LOSE THE FRAMES: EVENT-BASED METRICS FOR EFFICIENT MUSIC STRUCTURE ANALYSIS EVALUATIONS

Qingyang (Tom) Xi
Music and Audio Research Lab
New York University
tom.xi@nyu.edu

Brian McFee
Music and Audio Research Lab
New York University
brian.mcfee@nyu.edu

ABSTRACT

Many evaluation metrics in Music Information Retrieval (MIR) rely on uniform time sampling of phenomena that unfold over time. While uniform sampling is suitable for continuously varying concepts such as pitch or dynamic envelope, it is suboptimal for inherently discrete or piecewise constant events, such as labeled segments. Current Music Structure Analysis (MSA) metrics for label evaluation are all implemented with time sampling, which can be inexact and inefficient. In this work, we propose event-based implementations of the three most widely used MSA metrics. Our approach yields evaluations that are more accurate, more computationally efficient, and more reproducible, streamlining MSA research workflows.

1. INTRODUCTION

Efficient and accurate evaluation metrics are vital for progress in MIR. Currently, many MIR metrics rely on uniform time sampling. While suitable for continuously varying phenomena like pitch, this approach introduces inaccuracies and computational inefficiencies for discrete or piecewise-constant annotations, such as chord estimation, sound event detection, and other labeled intervals. Moreover, sampling introduces an arbitrary hyperparameter, the frame size, that forces a trade-off between numerical accuracy and computational efficiency.

These issues are especially pronounced for Music Structure Analysis (MSA) metrics, where labeled segmentations are routinely evaluated using the Pairwise Frame Clustering (PFC) score [1], the Normalized Cross Entropy (NCE) score [2], the V-measure [3], and the L-measure [4]. Although the current implementations provided by the standard MIR evaluation toolkit `mir_eval` [5] are widely adopted and optimized, they still operate with a paradigm based on uniform time sampling. Under this sampling-based paradigm, metric computation can become prohibitively expensive, often quadratic or worse in the number of frames, and numerically sensitive to the choice of frame size. This inefficiency hinders robust hyperparameter tuning, slows iterative model development, and makes large-scale studies impractical.

We introduce a new event-based paradigm for implementing three common MSA metrics: the PFC, the V-measure, and the L-measure. [1] Our event-based paradigm offers a general strategy for evaluating time series annotations by moving away from frame-based sampling. This approach decouples numerical precision from computational cost, providing exact evaluations at speeds that are orders of magnitude faster than frame-based methods. We use MSA metrics to demonstrate how this paradigm inherently improves numerical stability and reproducibility. This unlocks the potential for more scalable and robust research not only for MSA but also for other common MIR tasks such as chord estimation or sound event detection.

2. RELATED WORKS

The motivation behind our event-based metrics originates from a long-standing reliance on sampling-based evaluation within the MIR community. This practice, established in the Music Information Retrieval Evaluation eXchange (MIREX) campaigns, is now codified in widely adopted toolkits such as `mir_eval` [5]. This reliance on a user-defined frame rate requires prior knowledge of a task's characteristic timescale, forcing a domain-specific heuristic into what should be an objective measurement.

Besides being a practical nuisance, the act of picking an evaluation frame rate introduces vulnerability into the evaluation itself, as reported performances are influenced by arbitrary parameter choices. This scenario introduces a methodological vulnerability, as reported performances can become susceptible to the "Clever Hans" effect, where a system's apparent success stems from exploiting arbitrary choices or artifacts in the test design, rather than from true understanding [6]. Our work focuses on removing this layer of methodological variance.

Music segmentation is not the only task that can benefit from an event-based formulation; related areas, such as sound event detection (SED), have also faced challenges associated with frame-based evaluations. The SED community's standard evaluation toolkit, `sed_eval` [7], employs a relatively coarse default frame rate of 1 second, due to the longer durations of audio recordings involved with

[1] https://github.com/tomxi/frameless-eval

the task. This has led to recent work by Bilen et al. [8] and Lostanlen and McFee [9] that explicitly proposes more efficient event-based metrics for SED.

