BETWEEN: Boundary Estimation through Time Warping, Energy, & Entropy Neutralization

Calvin Breseman, Igor Moiseev, Bruno Abbate, and David M. Woollard
Standard AI
San Francisco, United States
Email: {calvin.breseman, igor.moiseev, bruno.abbate, david.woollard}@standard.ai

***Abstract*—We introduce BETWEEN—Boundary Estimation through Time-Warping, Energy, & Entropy Neutralization—an unsupervised framework for change point detection (CPD) with multidimensional spatiotemporal data. BETWEEN augments greedy binary segmentation with iterative revision and consolidation steps that steer partitions toward global optimality without the restrictive monotonic-additive assumptions of PELT[1] or FPOP[2]. Guided by principled limits on what CPD can resolve, we apply an adaptive Butterworth filter that trims the search space to genuine functional inflection points, slashing the effective workload by roughly a factor of twenty-five and yielding a comparable reduction in computational overhead relative to Binary Segmentation—even after BETWEEN's additional revision and consolidation passes. A novel gain function couples dynamic time warping, energy density ratios, and Kullback–Leibler divergence, allowing BETWEEN to recognize both subtle shape shifts and large distributional jumps while remaining agnostic to class labels, sampling rates, and segment length.**

Experiments on two multivariate spatiotemporal benchmarks, Human Activity Recognition (HAR) and Bee Waggle Dance, show improvements of 10-20% over state-of-the-art CPD baselines. Crucially, BETWEEN segments unstructured human and animal behavior streams into coherent, self-similar "tokens" that feed directly into downstream deep learning pipelines and allow for a graphical representation of complex real-world processes such as human behavior, enabling methodologies which can help examine the transitions between discrete behavioral states. By doing so, we re-frame CPD as an pre-processing step which reduces dimensionality, rather than an end goal itself.

***Index Terms*—change point detection, multivariate time series, unsupervised segmentation, dynamic time warping, behavioral boundary detection, spatiotemporal data.**

I. Introduction

In spatiotemporal data analysis, changepoint detection and temporal classification are often treated as distinct tasks, despite both relying on multidimensional time series. Automated changepoint detection is of the utmost importance in contexts where generated data volumes are extremely high, changepoints are rare, and the consequences of a missed changepoint can prove devastating (i.e., fields like medicine [1] or electrical power generation [2]). Meanwhile, spatiotemporal classification has gained prominence for analyzing human behavior, using data from sources such as smartphones, wearables, and pose estimation in video [3]. These classifiers detect actions like running or jumping from time series data. Yet, in any sequence with multiple labeled actions, classification inherently involves setting boundaries–implicitly detecting changepoints–especially when predictions are made at the frame or window level. Ensuring coherent transitions between actions often requires explicit post-processing.

As sources of rich spatiotemporal data become more numerous, there is reason to revisit the implicit separation of these techniques. Current paradigms of deep learning analyses on spatiotemporal data require extensive labeling of activity classes to enable fully-supervised learning pipelines. Models that have access to these rich labeled data perform very well, but it is difficult to generate labeled action data in an efficient and cost-effective manner; such labels typically require time-intensive human work. In addition, most activities that humans engage in occur in unstructured environments without clear boundaries between activity classes.

To frame our approach, we distinguish between **behaviors** and their **attributes**. In structured settings like orchestras, binary attributes–e.g., a musician is playing or no–can effectively characterize the behavior. In unstructured environments, however, this correspondence breaks down. Someone may be talking, eating, or sitting, but these attributes don't fully define what they are *doing*. In such settings, behaviors often span overlapping and evolving attributes that lack clear labels. This ambiguity challenges both classification and changepoint detection models. Yet, partitioning data into coherent segments remains essential for downstream modeling.

While some changepoint detection models such as Bayesian On-line Changepoint Detection (BOCPD) have similar aims, there are crucial assumptions and architectural decisions which reduce their efficacy in the envisioned application. The BOCPD family of models depend inherently on a deeply understanding the data; the distribution of the data points expected to be encountered, the average frequency of changepoints, the threshold for making decisions, and the data windows that the model sees. Hyperparameter tuning based on the performance of the model on a training dataset is almost always employed to set the model in an appropriate state for the domain and therefore is a fundamental challenge in the application of these approaches to unstructured environments where such knowledge cannot be assumed *a priori*.

Therefore, in unstructured environments where one intends to apply deep-learning frameworks to better understand spatiotemporal data, we propose a new process– Boundary Estimation through Time Warping, Energy, & Entropy

[1]PELT—Pruned Exact Linear Time
[2]FPOP—Functional Pruning Optimal Partitioning

Neutralization (BETWEEN) – for spatiotemporal data segmentation which inverts the typical framework, envisioning automated and efficient changepoint detection as a pre–not post–processing step to enhanced deep learning modeling on spatiotemporal data in unstructured environments. It is designed to partition unstructured spatiotemporal data into self-similar and externally dissimilar behavioral processes without manual labeling, with significantly reduced demands for specialized domain knowledge. At the end of the CPD process, a complex process like human behavior can easily be represented graphically, enabling numerous techniques for examining many dimensional spatiotemporal data.

We review key developments in changepoint detection and examine limitations in boundary detection for unstructured environments. We introduce our method, detailing data representation, segmentation, and cost functions, followed by a performance comparison on two benchmarks. We conclude with a discussion of future work in behavioral segmentation.

II. Related Work

CPD algorithms and methods such as greedy binary segmentation [4], Kernel-Based Change Detection [5], [6], Bayesian Online Changepoint Detection (BOCPD) [7], and PELT (Pruned Exact Linear Time) [8] have brought the field far beyond heuristics such as control thresholds on windowed moving averages. More recent work has applied deep learning methodologies such as Variational Autoencoders (VAEs) [9] and RNNs [10] to the problem of changepoint detection. Advancements in modern CPD techniques have greatly enhanced our ability to identify critical changes across a wide range of complex, real-world datasets.

As diverse multidimensional spatiotemporal data about the world has become increasingly available, researchers have applied CPD frameworks to detect changepoints outside of the domains where they were initially developed, tackling problems such as changes in human behavioral states and bee motion patterns with notable successes, but also with significant drawbacks to each algorithmic approach.

Bayesian Online Changepoint Detection (BOCPD) [7] assumes that each segment can be well-modeled by a simple parametric family (often Gaussian), and it treats changepoints as instantaneous "jumps" rather than gradual shifts. In practice, BOCPD also requires careful prior-specification and hyperparameter tuning of window sizes and even frequencies; decisions typically made by domain experts.

Pruned Exact Linear Time (PELT) [8] and FPOP [8] rely on additive cost functions (negative log-likelihood under a Gaussian assumption) and assume piecewise-constant parameters within each segment. These methods treat changepoints as abrupt and require that in-segment noise be homoscedastic or follow a tractible distribution. Their assumptions greatly limit the set of available cost functions.

