_P\}, \quad \mathcal{I} = \{i_1, \ldots i_I\}. \tag{3}$$

The musical score can then be represented as a set of notes

$$\mathcal{N} \subset \mathcal{P} \times \mathcal{T}_{\mathrm{s}} \times \mathcal{T}_{\mathrm{s}} \times \mathcal{I} \tag{4}$$

of size N. Each note $n \in \mathcal{N}$ is represented as a tuple

$$n = (p,\, t^{\mathrm{on}},\, t^{\mathrm{off}},\, i) \tag{5}$$

where $p \in \mathcal{P}$ denotes the pitch, $t^{\mathrm{on}} \in \mathcal{T}_{\mathrm{s}}$ and $t^{\mathrm{off}} \in \mathcal{T}_{\mathrm{s}}$ are the note onset and offset positions on the score timeline, and $i \in \mathcal{I}$ is the instrument class.

A *sequence-level alignment* is defined as a function

$$\mathcal{W} : \mathcal{T}_{\mathrm{s}} \to \mathcal{T}_{\mathrm{a}}, \tag{6}$$

which maps each score-frame index to a corresponding audio-frame index. Such alignments are typically estimated from frame-level similarity using DTW or similar methods.

A *note-onset-level alignment* is a function

$$\mathcal{M} : \mathcal{N} \to \mathcal{T}_{\mathrm{a}}, \tag{7}$$

mapping each note in the musical score $\mathcal{N}$ to its corresponding onset time on the audio timeline $\mathcal{T}_{\mathrm{a}}$. In this work, we

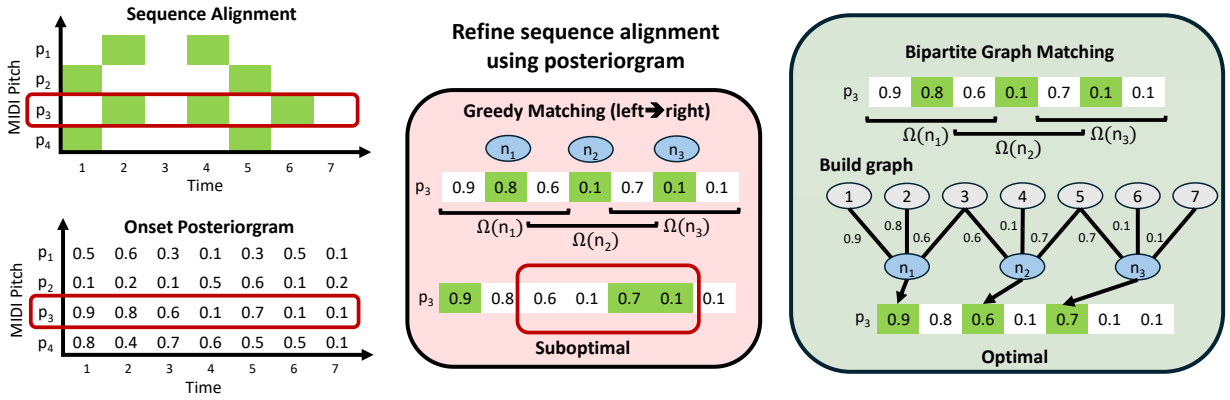

Figure 2. Snapping using greedy matching, compared to snapping based on bipartite graph matching.

focus on onset detection and consider only the onset component of each note, although a more general note-level alignment could also include note offsets.

A sequence-level alignment $\mathcal{W}$ induces an onset-level alignment $\mathcal{M}^{\text{seq}} : \mathcal{N} \to \mathcal{T}_{\text{a}}$ defined by:

$$\mathcal{M}^{\text{seq}}(n) = \mathcal{M}^{\text{seq}}((p, t^{\text{on}}, t^{\text{off}}, i)) = \mathcal{W}(t^{\text{on}}). \quad (8)$$

When aligning a musical score to an audio recording of a performance of the same score, a good sequence-level alignment does not necessarily guarantee a good onset-level alignment. For example, if two note onsets occur within the same frame on the score timeline but in different frames on the audio timeline (as in arpeggios or similar expressive timing variations), then any onset-level alignment induced directly from the sequence-level alignment will necessarily assign an incorrect onset time to at least one of the notes.

Denoting the true (latent) onset-level alignment by

$$\mathcal{M}^* : \mathcal{N} \to \mathcal{T}_{\text{a}}, \quad (9)$$

and the onset-level alignment induced by the sequence-level alignment by $\mathcal{M}^{\text{seq}}$, we assume in the following the existence of a uniform error bound B such that

$$|\mathcal{M}^{\text{seq}}(n) - \mathcal{M}^*(n)| \leq B, \qquad \text{for all } n \in \mathcal{N}. \quad (10)$$

Improved sequence-level alignment results in a smaller B.