Despite this clear trend towards event-based evaluation in related fields, the core metrics for Music Structure Analysis within the field's standard toolkit have remained exclusively dependent on the sampling paradigm. This paper addresses this gap by presenting event-based implementations for the canonical MSA metrics, removing a long-standing inefficiency in MSA evaluation, and enabling more robust, reproducible research.

3. SEGMENTATION METRICS

Different metrics have been proposed to evaluate flat and hierarchical music segmentations. Flat segmentations are often evaluated using metrics that are based on the concept of clustering. [2] The pairwise frame clustering score (PFC) [1] and the V-measure [3] both fall into this category. For evaluating hierarchical segmentations, the L-measure [4] has been used in many recent works on MSA [10–13].

All three metrics are implemented in `mir_eval` [5] by sampling, with a default sampling rate of 10 Hz. Although subtle, the frame size for these metrics also affects the evaluation results in unexpected ways. We now review these three metrics and provide our event-based formulations.

3.1 Pairwise Frame Clustering

For a piece of music with time span $T = [t_0, t_1]$, a segmentation has a label mapping $S(t)$ that maps time points in T to a set of k unique labels $\gamma = \{y_1, y_2, \dots y_k\}$:

$$S : T \to \gamma \tag{1}$$

When comparing an estimated segmentation $\hat{S}$ with a reference S, the two do not need to share the same set of labels. Instead, their comparison relies on internal labeling consistency, which identifies points labeled identically within each segmentation.

This consistency is captured by the *label agreement map*, defined as:

$$M_S(u, v) := [S(u) = S(v)]_{\mathbb{1}}\,, \tag{2}$$

where $[\cdot]_{\mathbb{1}}$ is the indicator function that returns 1 if the condition is true and 0 otherwise. It should be noted that although $M_S(u, v)$ is piecewise constant, it is a continuous-time function mapping $M_S : T^2 \to \{0, 1\}$.

We use M_S to define the set of time pairs that meet, which forms a set of significant time pairs that can be considered as information to be recalled:

$$\mathcal{A}(S) := \{(u, v) \mid M_S(u, v) = 1\} \tag{3}$$

Figure 1 shows a simple example of a set of reference and estimated segmentations $S, \hat{S}$ and their corresponding set of meeting positions $\mathcal{A}(S)$, $\mathcal{A}(\hat{S})$ in its top two rows.

[2] The boundary hit rate metric already uses an event-based formulation, and therefore we focus our attention on the two metrics that focus on labeling.

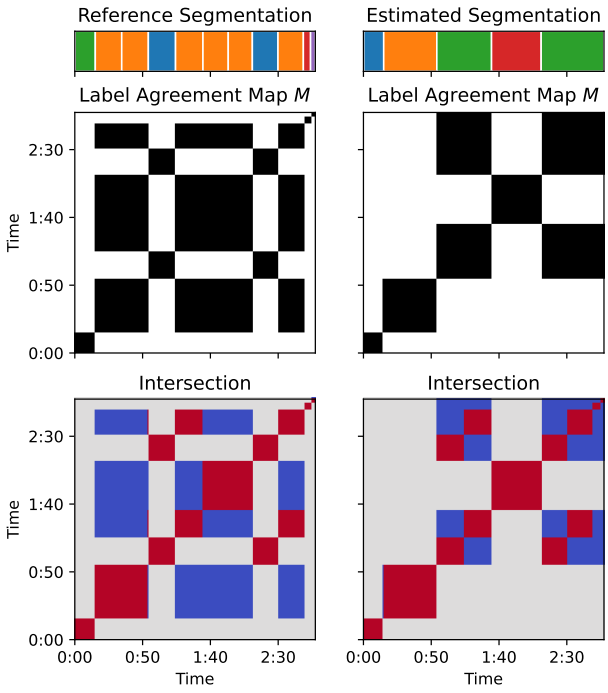

Figure 1. Visualizing PFC as ratio of area. Top: reference S and estimated $\hat{S}$. Middle: meeting positions for each segmentation: $\mathcal{A}(S)$, $\mathcal{A}(\hat{S})$. Bottom: intersection $\mathcal{A}(S) \cap \mathcal{A}(\hat{S})$ highlighted in red.