Kernel-based change detection [5], [6] can handle non-Gaussian or non-linear data, but typically still assumes a sharp boundary between segments. Kernel bandwidth selection and window-size tuning often necessitate extensive experimentation or expert knowledge, aided by labeled data.

Recent deep learning approaches (e.g., Variational Autoencoder or RNN-based CPD) either rely on large volumes of labeled sequences to learn where boundaries lie [11] or training the RNN as a time series forecaster based on large volumes of domain data [10]. More than any other approach, they view CPD as a supervised task emerging from a set of known and labeled classes. While models like BOCPD make assumptions about the ways in which transitional points occur, trained approaches make fundamentally limiting assumptions about which transitions the model will be capable of detecting.

Researchers have begun to address some of these limitations in works such as Dynamic Interpretable Change Point Detection [12]. Even that approach, which does deal with spatiotemporal data and attempts to implement a solution which requires as little domain knowledge as possible, is hampered by the necessity of extensive hyperparameter tuning for application on the dataset of choice.

III. Algorithmic Implementation

In developing BETWEEN, we have reasoned from first principles about the nature of behavioral change. What does it mean for the physical expression of behavior to have changed in an unstructured environment, where each state might be ineffable? Posing this question has allowed us to provide reasonable hypotheses about how we measure aspects of changepoints in practice. Fundamentally, we think that boundaries ought to lie between changes in the "true" functional expression underlying a behavior. Implicitly, many CPD methods recognize this, with approaches that reduce to "fitting models to windows and checking when the best model changes " [7] [13] [11] [10] [1] [14]. We agree in principle, but think that approach has practical drawbacks if our aim is to recognize that the best functional representation has changed without fitting functions or proxies.

Instead, we crafted BETWEEN around uncovering differences in pattern, position, amplitude, and entropy. Through the use of z-normalized dynamic time warping, we search for changes in the basic pattern of the data. The energy density ratio simultaneously captures changes in position and amplitude. Finally, a Kullback-Leibler neutralization insists that any potential changepoint demonstrate evidence of a change in the underlying information.

We also made an initial assumption that all changes in behavior which are observable from spatiotemporal data will also be inflection points in functional representations of those time series. This assumption is grounded in an understanding of the target change points as transitional points between different states of behavior, even if those behaviors are complex. While humans can transition behaviors in a manner practically unobservable in spatiotemporal data (such as when a person stops reading and their mind wanders), this does not break our core assumption and in CPD based on time series, the only data on which one can make predictions are the time series themselves. This assumption serves to reduce the complexity of computations significantly, allowing for additions to the typical binary segmentation that improve performance and more closely approach global optimality.

A. Functional Representation

The functional representations of the data are created using adaptively tuned Butterworth filters. The right sampling frequency is imperative to the proper functioning of the segmentation algorithm, especially in datasets representing diverse underlying behaviors. Higher sampling frequencies preserve more of the underlying information but introduce additional computational overhead by generating a larger number of inflection points for evaluation. Too low a sampling frequency and the filter will remove information that might help determine the desired changepoints. BETWEEN automatically selects the cutoff frequency for the filter by analyzing the spectral density of the time series at regular intervals between $[.05, .95 * f_N]$. The Nyquist frequency is a standard measure of the maximum meaningful sampling frequency of a signal; the coefficient is employed to avoid edge effects.

B. Segmentation Algorithm

BETWEEN's segmentation algorithm is a variant of binary segmentation. Binary segmentation approaches have offered a solid compromise between algorithmic efficiency and evaluative performance on changepoint benchmarks. They work by iteratively partitioning segments of time series into distinct parts based on whether additional partitions offer improvements to the loss function [15]. Revised implementations such as PELT and FPOP achieve global optimality in $\mathcal{O}(n)$ time, but they rely on assumptions which constrain the sort of loss implementation that one can utilize, namely those of additive losses and monotonicity [8] [16]. BETWEEN utilizes a complex gain function for segmentation decisions, and as such, implementation of those algorithms is not possible.

BETWEEN's implementation of binary segmentation stands apart from others in a few key aspects. A major downside to greedy binary segmentation is the lack of a guarantee of global optimality on the partitioning solution; the optimal 3 segment solution to minimize the loss function might not have the 2 segment boundary as one of its two boundaries. To reach a solution closer to global optimality, BETWEEN introduces a revision stage every $\{n\}$ iterations of the greedy binary segmentation. For each of the candidate boundaries that have been agreed upon in prior steps, BETWEEN tests the two closest inflection points to that candidate. If either of them improves the cost function, the candidate boundary is shifted to that point. The order of evaluation for candidate boundary shifts is randomized to ensure that early candidates are not shifted more frequently than later ones.

Boundary-pair consolidation. After every r greedy iterations (Fig. 1, bottom) BETWEEN performs a local consolidation:

1) Let $\{c_{k-1}, c_k\}$ be two consecutive interior boundaries. Their enclosing endpoints are $a = c_{k-2}$ and $b = c_{k+1}$. Thus the current segmentation of $[a, b]$ is $[a, c_{k-1}]$, $[c_{k-1}, c_k]$, $[c_k, b]$.

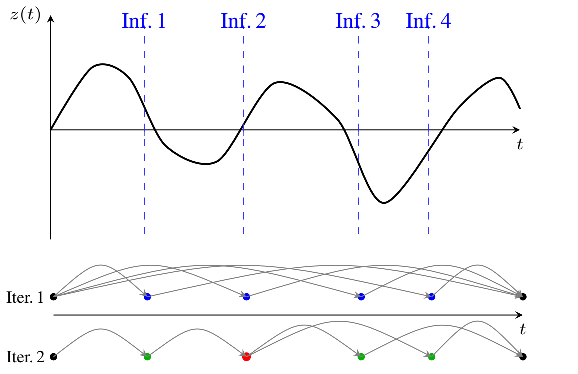

Fig. 1. Top: signal with four inflection candidates (blue dashed). Bottom: binary segmentation *Iter 1* scores every split ($0 \to \text{Inf}_j$ and $\text{Inf}_j \to T$); *Iter 2* selects Inflection 2 (red) and evaluates only splits within $[0, \text{Inf}_2]$ and $[\text{Inf}_2, T]$ (green).

2) Compute the $L_{\text{current}} = L_{[a,c_{k-1}]} + L_{[c_{k-1},c_k]} + L_{[c_k,b]}$.
3) Let $\mathcal{I}_{ab} \subset (a, b)$ be the ordered set of inflection candidates inside $[a, b]$. For every ordered pair $(i_L, i_R) \in \mathcal{I}_{ab} \times \mathcal{I}_{ab}$ with $a < i_L < i_R < b$ we compute the *replacement loss* $L_{\text{test}} = L_{[a,i_L]} + L_{[i_L,i_R]} + L_{[i_R,b]}$.
4) If $L_{\text{test}} < L_{\text{current}}$ for any pair (i_L, i_R), replace $\{c_{k-1}, c_k\}$ with the best-scoring pair $\{i_L^\star, i_R^\star\}$. If no such pair improves the loss, the original boundaries are kept.