We refer to *snapping* as the process of refining a sequence-level alignment into an onset-level alignment, i.e., estimating $\mathcal{M}^*$ from $\mathcal{M}^{\text{seq}}$, by incorporating additional information from a note-onset posteriorgram.

3.2 Snapping as Bipartite Graph Matching

We formulate *snapping* as a *bipartite graph-matching* problem, applied independently to each pitch, as illustrated in Figure 2 (right). For each pitch $p \in \mathcal{P}$, let $\mathcal{N}_p \subseteq \mathcal{N}$ denote the subset of notes whose pitch is p. We define

$$\mathcal{M}_p^* = \mathcal{M}^*|_{\mathcal{N}_p} : \mathcal{N}_p \to \mathcal{T}_{\text{a}}. \quad (11)$$

We estimate $\mathcal{M}_p^*$ with an onset-level alignment

$$\mathcal{M}_p : \mathcal{N}_p \to \mathcal{T}_{\text{a}}, \quad (12)$$

and subsequently unify these pitch-wise mappings to obtain a single onset-level alignment over the full note set,

$$\mathcal{M} : \mathcal{N} \to \mathcal{T}_{\text{a}}, \quad (13)$$

that estimates $\mathcal{M}^*$.

We assume that for each $p \in \mathcal{P}$ both the audio timeline and the score timeline are much longer than the number of occurrences $\mathcal{N}_p$, i.e., $T_{\text{a}}, T_{\text{s}} \gg |\mathcal{N}_p|$.

Although $\mathcal{M}_p^*$ is unknown, we assume access to a coarse onset-level alignment $\mathcal{M}^{\text{seq}} : \mathcal{N} \to \mathcal{T}_{\text{a}}$ induced by a sequence-level alignment (Figure 2, top left), which is locally accurate within a bounded temporal deviation B, independent of p, as in Equation 10.

We further assume the existence of a posteriorgram

$$f_\theta^p : \mathcal{T}_{\text{a}} \to [0, 1], \quad (14)$$

where $f_\theta^p(t)$ denotes the likelihood that an onset of pitch p occurs in the audio at time t. In practice, f_θ^p can be obtained from a neural-network-based automatic transcriber pre-trained on a related yet distinct domain.

Our goal is to estimate $\mathcal{M}_p^*$ using the prior information provided by f_θ^p and $\mathcal{M}^{\text{seq}}$; that is, we aim to refine the coarse estimation $\mathcal{M}_p^{\text{seq}} = \mathcal{M}^{\text{seq}}|_{\mathcal{N}_p}$ within the admissible windows defined by the temporal error bound B, guided by the posteriorgram f_θ^p. For each note occurrence $n \in \mathcal{N}_p$, we define the set of admissible candidate matches as

$$\Omega(n) = \{\, t \in \mathcal{T}_{\text{a}} \;\mid\; |t - \mathcal{M}^{\text{seq}}(n)| \leq B \,\}. \quad (15)$$

We refer to $\Omega(n)$ as the *snapping window*, and to B as the *snapping window size*. Since windows $\Omega(n)$ may overlap for different n, greedy per-window candidate selection (e.g., from left to right) may yield a suboptimal global matching (Figure 2, middle). An optimal matching can instead be obtained using classical bipartite graph matching algorithms [19, 20] [1] (Figure 2, right). We define a weighted bipartite graph

$$G = (V_1, V_2, E), \quad (16)$$

with vertex sets $V_1 = \mathcal{N}_p$ and $V_2 = \mathcal{T}_{\text{a}}$, and edge set

$$E = \bigcup_{n \in \mathcal{N}_p} \{\, (n, t) \mid t \in \Omega(n) \,\}. \quad (17)$$

Each edge (n, t) carries a weight $w_{n,t} = f_\theta^p(t)$, which quantifies the likelihood of matching the onset of the note event n to audio frame t. For all $t \notin \Omega(n)$, we set $w_{n,t} = 0$.

[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csgraph.min_weight_full_bipartite_matching.html

We seek the matching

$$\widehat{\mathcal{M}_p} = \arg\max_{\mathcal{M}_p} \sum_{n \in \mathcal{N}_p} w_{n,\mathcal{M}_p(n)} \tag{18}$$

subject to the following constraints:

$$\mathcal{M}_p(n) \in \Omega(n), \quad \text{for all } n \in \mathcal{N}_p, \tag{19}$$

$$n \neq n' \implies \mathcal{M}_p(n) \neq \mathcal{M}_p(n'), \quad \text{for all } n, n' \in \mathcal{N}_p. \tag{20}$$

We solve for $\widehat{\mathcal{M}_p}$ using the algorithms of [19, 20].