Introduced by Levy and Sandler [1], the pairwise frame clustering (PFC) metric evaluates segmentation agreement by considering these meeting pairs. The PFC metric quantifies the proportion of meeting pairs common to both segments relative to those unique to each. The time pairs (u, v) that meet in both segmentations are colored red in the bottom of Figure 1. PFC recall and precision are then defined as the ratio of these areas.

$$\text{PFC}_R = \frac{\left|\mathcal{A}(S) \cap \mathcal{A}(\hat{S})\right|}{|\mathcal{A}(S)|}, \quad \text{PFC}_P = \frac{\left|\mathcal{A}(S) \cap \mathcal{A}(\hat{S})\right|}{|\mathcal{A}(\hat{S})|} \tag{4}$$

Here, $|\mathcal{A}(S)|$ represents the size of the set of meeting positions under S. The size of this region can be computed by integrating over the time pair space T^2.

$$\left|\mathcal{A}(S) \cap \mathcal{A}(\hat{S})\right| = \int_{T^2} M_S(u, v) \cdot M_{\hat{S}}(u, v)\ \mathrm{d}(u, v)$$
$$|\mathcal{A}(S)| = \int_{T^2} M_S(u, v)\ \mathrm{d}(u, v)$$

Notice that since $S(t)$ consists of discrete events and is therefore piecewise constant, the integrals can be computed as the sum of areas of rectangles, which have simple closed-form solutions.

To achieve this efficiently, we define a common set of intervals from the union of both segmentations' boundaries. Within each resulting interval, the segment labels remain jointly constant, allowing us to query each segmentation's label exactly once per interval, irrespective of in-

terval duration. This strategy leverages the piecewise constant property of segmentations, ensuring exact computations with minimal sampling.

In terms of computational complexity, the PFC metric requires sampling the label agreement map M over pairs, which is quadratic in the number of frames (n^2) for frame-based approaches. For the continuous-time approach, the complexity is $(s + \hat{s})^2$, where s and $\hat{s}$ are the number of segments for reference and estimation, respectively.

3.2 V-measure

While conceptually simple, PFC overlooks the issues of over- and under-segmentation, which motivated the adoption of Normalized Cross Entropy (NCE) [2], and the closely related V-measure [3]. For comparing flat segmentations' labels, the V-measure is a modern metric that improves upon the Normalized Conditional Entropy (NCE) with proper normalization.

The entropy of a segmentation $S : T \to \gamma$ can be examined by randomly sampling along its duration. We denote the sampled label as a random variable Y:

$$\mathbb{P}[Y = y] = \mathbb{P}_{t\sim T}[S(t) = y],$$
$$\mathbb{H}(S) = \mathbb{E}[-\log \mathbb{P}[Y]]$$

In particular, when segmentation is constant (that is, $S(t) = y$ for all t), $\mathbb{H}(S) = 0$.

The conditional entropy $\mathbb{H}(\hat{S}|S)$ measures the average entropy of the estimated labels $\hat{S}$ for each given segment label in the reference annotation S:

$$\mathbb{H}(\hat{S}|S) = \mathop{\mathbb{E}}_{y\sim\gamma}\left[\mathbb{H}\left(\hat{S}|S = y\right)\right] \tag{5}$$

The conditional entropy estimates the amount of uncertainty left in predicting the reference labels given the estimate segmentation. When the uncertainty for the reference label is low given the estimate, the estimate recalls the labeling information presented in the reference.

The V-measure score reflects how much information is shared between the two segmentations, relative to the amount of information contained within each one. [3] :

$$\mathrm{V}_R = 1 - \frac{\mathbb{H}(S|\hat{S})}{\mathbb{H}(S)}, \quad \mathrm{V}_P = 1 - \frac{\mathbb{H}(\hat{S}|S)}{\mathbb{H}(\hat{S})} \tag{6}$$

The probability and entropy are estimated by sampling in the frame-based paradigm but can be calculated exactly since $S(t)$ consists of discrete labeled sections and is piecewise constant.

$$\mathbb{P}[Y = y] = \frac{1}{|T|}\int_T [S(t) = y]_{\mathbb{1}}, \mathrm{d}t$$

To calculate the V-measure, a contingency table is used to represent the co-occurrence of segment labels between the reference and estimated segmentations. This table allows for the straightforward calculation of joint and marginal probabilities of label assignments.