Because adjusting $\{c_{k-1}, c_k\}$ changes the three segments they delimit, the exterior losses $L([a, c_{k-1}])$ and $L([c_k, b])$ *must* be recomputed; it is not enough to examine only the interior segment. Empirically, the number of candidates $|\mathcal{I}_{ab}|$ is small ($\approx 3\%$ of frames), so the $\mathcal{O}(|\mathcal{I}_{ab}|^2)$ scan is inexpensive. This consolidation lets BETWEEN escape local optima that greedy BinSeg alone would keep. Baronowski et. al take a similar approach in their 2016 paper "Narrowest-Over-Threshold Change-point Detection", but as far as the authors of this paper are able to tell, the consolidation stage is a novel contribution to the traditional binary segmentation algorithm.

C. Cost Function

BETWEEN maximizes a gain function based on the difference between the product of the energy density ratio and the normalized dynamic time warping distance between two segments and a ratio of the squared median KL divergence over the segments' specific KL divergence.

Dynamic Time Warping (DTW) is an algorithm designed to allow distance comparisons between two time series that may differ in scale or sampling frequency; indeed, even within-window frequency variations are permitted. Formally, let: $\mathbf{s}^{(a)} = \left(s_1^{(a)}, s_2^{(a)}, \ldots, s_{n_a}^{(a)}\right)$ and $\mathbf{s}^{(b)} = \left(s_1^{(b)}, s_2^{(b)}, \ldots, s_{n_b}^{(b)}\right)$ be two real-valued sequences of lengths n_a and n_b. A warping path $\pi = ((i_1, j_1), (i_2, j_2), \ldots, (i_K, j_K))$ is a sequence of index pairs satisfying:

$$(i_1, j_1) = (1, 1), \quad (i_K, j_K) = (n_a, n_b),$$
$$(i_{k+1}, j_{k+1}) \in \{ (i_k + 1, j_k), (i_k, j_k + 1), (i_k + 1, j_k + 1) \}$$

where $k = 1, \ldots, K-1$, $i_1 \leq i_2 \leq \cdots \leq i_K$ and $j_1 \leq j_2 \leq \cdots \leq j_K$. Given a local point-wise dissimilarity $d\big(s_i^{(a)}, s_j^{(b)}\big)$ (we use the Euclidean norm), the cost of a warping path π is $\mathrm{Cost}(\pi) = \sum_{(i,j)\in\pi} \big\|\widetilde{s}_i^{(a)} - \widetilde{s}_j^{(b)}\big\|_2$. The dynamic-time-warping (DTW) distance between the two sequences is the minimum such cost over all admissible paths:

$$\mathrm{DTW}_{\text{raw}}\big(\mathbf{s}^{(a)}, \mathbf{s}^{(b)}\big) \;=\; \min_{\pi} \sum_{(i,j)\in\pi} \big\|\widetilde{s}_i^{(a)} - \widetilde{s}_j^{(b)}\big\|_2 \qquad (1)$$

We utilize the `fastDTW` Python package to compute a near-optimal $\mathrm{DTW}_{\text{raw}}$ in $\mathcal{O}\big(\max(n_1, n_2)\big)$ time. Because BETWEEN is maximizing a gain function over the entire time series T, we keep $\mathrm{DTW}_{\text{raw}}$ in its raw (absolute) form rather than normalizing by sequence length. Normalization would, counterintuitively, reward shorter segments and allow "reward hacking" by alternating very short and very long segments.

Interpolation and Length Heuristic. Evaluated segment boundaries τ may yield subsequences of widely varying lengths. To compare two segments $\mathbf{s}^{(a)}$ and $\mathbf{s}^{(b)}$ of lengths n_a and n_b, we resample in fixed order so that both become sequences of equal interpolated length ℓ. Denote $M = \max\{n_a, n_b\}$, $\ell = \min\big(\ell_{\max}, \lfloor \frac{M}{2} \rfloor\big)$, where $\ell_{\max}$ is a user-specified constant (e.g., $\ell_{\max} = 4000$). Choosing $\ell = \lfloor M/2 \rfloor$ rather than $\ell = M$ prevents artificially inflating $\mathrm{DTW}_{\text{raw}}$. For example, if consecutive segments have lengths $n_a = 20$ and $n_b = 200$, then a naive interpolation to $\ell = 200$ would force evaluations over 400 interpolated indices, rather than the true total of 220 observations in the original sequence. By halving the maximum length, the effective DTW domain is restricted to roughly the true combined span, which prevents the gain function from "hacking" via alternating extremely long and extremely short segments. (A Taylor-series analysis of such oscillations shows that choosing $\ell = M$ introduces a multiplier proportional to $\sqrt{n_b/n_a}$ in the gap between short- and long-segment DTW scores, which $\lfloor M/2 \rfloor$ mitigates.)

Once ℓ is chosen, we define interpolation grids

$$t_a = \big\{0,\, \tfrac{1}{n_a - 1},\, \tfrac{2}{n_a - 1},\, \ldots,\, 1\big\},$$
$$u = \big\{0,\, \tfrac{1}{\ell - 1},\, \tfrac{2}{\ell - 1},\, \ldots,\, 1\big\}$$

and similarly for t_b. We then obtain $\widetilde{\mathbf{s}}^{(a)} \in \mathbb{R}^{\ell}$ by

$$\widetilde{s}_k^{(a)} = \text{Interpolate}\Big(\mathbf{s}^{(a)},\, t_a,\, u_k\Big), \quad k = 1, \ldots, \ell$$

and likewise for $\widetilde{\mathbf{s}}^{(b)}$. The raw DTW score on interpolated sequences is $\mathrm{DTW}_{\text{raw}}\big(\widetilde{\mathbf{s}}^{(a)}, \widetilde{\mathbf{s}}^{(b)}\big)$.

To temper the bias towards shorter segments and ensure that short segments represent definitive pattern shifts, we then scale by a sigmoid-shaped factor. Let

$$\sigma(x) = \frac{1}{1 + \exp\big(-s(x - m)\big)}, \quad m = 15, \quad s = 0.1,$$

where $m = 15$ is a heuristic midpoint (e.g., for a 10 Hz sensor, by 30 frames the sigmoid is effectively 1, reflecting that $\mathrm{Var}\big(\mathbf{s}\big)$ stabilizes by $n \approx 30$). Define

$$\text{scale\_factor} \;=\; \min\big(\sigma(n_a),\, \sigma(n_b)\big),$$

and set

$$\mathrm{DTW}_{\text{final}} \;=\; \mathrm{DTW}_{\text{raw}}\big(\widetilde{\mathbf{s}}^{(a)},\, \widetilde{\mathbf{s}}^{(b)}\big) \;\times\; \text{scale\_factor}.$$

Minimum-Window Constraint. Because DTW and KL-divergence estimates become unstable when a segment has too few frames, we enforce a hard lower bound of $n_{\min} = 5$ frames per segment. Empirically, when $n < 5$, $\mathrm{KL}\big(\mathbf{s}^{(a)} \,\|\, \mathbf{s}^{(b)}\big)$ is numerically unstable, and the DTW distance is driven toward zero by the sigmoid based length scaler for segments of that or lesser length.