As onset-level alignments obtained via snapping are used as labels for training a transcription model, the snapping window size B directly controls the strength of supervision. It can be adjusted based on the expected accuracy of the initial sequence-level alignment: smaller windows provide stronger supervision but require more precise sequence-level alignment, while larger windows tolerate greater alignment errors at the cost of weaker supervision.

4. INSTRUMENT-AGNOSTIC TRANSCRIPTION

Instrument-agnostic transcription estimates note activity without distinguishing between instruments. In this section we formalize this task, and outline the simplifying assumptions used in this work.

Starting from a note-event list (Equations 4, 5), we derive a note-onset piano roll $M^{\text{on}} \in \{0,1\}^{T_\text{s} \times P}$, where T_s is the number of timesteps and P the number of pitch bins:

$$M^{\text{on}}_{t,p} = \begin{cases} 1, & \exists\, t^{\text{off}} \in \mathcal{T}_\text{s},\, i \in \mathcal{I} : (p, t, t^{\text{off}}, i) \in \mathcal{N}, \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

Thus, $M^{\text{on}}_{t,p} = 1$ if and only if there exists a note of pitch p that starts at time t. In this work we focus on onset detection, though a note-activity piano roll considering note duration could be defined similarly.

Because instrument labels are discarded, notes from different instruments with the same pitch and onset time are merged into a single active entry in M^{on}. This is an inherent property of the instrument-agnostic formulation.

Our objective is to train a transcriber to directly predict from audio the piano roll corresponding to the underlying (latent) note-event list of the performance.

Training data are obtained by aligning score-derived piano rolls with real performances. In multi-instrument recordings, inconsistencies may arise when notes of the same pitch simultaneous in the score occur at different audio frames, or when notes from different score timesteps coincide in audio. Focusing on chamber music and piano, we assume such cases are rare and do not substantially affect training. They may be more prominent in orchestral settings; however, investigating this phenomenon is left for future work.

5. EXPERIMENTS

In this section, we present our experiments. We begin by introducing the datasets used for training and evaluation (Section 5.1). We then describe our approach for evaluating transcription and alignment accuracy (Section 5.2), followed by cross-dataset transcription evaluation (Section 5.3) and alignment accuracy evaluation (Section 5.4).

5.1 Datasets

Below, we describe the datasets used in this work. Importantly, MusicNet [13] is the *only* dataset used for training; all other datasets—including MAESTRO [11] which provides its own training split—are used solely for evaluation.

MusicNet [13] is a 34-hour dataset of 11 instruments, including multi-instrument chamber music ensembles of 1–8 players. Its main advantage is acoustic diversity and exclusive use of professional recordings. Its main limitation is low alignment accuracy of score annotations, making them insufficient for training transcription systems. We correct annotation timing errors using snapping, which we show to be highly effective.

MAESTRO [11] is a 140-hour dataset of solo piano performances recorded on a Disklavier. It provides highly accurate onset and offset annotations, but is limited in acoustic diversity. We use only the 20-hour test set for evaluation, disregarding the training set.

Saarland Music Data (SMD) [21] is a 5-hour dataset of solo piano performances recorded on a Disklavier, similar to MAESTRO but offering different acoustic conditions.

URMP [22] is a multi-instrument, multi-track chamber music dataset totaling 80 minutes. Each musical piece comprises up to five instrumental tracks. Its isolated monophonic recordings enable relatively accurate onset annotations (although not as precise as those of a Disklavier). Nevertheless, the dataset is small and acoustically uniform.

ChoraleBricks [23] is a multi-track wind music dataset consisting of four-part chorales, created using methods similar to those employed for URMP. The total duration of all pieces is less than 7 minutes. However, each of the four parts in every piece is recorded with multiple instruments, which gives the possibility of creating many combinations of instruments for each chorale. For our work, we use all possible four-part combinations, resulting in over 52 hours.

PHENICX [24] is an orchestral multi-track dataset created similarly to URMP and ChoraleBricks, but featuring full orchestral performances and a total duration of just over 10 minutes. Each piece comprises between 10 and 40 instrumental tracks. Like URMP, onset annotations are relatively accurate, but the dataset lacks acoustic diversity due to its uniform recording environment.

The Beethoven Symphony Excerpts Dataset (BSED) is an internal orchestral evaluation dataset with a total duration of 37 minutes, created by aligning musical scores with corresponding audio recordings. Its primary advantages are its acoustic diversity and the professional quality of the recorded performances. As separated tracks are not available, annotations were produced using audio–score alignment techniques [1, 2].