[3] NCE differs from V-measure only by normalizing relative to the uniform distribution instead of the marginals

Populating the $k \times \hat{k}$ contingency table has a time complexity of $O(n)$, as it requires a single pass through all n frames. Once this $k\times\hat{k}$ table is created, the final V-measure score is calculated by computing the marginal entropies from the contingency table, a $O(k \times \hat{k})$ step, leading to an overall complexity of $O(k \times \hat{k} + n)$. Similarly, the event-based method results in a complexity of $O(k \times \hat{k} + s + \hat{s})$.

3.3 L-measure

Unlike flat segmentation metrics, which evaluate whether pairs of time points are assigned the same label, hierarchical metrics assess relationships across multiple levels of granularity. In this setting, the goal is not just to determine whether a pair of time points have matching label, but to evaluate how well the hierarchical structure of an estimated segmentation aligns with that of a reference.

The L-measure [4] addresses the task by explicitly considering differences in hierarchical depth as a ranking problem. Rather than evaluating pairs of points, it uses a triplet-based comparison that asks, for a given anchor time t, if a second time point u is more closely related to t than a third point v. If both the reference and estimated hierarchies agree on this ranking, then the structural information contained at time t is recalled.

In its current implementation within `mir_eval`, L-measure computes precision and recall densities separately at each anchor time point, before aggregating these values throughout the time domain. This is done to ensure that each time point t contributes equally to the overall metric. This calculation per time point is illustrated in Figure 2, which can be interpreted as a local recall density before aggregating it over the time domain.

We start by extending the notion of segments and its label agreement mapping $M(u, v)$ to hierarchies and its counterparts. A hierarchy of depth d is a sequence of progressively finer flat segmentations and has a label mapping $H : T \to (\gamma_1, \gamma_2, \ldots, \gamma_d)$:

$$H(t) = (S_1(t), S_2(t), \ldots, S_d(t)) .$$

The label agreement mapping for a hierarchy is defined for each time pair as the deepest level at which they receive the same label. For any pair of time points $u, v \in T$, the depth of their shared label is defined as

$$M_H(u, v) := \max\left\{d \,|\, S_d(u) = S_d(v)\right\}. \tag{7}$$

We plot two hierarchies and their label agreement maps in the first two rows of Figure 2.

Using this depth mapping, we can define a triplet (t, u, v) to be significant under hierarchy H if u is more closely related to t than v is:

$$\mathcal{A}(H; t) := \{(t, u, v) \in T^3 | M_H(t, u) > M_H(t, v)\}. \tag{8}$$

The third row in figure 2 shows $M_H(t, \cdot)$ for all times in T, and the fourth row shows significant triplets associated with query time t: where maroon marks $M_H(t, u) > M_H(t, v)$, i.e. $\mathcal{A}(H; t)$.