IV. Energy Density Considerations

The second positive component of the function is formed by the symmetric ratio between the per-frame average squared magnitude of the underlying kinematic or positional signals over time, capturing both relative measures of activity intensity as well as the "position" of that activity within the graph.

Formally, suppose a single segment is composed of F dimensional observations $\mathbf{x}_t = \big(x_{t,1},\, x_{t,2},\, \ldots,\, x_{t,F}\big)$ represented by the time series

$$\mathbf{X} \;=\; \{\, \mathbf{x}_t \in \mathbb{R}^F : t = 1, \ldots, T \},$$

We define the *energy density* of $\mathbf{X}$ by

$$E\big(\mathbf{X}\big) = \frac{1}{T} \sum_{t=1}^{T} \sum_{f=1}^{F} x_{t,f}^2 \qquad (2)$$

We square each channel value at every time step, sum over all F channels and over all T timestamps, and finally normalize by the segment length T. This yields a single scalar reflecting the overall "power" or "activity level" inside that window.

Because BETWEEN is particularly interested in how two *adjacent* segments differ in motion intensity near their shared boundary, we employ a *boundary-weighted* energy density ratio. Let c denote a candidate boundary at time index t_0. We select a small buffer of at most M frames before and after t_0, and assign each frame τ a weight $w(\tau)$ so that observations closer to t_0 contribute more heavily. Specifically, we define

$$W_{\text{full}} = 3f_s, \qquad (3)$$
$$W_{\text{decay}} = 3f_s, \qquad (4)$$
$$M = W_{\text{full}} + W_{\text{decay}} = 6f_s, \qquad (5)$$

where f_s is the sampling frequency of the time series in frames per second (Hz). In practice, for human spatiotemporal behaviors, one rarely needs to look more than 3 seconds (as humans typically complete behavioral transitions within a few seconds) with full weight, and the cosine taper then extends for an additional 3 seconds before the weight goes to zero entirely in order to avoid abrupt cutoff. Thus $M = 6f_s$ frames corresponds to a maximum 6-second buffer (so that at $|\tau - t_0| = 6f_s$, $w(\tau) = 0$).

We assign each frame τ a weight

$$w(\tau) = \begin{cases} 1, & |\tau - t_0| \le W_{\text{full}}\,, \\ \dfrac{1+\cos\!\Big(\frac{\pi\,(|\tau-t_0|-W_{\text{full}})}{W_{\text{decay}}}\Big)}{2}, & W_{\text{full}} < |\tau - t_0| \le M\,, \\ 0, & |\tau - t_0| > M\,. \end{cases} \tag{6}$$

Because $W_{\text{full}} = 3f_s$ and $W_{\text{decay}} = 3f_s$, the weight decays smoothly from 1 to 0 over the interval $(3f_s, 6f_s]$. In particular,

$$\big|\tau - t_0\big| \;=\; W_{\text{full}} + \frac{W_{\text{decay}}}{2} \;=\; 3f_s \;+\; \frac{3f_s}{2} = 4.5f_s,$$

$w(\tau) \;=\; \frac{1}{2}\big[1+\cos(\frac{\pi}{2})\big] \;=\; \frac{1}{2}$. For example, if $f_s = 50$Hz, then $4.5f_s = 225$ frames—i.e. at 225 frames away from the boundary, the contribution is 50%. By 300 frames ($6f_s$), $w(\tau) = 0$.

Because frames with $|\tau - t_0| > M$ receive zero weight, it is equivalent (and clearer) to restrict all sums to the index set

$$\mathcal{N}_{t_0} \;=\; \big\{\tau \;\big|\; |\tau - t_0| \;\le\; M\big\} \,\cap\, \{1,2,\ldots,T\}.$$

Hence the *boundary-weighted energy density* can be written

$$E_{\text{bw}}\big(\mathbf{X};\, t_0\big) \;=\; \frac{1}{\sum_{\tau\in\mathcal{N}_{t_0}} w(\tau)} \sum_{\tau\in\mathcal{N}_{t_0}} w(\tau) \sum_{f=1}^{F} x_{\tau,f}^2 \tag{7}$$

By focusing energy computation on frames immediately surrounding the boundary, BETWEEN more robustly detects changes in motion intensity that coincide with a shift from one behavioral phase to another. Note that because $w(\tau) = 1$ for $|\tau - t_0| \le 3f_s$, the central $\pm 3f_s$ frames contribute fully; the cosine taper over the next $3f_s$ frames smoothly attenuates the contribution to 0 at $|\tau - t_0| = 6f_s$. In practical terms, if a human truly changes behavior at t_0, by 9 seconds later ($9f_s$ frames in total time), the segment's kinematic pattern will have shifted measurably, so there is no justification for reaching past $6f_s$ frames when evaluating energy near t_0.

Sigmoid-Based Dampening for Short Segments. As noted above, when either adjacent segment is very short ($T < 5$ frames), the energy computation can be unstable and the raw energy ratio may blow up. To avoid spurious large ratios when one segment is too short, we introduce a *length-based dampening* that smoothly returns the energy ratio to 1 whenever either segment is near or below a minimal length. Specifically, define the *sigmoid function*

$$\sigma(n) = \frac{1}{1+\exp\!\big(-0.1(n-15)\big)} \tag{8}$$

where $n =$ segment length in frames. Because $\sigma(n) \approx 0$ when $n \le 1$, and $\sigma(30) \approx 0.99$, by 30 frames the sigmoid is effectively saturated at 1.0. Now, suppose two neighboring segments $\mathbf{X}_{\text{L}}$ ("left") and $\mathbf{X}_{\text{R}}$ ("right") have lengths n_{L} and n_{R}, respectively. Define the *raw energy ratio*

$$R_{\text{raw}} = \frac{E_{\text{bw}}\big(\mathbf{X}_{\text{L}};\, c\big)}{E_{\text{bw}}\big(\mathbf{X}_{\text{R}};\, c\big)} \quad \text{or} \quad \frac{E_{\text{bw}}\big(\mathbf{X}_{\text{R}};\, c\big)}{E_{\text{bw}}\big(\mathbf{X}_{\text{L}};\, c\big)},$$

whichever exceeds 1. Let

$$L = \min\!\big(\sigma(n_{\text{L}}), \sigma(n_{\text{R}})\big),$$

and define the *dampened energy ratio*

$$R_{\text{damp}} = \big(1 - L\big)\cdot 1 + L\,R_{\text{raw}}. \tag{9}$$

In other words, R_{damp} is a convex combination of the identity value 1 and the raw ratio R_{raw}, with the weight L. When both segments have at least 30 frames, $\sigma(n) \approx 1$, so $L \approx 1$ and $R_{\text{damp}} \approx R_{\text{raw}}$. When one segment has fewer than ≈ 15 frames, $\sigma(n) \ll 1$, so $L \ll 1$ and $R_{\text{damp}} \approx 1$, preventing any spurious large ratio.