5.2 Evaluating Alignment Accuracy

There are two general approaches to evaluating alignment accuracy. The first is direct, comparing the estimated alignment to a ground-truth reference. The second is indirect, measuring how alignment affects transcription performance when aligned scores serve as training labels. The direct approach is explicit but less practical, as reference alignments at millisecond precision are rare. The indirect approach, though less explicit, is more scalable and provides meaningful evaluation via its effect on transcription metrics.

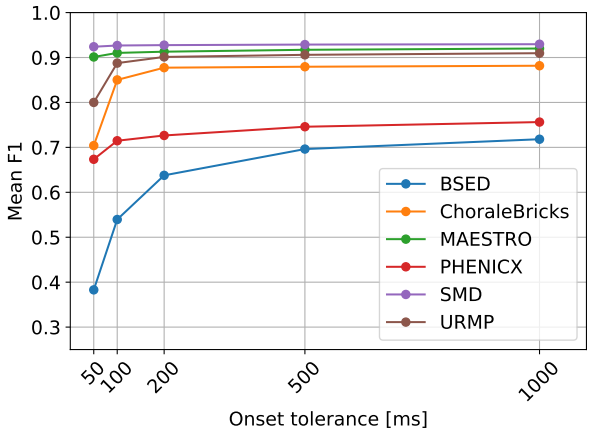

Figure 3. Note-level F1 score as a function of onset tolerance threshold, using the `DTW-BiP` (0.64s) model.

Accordingly, we adopt the indirect approach as our primary evaluation, measuring alignment quality through its effect on transcription performance. To complement this, we also perform a direct evaluation in a controlled setting. Our experiments correspond to the two approaches:

(i) **Cross-dataset transcription (Section 5.3):** We compare how different alignment policies used in training data annotation affect transcription performance.

(ii) **Direct alignment evaluation (Section 5.4):** Using the MAESTRO dataset which has high-accuracy reference alignments, we test alignment quality by attempting to recover the ground truth from intentionally perturbed labels.

5.2.1 Metrics

Our evaluation metrics consist of note-level precision, recall, and F1 score, where a predicted note is counted as correct if its onset is within a temporal threshold of the reference onset timing. In the following section (5.2.2) we elaborate on the exact threshold used.

5.2.2 Handling Evaluation Set Misalignment

It is important to recognize that some datasets may contain timing inaccuracies in their annotations, particularly those not recorded with specialized hardware such as a Disklavier. To prevent distorted evaluation, we adjust the onset tolerance according to the expected annotation quality.

For the piano datasets MAESTRO and SMD, the Disklavier provides temporal precision on the order of 3 ms. Consequently, for these datasets we adopt the standard 50 ms onset-tolerance threshold.

In the strings and winds datasets—URMP, ChoraleBricks, PHENICX, and BSED [2] —note-onset and offset annotations may be misaligned. In principle, these misalignments could be corrected using snapping; however, because snapping is precisely the method we aim to evaluate, doing so would bias the results. To accommodate small alignment errors, we instead apply a 100 ms onset tolerance threshold for URMP, ChoraleBricks, and PHENICX.

[2] In addition to sequence-level alignments, BSED also provides accurate note-level alignments obtained using snapping. Since this evaluation investigates snapping itself, we intentionally do not use them. Furthermore, for similar reasons, we use sequence-level alignments derived from signal-processing features rather than neural transcription features.

As shown in Figure 3, this results in a substantial increase in F1 for URMP, ChoraleBricks, PHENICX, whereas the impact on MAESTRO and SMD—whose annotations are highly accurate—is minimal. This pattern suggests that lower performance under tight thresholds often reflects annotation errors rather than transcription errors. We therefore consider 100 ms a reasonable approximation of typical human annotation error for these datasets.

BSED labels are obtained via DTW-based sequence alignment [2] on full polyphonic mixes, which can introduce larger timing errors. To account for sequence-level alignment inaccuracies, which may reach 0.5–1 seconds as shown in the following sections, we use a tolerance threshold of 500 ms. Figure 3 supports this choice: whereas in the multitrack datasets most of the F1 increase occurs when raising the tolerance from 50 ms to 100 ms, the F1 scores for BSED increase substantially up to a 500 ms tolerance.

5.3 Cross-dataset Transcription: MusicNet

In this section we focus exclusively on cross-dataset evaluation to better reflect real-world conditions: All compared models are trained on MusicNet, and tested on MAESTRO, SMD, URMP, ChoraleBricks, PHENICX, and BSED. All models are initialized from the same instrument-agnostic synthetic pre-training, released by [6].