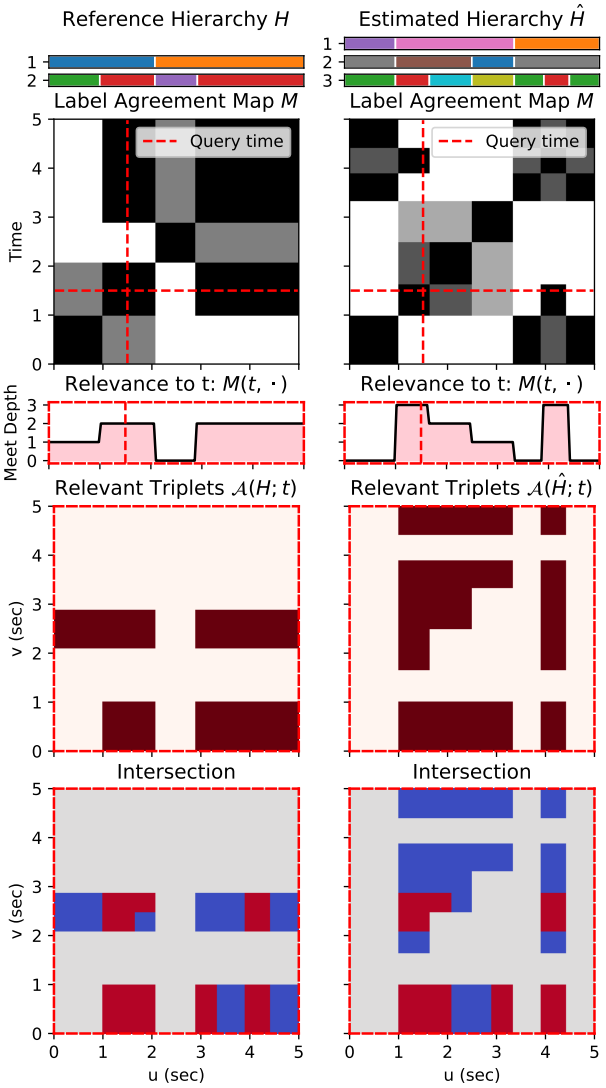

Figure 2. Visualizing L-measure's density at time t as ratio of area. Row 1: two hierarchies H and $\hat{H}$. Row 2: label agreement maps M_H and $M_{\hat{H}}$. Row 3: local relevance $M_H(t, \cdot)$. Row 4: significant triplets $\mathcal{A}(H; t)$, $\mathcal{A}(\hat{H}; t)$. Row 5: matching significant triplets $\mathcal{A}(H; t) \cap \mathcal{A}(\hat{H}; t)$ highlighted in red.

The precision and recall scores are then defined for each instant of time by counting the proportions of triplets shared between the two sets of triplets $\mathcal{A}(H; t) \cap \mathcal{A}(\hat{H}; t)$, shown in red on the bottom row of Figure 2.

For an estimated hierarchy $\hat{H}$ and a reference H, the density of L-measure at time t is defined as follows:

$$\rho_{\text{recall}}(\hat{H}|H; t) = \frac{\left|\mathcal{A}(\hat{H}; t) \cap \mathcal{A}(H; t)\right|}{\left|\mathcal{A}(H; t)\right|},$$
$$\rho_{\text{precision}}(\hat{H}|H; t) = \frac{\left|\mathcal{A}(\hat{H}; t) \cap \mathcal{A}(H; t)\right|}{\left|\mathcal{A}(\hat{H}; t)\right|}. \quad (9)$$

Notice how ρ is not defined when its denominator is $|\mathcal{A}(H; t)| = 0$. In the continuous-time formulation, this would imply that $M_H(t, u) = M_H(t, v)$ for all (u, v); or, in other words, a flat segmentation with constant labeling.

With ρ defined, the overall L-measure recall is defined as the average recall density over T:

$$\text{L}_{\text{recall}}(\hat{H}|H) = \frac{1}{|T|} \int_T \rho_H(\hat{H}; t)\ \text{d}t,$$
$$\text{L}_{\text{precision}}(\hat{H}|H) = \frac{1}{|T|} \int_T \rho_{\hat{H}}(H; t)\ \text{d}t. \quad (10)$$

In practical terms, the task of identifying the set of temporal triplets presents a computational complexity of $O(n^3)$ for n frames when employing a naive methodology. This becomes particularly challenging for extended sequences. To improve efficiency in this process, `mir_eval` implements an inversion counting algorithm to assess ranking discrepancies between two lists, thereby reducing the computational complexity to $O(n^2 \log n)$ time specifically in the context of the sampling case. Taking advantage of the same inversion counting algorithm, the complexity of the continuous approach is $O((s + \hat{s})^2 \log(s + \hat{s}))$.

Similarly to PFC, the L-measure also requires sampling the hierarchical label agreement map M_H, which is quadratic in the number of segments $s + \hat{s}$ or the number of frames n for each of the $d + \hat{d}$ levels.

3.4 Computational Complexity

Table 1 describes the computational time complexity of the three MSA metrics analyzed in this paper. We list the current frame-based approaches and the new event-based approaches for each metric side by side. We will see that when $s < n$ in section 5, our event-based implementations are significantly faster than the sampled versions, while maintaining accuracy.

Metric	Frame-Based	Event-Based
Pairwise	$O(n^2)$	$O(s^2)$
V-measure	$O(k^2 + n)$	$O(k^2 + s)$
L-measure	$O(n^2(d + \log n))$	$O(s^2(d + \log s))$

Table 1. Computational complexity of frame-based and event-based versions of MSA metrics. k is the number of unique labels, s is number of segments, n is number of frames, d is the depth of the hierarchy.