We intentionally use the *uncentered* energy in Eq.(2) rather than variance or mean-centered energy. In spatial data, a change in "position" (i.e., accelerating upward when sitting $\rightarrow$ standing) will shift *all* channels by a roughly constant offset. If we had subtracted the mean first, that "static" tilt or shift in position might vanish. By retaining uncentered squared magnitudes, BETWEEN captures both changes in *motion intensity* and changes in *absolute displacement*. Further, a boundary without an $R_{\text{damp}>1}$ is never precluded; In short:

$$E(\mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T}\sum_{f=1}^{F} x_{t,f}^2$$

reflects both "activity" (variance) and "position" (mean).

Length-Scaling Sigmoid for DTW Penalty. For completeness, recall that BETWEEN also uses a sigmoid of identical form to Eq.(8) when scaling DTW distances by segment length. If a segment has length n, we define

$$\ell(n) = \frac{1}{1+\exp\!\big(-0.1(n-15)\big)} = \sigma(n). \tag{10}$$

Since the midpoint and steepness parameters are the same in Eq.(8) and Eq.(10), both sigmoid functions share identical behavior. In particular, $\ell(n) \approx 0$ when $n \le 1$ and $\ell(n) \approx 1$ when $n \ge 30$. When computing the *DTW scaling factor* for two segments of lengths n_a, n_b, and denoting

$$\text{scale\_factor} = \min\!\big(\,\ell(n_a),\; \ell(n_b)\,\big)$$

we scale the raw DTW distance by scale_factor

$$\text{DTW}_{\text{final}} \;=\; \text{DTW}_{\text{raw}} \;\times\; \text{scale\_factor}.$$

Thus, when either segment is very short ($n < 15$), the DTW penalty vanishes, and when both segments exceed 30 frames, the DTW distance is unscaled.

A. KL Divergence as a Penalty Term in BETWEEN

The maximization of cross-boundary entropy has been well established in change-point detection (CPD) research for years. In particular, BOCPD (Bayesian Online Change-Point Detection), one of the most commonly employed and high-performing forms of CPD, is inherently founded on maximizing entropy differences between the distributions formed by the points on either side of a boundary. Summing across

run-lengths $\ell = 0, 1, \ldots, \min(t-1, \ell_{\max})$, BOCPD updates the probability that a change-point occurred at time t with run-length ℓ (i.e., ℓ frames since the last change-point) in a Bayesian manner; this probability is obtained by normalizing two unnormalized "masses" (growth vs. change) for each ℓ.
Growth mass:

$$\begin{aligned} G_t(\ell) \;&=\; \Pr\bigl(r_{t-1} = \ell \mid x_{1:t-1}\bigr)\,\bigl(1 - H(\ell)\bigr) \\ &\quad \times p\bigl(x_t \mid r_{t-1} = \ell,\; x_{t-\ell:t-1}\bigr). \end{aligned}$$

Change mass:

$$C_t(\ell) = \Pr\bigl(r_{t-1} = \ell \mid x_{1:t-1}\bigr)\; H(\ell)\; p\bigl(x_t \mid \text{new-run prior}\bigr),$$

where:

- $\Pr\bigl(r_{t-1} = \ell \mid x_{1:t-1}\bigr)$ is the posterior probability that the current segment began ℓ frames ago,
- $(1 - H(\ell))$ is the probability of "no change" (i.e. no new change-point) at run-length ℓ, and $H(\ell)$ is the baseline (hazard) probability of a change occurring exactly at run-length ℓ,
- $p\bigl(x_t \mid r_{t-1} = \ell,\; x_{t-\ell:t-1}\bigr)$ is the predictive likelihood of x_t under the existing segment that started ℓ frames before, and
- $p\bigl(x_t \mid \text{new-run prior}\bigr)$ is the prior predictive likelihood if x_t starts a brand-new segment.

Once all growth and change masses for $\ell = 0, \ldots, \min(t-1, \ell_{\max})$ have been computed, BOCPD normalizes them so that

$$\sum_{k=0}^{\min(t,\ell_{\max})} \Pr\bigl(r_t = k \mid x_{1:t}\bigr) \;=\; 1.$$

Because BOCPD typically assumes a conjugate-exponential family (e.g., Gaussian likelihood with a Normal–Inverse-Gamma prior), only fixed-dimension sufficient statistics (counts, sums, sums of squares, etc.) need to be carried forward for each candidate ℓ to compute each predictive likelihood. However, there are also versions of BOCPD which replace the parametric predictive model with an empirical distribution over the last ℓ observations. [7] In those cases, the update at each time t is exactly the asymmetric KL divergence between the empirical distribution formed by the previous ℓ points and the prior predictive. Even in the exact Bayesian recursion, the single-step log-likelihood ratio

$$\begin{aligned} \Delta_t(\ell) =& \log p\bigl(x_t \mid r_{t-1} = \ell,\; x_{t-\ell:t-1}\bigr) \\ &- \log p\bigl(x_t \mid \text{new-run prior}\bigr) \end{aligned}$$

serves as a measure of relative entropy between "continue this run" and "start a new run." In that sense, the Bayesian updating process ultimately converges to a sum of log-likelihood ratios (an asymmetric KL) over candidate ℓ, which aligns the total change-point probability with the cumulative evidence for a distributional shift between adjacent windows.