We report results in order of increasing ensemble complexity. We begin with piano transcription (Section 5.3.2), proceed to instrument-agnostic transcription for small string and wind ensembles (Section 5.3.3), and finally address instrument-agnostic orchestral transcription (Section 5.3.4).

5.3.1 Compared Models

We compare models trained in an EM manner [4] under different alignment strategies varying in snapping algorithm, window size, and initial sequence alignment. We consider two snapping policies: Our proposed bipartite graph matching (`BiP`) and greedy snapping (`Gre`), commonly used in prior work. Snapping windows range from 0.1 s to 60 s.

Two types of initial sequence-level alignments are considered: Dynamic Time Warping based on chroma and onset features [1] (`DTW`) and linear stretching of the score to the audio timeline (`LS`). For `DTW`, snapping windows range from 0.1 s to 2 s. Since `LS` can introduce substantial alignment errors, it requires larger snapping windows to compensate; accordingly, we use windows of 2 s to 60 s.

For example, `DTW-BiP` (0.64s) denotes bipartite graph-based snapping with 0.64-second windows applied after DTW, whereas `LS-Gre` (60s) denotes greedy snapping with 60-second windows applied after linear stretching.

We also compare our method to a recent top-K peak-picking histogram-based approach [6] (`Hist`).

Finally, `Synth` refers to the instrument-agnostic, synthetically pre-trained model released by [6], from which all compared models are initialized.

5.3.2 Piano Transcription

Table 1 presents piano transcription results, training on MusicNet and evaluating on MAESTRO and SMD.

As shown, `DTW` alone produces poor labels, substantially reducing F1 compared to the `Synth` baseline (e.g., MAESTRO: 84.7% $\rightarrow$ 62.1%). Applying snapping after `DTW`

Transcriber	MAESTRO			SMD		
	P	R	F1	P	R	F1
Synth	88.4	81.6	84.7	93.0	85.6	88.9
DTW	96.6	48.6	62.1	96.1	56.0	69.0
Snapping: Previous Work [4, 6]						
DTW-Gre (2s)	94.0	84.8	89.0	97.5	87.4	92.1
DTW-Gre (0.64s)	93.8	85.2	89.2	97.5	87.2	91.9
DTW-Gre (0.1s)	95.8	84.5	89.6	98.1	85.9	91.4
LS-Gre (60s)	92.8	79.0	85.2	95.1	84.5	89.3
Hist	94.6	86.0	89.9	97.6	88.1	92.5
Snapping: Ours						
DTW-BiP (2s)	94.4	**86.6**	**90.2**	97.9	**88.2**	**92.7**
DTW-BiP (0.64s)	95.0	85.9	90.1	**98.2**	87.5	92.4
DTW-BiP (0.1s)	**96.3**	84.6	89.9	98.1	86.9	92.0
LS-BiP (60s)	93.1	84.2	88.3	96.9	87.6	91.9
LS-BiP (20s)	94.5	83.1	88.2	97.8	86.6	91.7
LS-BiP (10s)	94.4	80.4	86.6	97.7	85.7	91.1
LS-BiP (2s)	94.5	52.4	65.1	96.1	64.0	74.4

Table 1. Piano transcription results. Models are trained on MusicNet and evaluated on MAESTRO and SMD.

Transcriber	URMP			ChoraleBricks		
	P	R	F1	P	R	F1
Synth	77.6	78.0	77.5	87.5	72.7	79.1
DTW	97.1	62.9	75.5	97.3	53.9	68.6
Snapping: Previous Work [4, 6]						
DTW-Gre (2s)	91.2	84.5	87.5	93.3	77.0	83.9
DTW-Gre (0.64s)	93.0	85.5	89.0	94.2	80.1	86.2
DTW-Gre (0.1s)	**94.2**	83.7	88.4	**95.2**	77.3	84.9
LS-Gre (60s)	83.3	78.4	80.3	81.1	78.4	78.9
Hist	88.7	85.7	87.1	90.1	79.0	83.7
Snapping: Ours						
DTW-BiP (2s)	90.8	**86.8**	88.6	91.7	**82.4**	86.5
DTW-BiP (0.64s)	92.8	86.3	**89.3**	93.4	82.2	**87.0**
DTW-BiP (0.1s)	93.4	85.2	88.9	94.9	77.5	84.8
LS-BiP (60s)	87.3	81.4	84.0	89.1	73.0	79.2
LS-BiP (20s)	85.8	83.5	84.4	89.1	75.6	80.9
LS-BiP (10s)	90.1	81.1	85.2	92.6	70.1	78.7
LS-BiP (2s)	88.8	72.9	79.2	91.9	63.6	74.6

Table 2. Transcription results on small multi-instrument ensembles. Models are trained on MusicNet and evaluated on URMP and ChoraleBricks.

yields large gains (∼23–28%), increasing MAESTRO F1 to 89–90%. Relative to Synth, snapping provides a ∼3–5% improvement (84.7% → 90.2% with DTW-BiP, 0.64 s), demonstrating effective domain adaptation with weakly aligned data. These results show that snapping is essential for training with weakly aligned labels and that it makes such training beneficial.