4. EXPERIMENTS

To empirically evaluate our event-based implementation of MSA metrics, we performed experiments using hierarchical structural annotations from the SALAMI dataset [14], a widely used corpus containing hierarchical music structure annotations. We used 884 tracks from SALAMI, each featuring two separate human-generated two-level hierarchies, allowing us to benchmark the proposed metrics using real annotations. We used the lower level of the hierarchy to compare flat segmentation metrics. Segmentations produced by Salamon et al.'s hierarchical MSA method [15] on the SALAMI dataset were also considered,

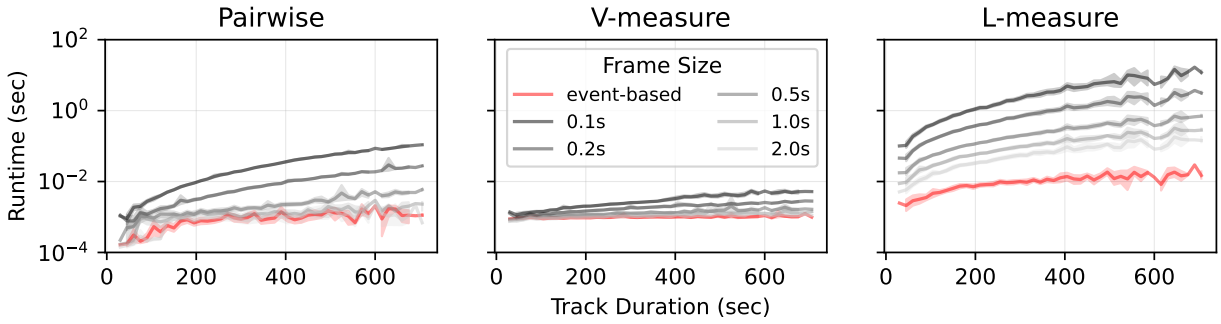

Figure 3. Runtime for MSA metrics against annotation duration, with colors representing different metric resolution. Shaded regions indicating 95% confidence interval.

providing deeper hierarchies that match realistic computational scenarios typical in current MIR research. We will refer to their approach as the Segment Fusion method.

4.1 Benchmarking Setup

We benchmarked our event-based implementations against the frame-based versions available in `mir_eval` [5]. All experiments were conducted on a 2021 MacBook Pro equipped with an M1 Max chip, 32 GB RAM, and Python 3.9 using `mir_eval` version 0.8.2. Larger frame sizes increase computation efficiency by reducing the number of frames to process, facilitating rapid prototyping iterations. However, these larger frames do not achieve the same level of accuracy as those with finer frame sizes. We conducted an assessment of both computational efficiency (in terms of run-time) and accuracy (in terms of metric score consistency). Our study evaluated the accuracy and runtime performance of `mir_eval`'s implementation across five distinct frame sizes $[0.1, 0.2, 0.5, 1.0, 2.0]$, measured in seconds.

4.2 Scenario: Hyperparameter Tuning

To contextualize the computational overhead in a real-world research scenario, we model our experiment on the hyperparameter tuning process of the Segment Fusion method [15]. Their optimization process involved evaluating 100 parameter combinations on a development set of 471 tracks. We report the cumulative run-time required to evaluate a single hyperparameter combination across this dataset using both our event-based and the standard frame-based methods.

4.3 L-measure and Depth

We also assessed the L-measure's runtime dependency on hierarchical depth, a concern for evaluating very deep hierarchies produced by segmentation algorithms. For this experiment, we took the 12-layer hierarchies from the Segment Fusion method and systematically reduced their depth, one layer at a time. At each depth, we reported the average runtime to compute the L-measure against a two-layer reference hierarchy.