In theory, any true changepoint in a time series can be captured as a cross-boundary entropy maximization, given the right parametric model [17]. BOCPD operates by fitting parametric models to segments since the last boundary and instantiating a new boundary when the new model explains incoming data significantly better than the old one. For example, if a model can distinguish "running" from "walking," BOCPD will favor the model that better fits new data. However, this highlights a key issue: achieving strong performance often requires building detailed activity-specific models—essentially turning changepoint detection into a classification task under a different name. Thus, although BOCPD's use of

$$D_{\mathrm{KL}}\bigl(p_{\mathcal{S}_L} \,\|\, p_{\mathcal{S}_R}\bigr)$$

is theoretically valid—since for any two empirical windows one can fit Gaussian densities to compute KL divergence—the practical burden of specifying and fitting those distributions can be prohibitive, especially if the set of activities becomes unknown. To avoid these issues while recognizing the theoretical importance of entropy changes at boundary points, BETWEEN approaches the situation as an entropy neutralization problem, and treats cross-boundary KL divergence as a *necessary but not sufficient* condition: only if

$$D_{\mathrm{KL}}\bigl(\mathcal{S}_L \,\|\, \mathcal{S}_R\bigr) \;>\; \mathrm{KL}_{\mathrm{baseline}}$$

do we incur the cost of evaluating the energy- and DTW-based pattern-change terms. Thereafter, chosen splits must maximize the divergence between the DTW-Energy Ratio product and the KL penalty term. Define

$$\mathrm{KL}_{\mathrm{penalty}}(b) \;=\; \frac{\bigl(\mathrm{KL}_{\mathrm{baseline}}\bigr)^2}{\bigl(D_{\mathrm{KL}}(\mathcal{S}_L \,\|\, \mathcal{S}_R) + 10^{-6}\bigr) \times \mathrm{LSF}(b)}, \quad (11)$$

where $\mathcal{S}_L$ and $\mathcal{S}_R$ are the two segments left and right of boundary b, and:

- $D_{\mathrm{KL}}(\mathcal{S}_L \,\|\, \mathcal{S}_R)$ is the symmetric KL divergence between Gaussian fits to $\mathcal{S}_L$ and $\mathcal{S}_R$,
- $\mathrm{KL}_{\mathrm{baseline}}$ is computed at step 0 and
- $\mathrm{LSF}(b)$ is a length-scaling factor based on segment lengths n_L and n_R (see earlier commentary).

As the distributional difference between $\mathcal{S}_L$ and $\mathcal{S}_R$ increases, the denominator grows, driving $\mathrm{KL}_{\mathrm{penalty}}(b)$ toward zero. We choose to square the median baseline in the numerator as a first pass—empirically this gives good segmentation—though future work could explore a gradnorm approach to balance the KL term against DTW more precisely.

1) Estimating $\mathrm{KL}_{\mathrm{baseline}}$*:* We compute $\mathrm{KL}_{\mathrm{baseline}}$ via random segmentation sampling on the entire time series $X = \{x_1, \ldots, x_T\}$. Using $N_{\mathrm{iter}} = 500$ iterations and $m = 10$ random boundary points per iteration (with minimum segment length $n_{\min} = 5$), the procedure is as follows.

We compute $\mathrm{KL}_{\mathrm{baseline}}$ by sampling random segment boundaries and aggregating the resulting symmetric KL divergences between adjacent segments. Let $X = \{x_1, \ldots, x_T\}$ be the full time-series, and fix

$$N_{\mathrm{iter}} = 500, \quad m = 10, \quad n_{\min} = 5.$$

Each iteration $n = 1, \ldots, N_{\mathrm{iter}}$ proceeds as follows:

1. Randomly draw m distinct indices $\{i_1, \dots, i_m\} \subset \{1, \dots, T\}$, then sort them so that

$$1 < i_{(1)} < i_{(2)} < \cdots < i_{(m)} < T.$$

2. Define the boundary sequence

$$b_0 = 1, \quad b_j = i_{(j)} \quad (j = 1, \dots, m), \quad b_{m+1} = T.$$

3. For each $j = 1, \dots, m$, form the two adjacent segments $\mathcal{S}_L = \{x_{b_{j-1}}, x_{b_{j-1}+1}, \dots, x_{b_j-1}\}$, $\mathcal{S}_R = \{x_{b_j}, x_{b_j+1}, \dots, x_{b_{j+1}-1}\}$. Only if $|\mathcal{S}_L| \geq n_{\min}$ and $|\mathcal{S}_R| \geq n_{\min}$, compute the symmetric KL divergence

$$d = D_{\mathrm{KL}}(\mathcal{S}_L \,\|\, \mathcal{S}_R),$$

and include d in $\mathcal{D}_{\text{all}}$ only if $d > 0$ and d is finite.
4. Repeat steps 1–3 for all $n = 1, \dots, N_{\text{iter}}$.

After all iterations, let $\mathcal{D}_{\text{all}}$ denote the multiset of all collected KL values. We then define

$$\widehat{m} = \operatorname{median}(\mathcal{D}_{\text{all}}), \qquad \mathrm{KL}_{\text{baseline}} = \max\{\widehat{m}, 10^{-6}\}.$$

Because $\mathcal{D}_{\text{all}}$ is typically right-skewed and heavy-tailed, using the median ensures that the square of $\mathrm{KL}_{\text{baseline}}$ reflects values in the far right tail of boundary divergences without having to directly specify the percentile we wish to cut KL divergences off at; if a series has a higher proportion of extremely high boundary divergences, that points to more changepoints, so we don't want to artificially limit the set of potential boundaries. Moreover, with $N_{\text{iter}} = 500$ and $m = 10$, the resulting $\mathrm{KL}_{\text{baseline}}$ is empirically very stable, and the sampling overhead is negligible compared to the remainder of the segmentation pipeline.

In practice, 500 iterations yield a stable median with negligible variation, and the sampling overhead is trivial compared to the overall segmentation.

2) Gaussian KL Computation: For two segments $\mathcal{S}_L, \mathcal{S}_R$, let n_L, n_R be their frame counts and let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{T \times F}$ be their F-dimensional feature arrays after differencing. If either $n_L < 2$ or $n_R < 2$ (or $F = 0$), we take $D_{\mathrm{KL}} = 0$. Otherwise, we extract boundary windows of up to $W_{\max} = \lfloor 10\, f_s \rfloor$ frames closest to the boundary:

$$W_L = \min\{W_{\max}, n_L - 1\}, \quad W_R = \min\{W_{\max}, n_R - 1\}.$$

We let $\widetilde{\mathbf{A}}$ be the last W_L rows of $\mathbf{A}$, and $\widetilde{\mathbf{B}}$ be the first W_R rows of $\mathbf{B}$. We then interpolate each column to length $\ell = \min\{\ell_{\max}, \max(W_L, W_R)\}$ (with $\ell_{\max} = 4000$ by default). After zero-centering and scaling by either provided global standard deviations or by $\max\{\operatorname{std}(\widetilde{\mathbf{A}}[:, j], \widetilde{\mathbf{B}}[:, j]), 10^{-9}\}$ per feature, we compute means μ_L, μ_R and covariances Σ_L, Σ_R. We add a regularization

$$\varepsilon = \max\Big\{10^{-8}, 10^{-6} \min\big(\operatorname{tr}(\Sigma_L), \operatorname{tr}(\Sigma_R)\big)\Big\}$$

to each diagonal, then compute:

$$d_{LR} = \mathrm{KL}\big(\mathcal{N}(\mu_L, \Sigma_L) \,\|\, \mathcal{N}(\mu_R, \Sigma_R)\big),$$
$$d_{RL} = \mathrm{KL}\big(\mathcal{N}(\mu_R, \Sigma_R) \,\|\, \mathcal{N}(\mu_L, \Sigma_L)\big).$$

If either determinant is non-positive or any value is non-finite, we set $D_{\mathrm{KL}} = 0$. Otherwise, $D_{\mathrm{KL}}(\mathcal{S}_L \,\|\, \mathcal{S}_R) = \max\{0, d_{LR} + d_{RL}\}$.