After showing snapping to be essential, we compare different snapping methods. Our graph-based approach yields marginal yet consistent improvements of approximately 0.5–1.2% across models and datasets over greedy snapping used in prior work [4]. For example, on MAESTRO, using a 2 s window, F1 increases from 89.0 (DTW-Gre 2s) to 90.2 (DTW-BiP 2s). We further observe that the recent top-K peak picking histogram-based method (Hist) slightly outperforms greedy snapping. For example, on SMD, the F1 score increases from 91.9 (DTW-Gre 0.64s) to 92.5 (Hist). We hypothesize that this is because the top-K method does not suffer from issues caused by overlapping snapping windows. However, graph-based snapping (e.g., DTW-BiP 2s) still slightly outperforms the histogram top-K method. Although the difference is small for piano transcription, we will see that the gap becomes larger for strings and winds, reaching up to 5% (Sections 5.3.3, 5.3.4).

Next, we analyze the effect of snapping window size. A 0.1 s window already substantially outperforms the Synth baseline and DTW, and larger windows yield further gains, mainly through higher recall. On SMD, increasing the window from 0.1 s to 2 s raises recall and F1 from $86.9/92.0$ to $88.2/92.7$ (DTW-BiP 2s).

Finally, we examine extremely large windows. With linear stretching, small windows fail (F1 on MAESTRO degrades from 84.7 to 65.1 with LS-BiP 2 s). Compensating with 60-second windows, greedy snapping (LS-Gre 60 s) yields minimal gains if any at all compared to Synth (F1 84.7 → 85.2, recall 81.6 → 79.0), whereas graph-based (LS-BiP 60 s) substantially improves both recall and F1 (F1 84.7 → 88.3, recall 81.6 → 84.2). This shows that as the window size grows, greedy snapping degrades due to increased overlaps, while graph-based snapping remains robust.

5.3.3 Strings & Winds Transcription—Small Ensemble

Table 2 presents results for *instrument-agnostic* (multi-instrument) transcription on small musical ensembles. The compared models were trained on MusicNet and evaluated on URMP (chamber music with strings and winds) and ChoraleBricks (small wind ensembles).

Similar trends are observed to those seen in piano transcription: DTW labels degrade performance relative to the Synth baseline. For example, F1 on ChoraleBricks drops from 79.1 to 68.6. Applying snapping after DTW substantially improves F1. For example, DTW-BiP with a 0.64 s window increases F1 relative to Synth from 77.5 to 89.3 on URMP and from 79.1 to 87.0 on ChoraleBricks.

Graph-based snapping outperforms both greedy snapping and the histogram-based method. With 2 s windows, graph-based snapping (DTW-BiP 2s) improves F1 over greedy snapping (DTW-Gre 2s) from 83.9 to 86.5 on ChoraleBricks and from 87.5 to 88.6 on URMP. Compared to Hist, F1 increases from 83.7 to 86.5 on ChoraleBricks and from 87.1 to 88.6 on URMP. Larger windows amplify the benefit of graph-based snapping; on URMP, LS-BiP 60s improves F1 relative to LS-Gre 60s from 80.3 to 84.4.

Overall, results on URMP slightly exceed those on ChoraleBricks, possibly due to the higher proportion of string instruments in MusicNet and the higher annotation precision of URMP; increasing the onset tolerance from 100 ms to 200 ms raises F1 by ∼3% on ChoraleBricks, but only by ∼1% on URMP (Figure 3).

Compared to piano results (Table 1), performance on URMP and ChoraleBricks falls within a similar range, indicating that onset detection can be effective for string and wind instruments despite their less pronounced onsets.