Frame Size	Pairwise	V-measure	L-measure
0.1s	-0.0009	0.0003	-0.0004
0.2s	-0.0019	0.0010	-0.0009
0.5s	-0.0047	0.0030	-0.0021
1.0s	-0.0093	0.0059	-0.0035
2.0s	-0.0187	0.0111	-0.0076

Table 2. Average metric deviation between frame-based and event-based approach

5. RESULTS

5.1 Computational Efficiency

Figure 3 plots the run time of different framing schemes versus the duration of the annotation in seconds. This shows that our event-based implementation consistently outperforms the frame-based method in terms of runtime. Specifically, the run-time of our event-based implementation for L-measure remained close to 10 milliseconds per computation, regardless of the duration of the track, whereas frame-based computations' run-time can sometimes exceeded 10 seconds and grows super-linearly with duration. This substantial improvement in computational efficiency highlights the potential for integrating these metrics into larger-scale analyses or iterative workflows, such as hyperparameter tuning. Although not as pronounced, the V-measure and PFC metrics also show a reasonable speed up, especially with increasing track duration. This is not surprising, as our implementation's computational complexity depends on the number of boundaries, as opposed to the number of frames.

5.2 Frame Size Sensitivity

Although coarser frame sizes make the original sampling-based implementation faster to compute, they compromise accuracy. Our experiments reveal considerable sensitivity of the original frame-based metrics to the chosen frame size.

Table 2 illustrates the average deviation of the frame-based implementation for the three metrics when evaluated with different frame sizes. Furthermore, Figure 4 presents a detailed analysis of the numerical discrepan-

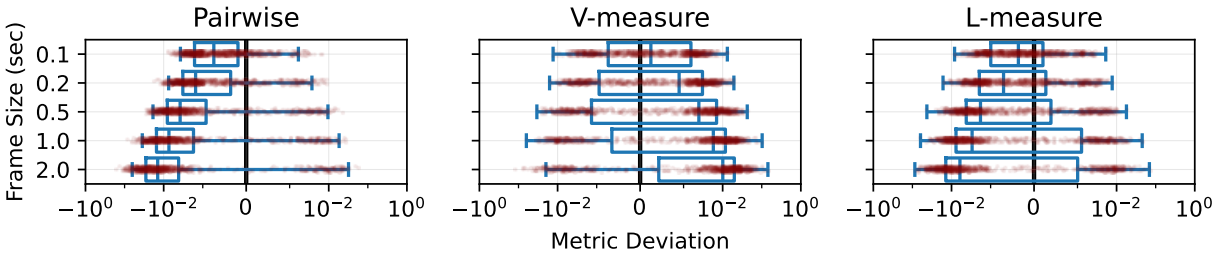

Figure 4. Deviation of frame-based metrics under different frame sizes from event-based metrics. The deviation is shown on a symmetric log scale, where negative (positive) value indicates under- (over-) predictions of the event-based score.

cies observed between our precise implementation and the sampling-based methods, which are sensitive to the frame size employed.

The results reveal systematic biases that vary by metric and frame size. Both PFC and L-measure tend to underpredict the true scores with increasing frame size, while the V-measure consistently inflated. Notice that while the average biases are systematic, the individual errors are two-sided; for any given frame size and metric, the deviations can be either positive or negative relative to the precise value, as shown in Figure 4

These opposing biases suggest that the reported ranking of different systems could change based on the chosen frame size, potentially undermining the reproducibility of comparative evaluations.

5.3 Scenario: Hyperparameter Tuning

To assess the practical impact on iterative research workflows, we evaluated the computational cost following the hyperparameter tuning process used to develop the Segment Fusion algorithm [15]. For one evaluation pass on the 471-track development set, using our event-based implementation took only 40 seconds. In stark contrast, the frame-based implementation of `mir_eval` (with the recommended 0.1-second frame size) required approximately 90 minutes to perform the same evaluation. This would mean that using our implementation during this development pipeline could save hundreds of hours of compute time, directly affecting the feasibility of certain research activities that depend on heavy evaluation.

We also found that this overhead is disproportionately affected by track duration; the ten longest tracks alone accounted for 22 minutes of the frame-based evaluation time, which provides an upper bound on how much efficiency could be gained by parallelism. Our event-based formulation avoids this dependency, offering not only significant computational savings but also enhancing the practicality of using large-scale datasets in modern MIR workflows.

5.4 L-measure and Depth

As noted in Section 3.4, the computational complexity of both event-based and frame-based implementations depends on hierarchies' depth. We confirm this empirically by showing average runtime for increasingly deeper estimated hierarchies against the same annotations in Figure 5.