3) Integration into BETWEEN's Gain Function: Within BETWEEN's gain function, each candidate boundary b that splits a segment into two subsegments of lengths n_L and n_R is evaluated by combining distributional, pattern, and energy information. Formally, we define:

$$\mathrm{Gain}(b) = \Delta_{\text{energy}}(b) - \lambda_{\text{kl}}\, \mathrm{KL}_{\text{penalty}}(b),$$

where $\Delta_{\text{energy}}(b)$ captures the change-in-energy and DTW-based pattern-change terms when splitting at b, $\lambda_{\text{kl}} > 0$ is a scalar weight balancing the KL penalty, and

$$\mathrm{KL}_{\text{penalty}}(b) = \frac{\big(\mathrm{KL}_{\text{baseline}}\big)^2}{\big(D_{\mathrm{KL}}(\mathcal{S}_L \,\|\, \mathcal{S}_R) + 10^{-6}\big) \times \mathrm{LSF}(b)},$$

using $\mathrm{KL}_{\text{baseline}}$ as above and $\mathrm{LSF}(b) = \min\big(\sigma(n_L), \sigma(n_R)\big)$, where $\sigma(n)$ is defined by Eq.(8), so that segments shorter than 5 frames yield nearly zero weight and segments above 30 frames yield weight near 1.

A candidate b^* is accepted if $\mathrm{Gain}(b^*) > 0$ and maximizes the net reduction in loss. This ensures that a split is retained only if its distributional divergence is both statistically significant relative to random baselines and accompanied by a meaningful energy/pattern change.

Gain Function Summary

The gain at a candidate boundary b decomposes into:

1) KL-Penalty:

$$\mathrm{KL}_{\text{penalty}}(b) = \frac{\big(\mathrm{KL}_{\text{baseline}}\big)^2}{\big(D_{\mathrm{KL}}(\mathcal{S}_L \,\|\, \mathcal{S}_R) + 10^{-6}\big)\, \mathrm{LSF}(b)}. \tag{12}$$

where $\mathrm{LSF}(b) = \min\{\sigma(n_L), \sigma(n_R)\}$, $\quad \sigma(n) = \frac{1}{1+\exp\big(-0.1\,(n-15)\big)}$.

2) DTW Distance:

$$\mathrm{DTW}_{\text{final}}(b) = \mathrm{DTW}_{\text{raw}}\big(\widetilde{\mathbf{s}}^{(L)}, \widetilde{\mathbf{s}}^{(R)}\big) \times \min\{\ell(n_L), \ell(n_R)\}. \tag{13}$$

3) Weighted Energy Density Ratio:

$$R_{\text{raw}}(b) = \max\Big(\frac{E_{\text{bw}}(\mathcal{S}_L; b)}{E_{\text{bw}}(\mathcal{S}_R; b)}, \frac{E_{\text{bw}}(\mathcal{S}_R; b)}{E_{\text{bw}}(\mathcal{S}_L; b)}\Big), \tag{14}$$

$$R_{\text{damp}}(b) = \big(1 - L(b)\big) + L(b)\, R_{\text{raw}}(b), \tag{15}$$

with

$$\ell(n) = \sigma(n), \quad L(b) = \min\{\sigma(n_L), \sigma(n_R)\},$$

and

$$E_{\text{bw}}(\mathcal{S}; b) = \frac{1}{\sum_\tau w(\tau)} \sum_\tau w(\tau) \sum_f x_{\tau,f}^2.$$

Combined Gain:

$$\mathrm{Gain}(b) = \mathrm{DTW}_{\text{final}}(b)\, R_{\text{damp}}(b) - \lambda_{\text{kl}}\, \mathrm{KL}_{\text{penalty}}(b). \tag{16}$$

When both $\mathrm{DTW}_{\text{final}}$ is large (indicating a functional pattern change) and R_{damp} is high (indicating an energy-density shift), the left hand term spikes, signaling a strong candidate for a true behavioral change-point. Finally, $\mathrm{Gain}(b)$ subtracts the scaled KL penalty, ensuring that only splits with sufficiently large distributional divergence *and* meaningful pattern/energy changes survive.

V. Computational Complexity

The gain function over greedy binary segmentation iterations and revisions is the primary contributor to overall complexity. Let T be the total number of frames, F the feature dimension ($F \ll T$), f_s the sampling rate, $W = \lfloor 10 f_s \rfloor$ on two-second window, $\ell_{\max} = 4000$, both treated as constants.

A. Baseline KL Estimation

Compute $\mathrm{KL}_{\text{baseline}}$ via $N_{\text{iter}} = 500$ random segmentations with $m = 10$ boundaries each. Each evaluation of symmetric KL on two $W \times F$ windows costs $\mathcal{O}\big(WF^2 + F^3\big)$ and there are 500×10 such evaluations (both constants). Hence the one-time setup is $\mathcal{O}(F^3)$, negligible for small F.

B. Per-Boundary Evaluation

For a split inside an interval of length n, we perform: 1. *KL screening* on two W-frame windows: $\mathcal{O}(WF^2 + F^3)$. 2. *DTW pattern change*: differencing and z-normalization cost $\mathcal{O}(nF)$; interpolating to $\ell = \mathcal{O}(1)$ and fastDTW cost $\mathcal{O}(1)$. 3. *Energy change*: extracting an $M = 6f_s = \mathcal{O}(1)$-frame buffer and computing weights costs $\mathcal{O}(F)$. Therefore, each candidate costs $\mathcal{O}\big(nF + F^3\big)$.

C. Greedy Segmentation with Caching

- **Iterations 1–2 (no cache).** A segment of length n has $n-1$ candidate splits. Evaluating each costs $\mathcal{O}(nF + F^3)$, so one full pass costs
$$\sum_{b=1}^{n-1} \mathcal{O}(nF + F^3) \;=\; \mathcal{O}\big(n^2 F + nF^3\big).$$
On the initial segment ($n = T$), that is $\mathcal{O}(T^2F + TF^3)$.
- **Iteration $k \geq 3$ (with cache).** Suppose an interval $[a, b]$ of length N is split at $i_* \in (a, b)$ into lengths $n' = i_* - a$ and $n'' = b - i_*$. We must recompute splits in:
  - $[a, i_*]$ (cost $\mathcal{O}(n'^2F + n'F^3)$),
  - $[i_*, b]$ (cost $\mathcal{O}(n''^2F + n''F^3)$),
  - the adjacent interval $[p, a]$ if its right neighbor changed from $[a, b]$ to $[a, i_*]$,
  - the adjacent interval $[b, q]$ if its left neighbor changed from $[a, b]$ to $[i_*, b]$.