Transcriber	PHENICX			BSED		
	P	R	F1	P	R	F1
Synth	83.8	54.8	66.0	83.2	39.5	51.3
DTW	93.7	41.9	57.2	89.3	15.4	24.3
Snapping: Previous Work [4, 6]						
DTW-Gre (2s)	85.8	60.0	70.4	81.6	59.9	67.2
DTW-Gre (0.64s)	85.5	60.3	70.4	78.9	62.4	67.6
DTW-Gre (0.1s)	87.1	60.6	71.2	83.9	53.4	62.3
LS-Gre (60s)	78.7	62.2	68.8	72.3	59.1	62.4
Hist	86.3	60.7	71.2	74.8	58.4	64.1
Snapping: Ours						
DTW-BiP (2s)	86.3	61.5	71.5	79.7	62.2	67.8
DTW-BiP (0.64s)	84.4	**62.3**	71.5	81.4	**63.8**	**69.6**
DTW-BiP (0.1s)	**87.6**	60.9	**71.7**	**84.4**	53.9	63.4
LS-BiP (60s)	85.1	61.4	71.0	72.6	62.2	64.7
LS-BiP (20s)	84.9	60.7	70.5	73.3	62.0	64.9
LS-BiP (10s)	85.4	60.3	70.3	73.3	61.3	64.5
LS-BiP (2s)	85.6	36.4	50.8	74.1	47.4	55.0

Table 3. Orchestral transcription results. Models are trained on MusicNet and evaluated on BSED and PHENICX.

5.3.4 Orchestral Transcription

Table 3 shows orchestral transcription results, traning on MusicNet and evaluating on PHENICX and BSED. As with piano and small ensembles, DTW alone is insufficient, while snapping (DTW-BiP, DTW-Gre) improves alignment, with graph-based snapping consistently outperforming greedy snapping. For example, DTW-BiP 0.64s raises F1 on PHENICX compared to DTW-Gre 0.64s from 70.4 to 71.5 and on BSED from 67.6 to 69.6. Improvement on BSED compared to Hist is substantial: From 64.1 to 69.6.

Overall metrics are lower than for piano or small-ensemble settings. For example, DTW-BiP (0.64 s) achieves an F1 score of 89.3 on URMP but only 71.7 on PHENICX. We attribute this gap to increased orchestral complexity: Performances in PHENICX involve 10–40 instruments, while URMP ensembles include no more than five.

5.4 Alignment Evaluation—MAESTRO

We evaluate alignment accuracy on the MAESTRO dataset, for which precise reference annotations are available. We randomly and uniformly perturb the MAESTRO onset labels within a window $[-w, w]$, where $w \in \{1, 5, 15, 60\}$ frames, corresponding to $0.032\,\text{s}$, $0.16\,\text{s}$, $0.48\,\text{s}$, and $1.92\,\text{s}$. The perturbed onsets are then recovered using snapping with the same window size. Alignment quality is measured by the F1 score with respect to the ground truth.

We compare three posteriorgrams obtained from the following: (i) the synthetically pre-trained model (Synth); (ii) the DTW-BiP 0.64 s model fine-tuned from Synth on MusicNet with weakly-aligned labels using snapping (MN); and (iii) the ground-truth piano roll (GT) as a sanity check.

Results are shown in Table 4. The GT posteriorgram achieves an F1 score of 100% for all w, for both greedy and graph-based snapping, confirming that snapping operates correctly with ideal posteriorgrams.

Method	w [s]	Pert	Synth	MN	GT
Gre	0.032	100	95.9	96.4	100
	0.16	30.1	92.5	93.0	100
	0.48	16.7	88.6	89.9	100
	1.92	9.6	84.7	87.1	100
BiP	0.032	100	95.9	96.4	100
	0.16	30.1	93.6	94.3	100
	0.48	16.7	91.0	92.8	100
	1.92	9.6	87.8	91.1	100

Table 4. F1 score evaluation of snapping applied to the perturbed MAESTRO test set (Pert) across varying perturbation windows w, using different onset posteriorgrams (Synth, MN, GT).

With Synth, F1 decreases as w increases, but graph-based snapping (BiP) still remains robust, achieving F1 of 91.0 at $w = 0.48\,\text{s}$ and 87.8 at $w = 1.92\,\text{s}$, compared to F1 scores of 16.7 and 9.6 for the perturbed labels (Pert). The differences between graph-based and greedy snapping grow with w: 93.6–92.5 = 1.1 for $w = 0.16\,\text{s}$, 91.0–88.6 = 2.4 for $w = 0.48\,\text{s}$, and 87.8–84.7 = 3.1 for $w = 1.92\,\text{s}$. This can be expected: larger windows increase overlap, making greedy snapping less optimal.

The MusicNet-trained model (MN) shows similar trends. However, it has consistently improved performance across snapping methods and window sizes compared to Synth; for example, at $w = 0.48\,\text{s}$ with graph-based snapping (BiP), F1 increases from 91.0 (Synth) to 92.8 (MN). This demonstrates that training with snapping on weakly-aligned data not only improves transcription metrics (as seen in Section 5.3), but also enhances alignment accuracy when using transcription features for snapping.