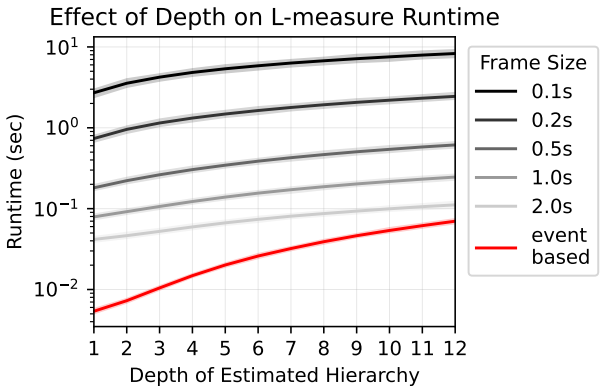

Figure 5. Runtime comparison of frame-based and event-based L-measure implementations as a function of hierarchy depth of $\hat{H}$. The y-axis shows the metric runtime in seconds (log scale), with shaded regions indicating 95% confidence interval.

6. DISCUSSION

We introduced an event-based paradigm for Music Structure Analysis (MSA) evaluation metrics. This new paradigm eliminates the arbitrary choice of frame size inherent in traditional frame-based approaches while drastically improving computational efficiency.

Our empirical evaluation using the SALAMI dataset demonstrated the substantial benefits of our event-based implementations. We showed significant gains in computational efficiency and, crucially, identified systematic biases present in frame-based metrics. These findings underscore that even seemingly minor choices regarding frame granularity can lead to non-trivial errors, potentially compromising evaluation integrity. While frame-based metrics can approach the accuracy of event-based counterparts with sufficiently small frame sizes, this precision comes at a considerable computational cost, severely limiting their utility in large-scale or iterative research workflows.

Looking ahead, our event-based paradigm can be readily extended to other evaluation metrics and broader MIR tasks. Adopting such an approach broadly within MIR evaluation frameworks will not only enhance accuracy and reproducibility, but will also foster the sustainable growth and scalability of future research efforts.

7. REFERENCES

[1] M. Levy and M. Sandler, "Structural segmentation of musical audio by constrained clustering," *IEEE transactions on audio, speech, and language processing*, vol. 16, no. 2, pp. 318–326, 2008.

[2] H. M. Lukashevich, "Towards quantitative measures of evaluating song segmentation." in *ISMIR*, 2008, pp. 375–380.

[3] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007, pp. 410–420.

[4] B. McFee, O. Nieto, M. M. Farbood, and J. P. Bello, "Evaluating hierarchical structure in music annotations," *Frontiers in Psychology*, vol. 8, p. 1337, 2017.

[5] 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 *ISMIR*, 2014, pp. 367–372.

[6] B. L. Sturm, "A simple method to determine if a music information retrieval system is a "horse"," *IEEE Trans. Multim.*, vol. 16, no. 6, pp. 1636–1644, 2014.

[7] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[8] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 61–65.

[9] V. Lostanlen and B. Mcfee, "Efficient evaluation algorithms for sound event detection," in *8th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2023)*, 2023.

[10] M. Buisson, B. McFee, S. Essid, and H. C. Crayencour, "Self-supervised learning of multi-level audio representations for music segmentation," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 32, pp. 2141–2152, 2024.

[11] C. J. Tralie and B. McFee, "Enhanced hierarchical music structure annotations via feature level similarity fusion," in *ICASSP*. IEEE, 2019, pp. 201–205.

[12] J. de Berardinis, M. Vamvakaris, A. Cangelosi, and E. Coutinho, "Unveiling the hierarchical structure of music by multi-resolution community detection," *Trans. Int. Soc. Music. Inf. Retr.*, vol. 3, no. 1, pp. 82–97, 2020.

[13] T. Chen, L. Su, and K. Yoshii, "Learning multifaceted self-similarity for musical structure analysis," in *APSIPA ASC*. IEEE, 2023, pp. 165–172.

[14] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. De Roure, and J. S. Downie, "Design and creation of a large-scale database of structural annotations." in *ISMIR*, vol. 11. Miami, FL, 2011, pp. 555–560.

[15] J. Salamon, O. Nieto, and N. J. Bryan, "Deep embeddings and section fusion improve music segmentation," in *ISMIR*, 2021, pp. 594–601.