Finally, comparing with piano transcription results in Table 1, we find that the difference between greedy and graph-based snapping is larger for alignment than for transcription (up to 4% versus about 1%). This may indicate that the training process is robust to some degree of label noise, provided that a large portion of onsets is accurately aligned.

6. CONCLUSION

In this work, we investigated *snapping*—the refinement of sequence-level alignments into precise onset-level alignments using neural onset posteriorgrams. Through cross-dataset evaluations and controlled experiments, we showed that snapping is highly effective for onset-level alignment, enabling training transcribers with weakly-aligned labels with further gains when following a graph-based approach. Although we focused on instrument-agnostic AMT, the methods used may be extended to instrument-sensitive transcription, and possibly other timing-critical MIR tasks such as drum transcription or multi-pitch estimation.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant No. 500643750 (MU 2686/15-1). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS.

7. REFERENCES

[1] S. Ewert, M. Müller, and P. Grosche, "High Resolution Audio Synchronization Using Chroma Onset Features," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009, pp. 1869–1872.

[2] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, "Sync Toolbox: A Python Package for Efficient, Robust, and Accurate Music Synchronization," *Journal of Open Source Software (JOSS)*, vol. 6, no. 64, pp. 3434:1–4, 2021.

[3] J. Zeitler, B. Maman, and M. Müller, "Robust and Accurate Audio Synchronization Using Raw Features from Transcription Models," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

[4] B. Maman and A. H. Bermano, "Unaligned Supervision for Automatic Music Transcription in The Wild," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2022, pp. 14 918–14 934.

[5] X. Riley, D. Edwards, and S. Dixon, "High Resolution Guitar Transcription Via Domain Adaptation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2024, pp. 1051–1055.

[6] J. Yaffe, B. Maman, M. Müller, and A. Bermano, "Count The Notes: Histogram-Based Supervision for Automatic Music Transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2025.

[7] N. Hu and R. B. Dannenberg, "Bootstrap learning for accurate onset detection," *Machine Learning*, vol. 65, no. 2-3, pp. 457–471, 2006.

[8] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic Music Transcription: An Overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[9] P. Smaragdis and J. C. Brown, "Non-Negative Matrix Factorization for Polyphonic Music Transcription," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2003, pp. 177–180.

[10] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. H. Engel, S. Oore, and D. Eck, "Onsets and Frames: Dual-Objective Piano Transcription," in *Proceedings of the International Society for Music Information Retrieval Conference, (ISMIR)*, 2018, pp. 50–57.

[11] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. H. Engel, and D. Eck, "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[12] 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 of Audio, Speech, and Language Processing*, vol. 29, pp. 3707–3717, 2021.

[13] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Learning Features of Music from Scratch," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[14] X. Riley, Z. Guo, and S. Edwards, Drew abd Dixon, "GAPS: A Large and Diverse Classical Guitar Dataset and Benchmark Transcription Model," *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

[15] N. C. Tamer, Y. Özer, M. Müller, and X. Serra, "High-Resolution Violin Transcription using Weak Labels," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023, pp. 223–230.

[16] Y. Wu, B. Chen, and L. Su, "Multi-Instrument Automatic Music Transcription with Self-Attention-Based Instance Segmentation," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 28, pp. 2796–2809, 2020.

[17] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. H. Engel, "MT3: Multi-Task Multitrack Music Transcription," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[18] Y. Wu, W. Wei, D. Li, M. Li, Y. Yu, Y. Gao, and W. Li, "Harmonic Frequency-Separable Transformer for Instrument-Agnostic Music Transcription," in *2024 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2024, pp. 1–6.

[19] R. M. Karp, "An algorithm to solve the m$\times$ n assignment problem in expected time O (mn log n)," *Networks*, vol. 10, no. 2, pp. 143–152, 1980.

[20] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, 1987.

[21] M. Müller, V. Konz, W. Bogler, and V. Arifi-Müller, "Saarland Music Data (SMD)," in *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[22] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a Multitrack Classical Music Performance Dataset for Multimodal Music Analysis: Challenges, Insights, and Applications," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 522–535, 2019.

[23] S. Balke, A. Berndt, and M. Müller, "ChoraleBricks: A Modular Multitrack Dataset for Wind Music Research," *Transaction of the International Society for Music Information Retrieval (TISMIR)*, vol. 8, no. 1, pp. 39–54, 2025.

[24] C. C. S. Liem, E. Gómez, and M. Schedl, "PHENICX: Innovating the Classical Music Experience," in *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2015, pp. 1–4.