  In particular, any interval whose neighbor boundary has moved requires recomputation for all its splits. Empirically, by iteration 10 only $\approx 30\%$ of the original splits remain active, and by iteration 30 $\approx 3\%$ remain.
- **Amortized cost** If splits $\approx$ *halve* intervals each iteration,
$$\sum_{i=0}^{\log_2 T} \Big(\frac{T}{2^i}\Big)^2 F = \mathcal{O}(T^2F), \qquad \sum_{i=0}^{\log_2 T} \Big(\frac{T}{2^i}\Big) F^3 = \mathcal{O}(TF^3).$$
But because only a decreasing fraction R_k of splits is recomputed at iteration k (30% at $k = 10$, 3% at $k = 30$), the effective runtime behaves like $\mathcal{O}\big(T \log TF\big)$.
- **Revision & Consolidation.** Every 20 iterations, each existing boundary c_j considers its two nearest inflections $i_{j,L}$ and $i_{j,R}$. Let m_j be the number of interior inflections in $(\, i_{j,L}\,,\, i_{j,R}\,)$. Testing all interior splits costs $\mathcal{O}\big(m_j^2 F + m_j F^3\big)$. Summing over $J \ll T$ boundaries and amortizing over 20 iterations adds at most
$$\mathcal{O}\Big(\frac{1}{20}\sum_{j=1}^{J}(m_j^2 F + m_j F^3)\Big),$$
which remains $\ll \mathcal{O}(T \log TF)$ since J and each m_j are small relative to T.

D. Inflection-Point Pruning

BETWEEN restricts candidates to inflection points ($\approx 3\%$ of frames in HAR, so only $\approx T/100$ splits are ever evaluated. Even if each of these is fully recomputed, the cost is reduced by $\sim$ 20–30$\times$ relative to scanning all T frames.

E. Comparison to Other CPD Methods

BETWEEN's one-time $\mathcal{O}(F^3)$ baseline setup is modest (see Table I). Its first two passes incur $\mathcal{O}(T^2F + TF^3)$, but cached scores, KL screening (active splits drop to $\approx 30\%$ by iter 10 and $\approx 3\%$ by iter 30), and inflection-point pruning ($\approx T/100$ candidates) reduce the effective runtime to near $\mathcal{O}(T \log T\, F)$. The infrequent (every 20 iterations) consolidation step adds an amortized cost $\ll \mathcal{O}(T \log T\, F)$. Consequently, BETWEEN matches fast parametric detectors in speed while handling unstructured, high-variance behavioral data gracefully.

VI. Evaluation

BETWEEN was evaluated on Human Action Recognition (HAR) [18], and Bee Waggle (2012), two spatiotemporal datasets consisting of positional and rotational time series.

A. Human Action Recognition (HAR)

Dataset. 30 subjects perform six scripted actions—walk, walk-upstairs, walk-downstairs, sit, stand, lie—while wearing a waist-mounted smartphone.
Sensors & Signals. 3-axis accelerometer + 3-axis gyroscope at 50 Hz (six-channel stream).
Characteristic. Sequences of $\approx 150 - 180$ samples of 2.56s (128 frames) with 50% overlap.
Labels / evaluation. Changepoints denoted by different activity labels during overlap. We run CPD on the uncut sequence and score against TiVaCPD's ± 5 half-window tolerance.

TABLE I
Asymptotic cost per sequence of length T (feature dim. F)

Method	Complexity	Main driver
BOCPD (conjug.)	$\mathcal{O}(TF^2)$	run-length grid
PELT / FPOP	$\mathcal{O}(TF^2)$	pruning, additive cost
Kernel CPD	$\mathcal{O}(T^2)$	kernel matrix
BETWEEN (worst)	$\mathcal{O}(T^2F + TF^3)$	first two passes
BETWEEN (typical)	$\mathcal{O}(T \log T\, F)$	cache + KL + prune

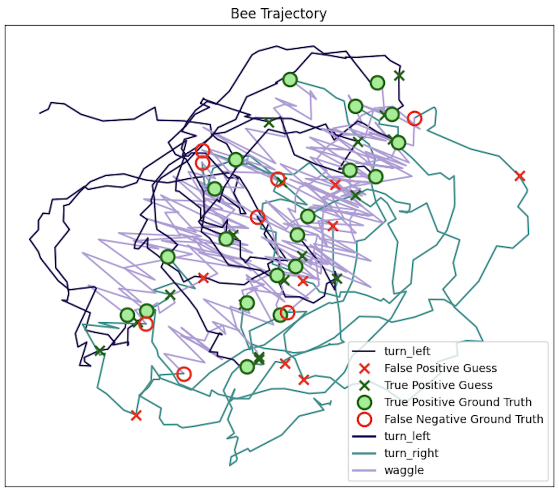

Fig. 2. Segmentation performed on Dance 5 of the Bee Waggle Dataset.

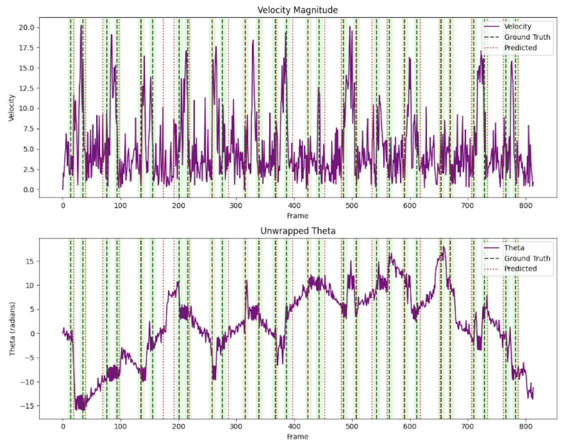

Fig. 3. Segmentation performed on Dance 5 of the Bee Waggle Dataset.

B. Bee Waggle Dance Dataset

Dataset. High-resolution hive videos; automatic tracking yields continuous (x, y, θ) trajectories for each bee.

Sampling & Signals. Six ∼30 Hz sequences (600–1200 triplets) capturing waggle runs and return loops; path unpredictability makes CPD non-trivial.

Labels / evaluation. Original authors annotate waggle-run phases (turning right, turning left, waggle). BETWEEN segments the raw trajectories without using these labels and is scored against the original annotation set (the TCPD extension is ignored, as it covers only one clip with two change points). See Fig. 2 & 3 for examples.

Commentary: The TCPD release of the Bee Waggle data covers only one sequence with two human-labeled changepoints, reflecting subjective shifts in perceived dance frequency rather than objective behavior. While some CPD models use TCPD labels, we benchmark against the original dataset.

TABLE II
PUBLISHED CHANGE–POINT–DETECTION RESULTS ON THE **BEE DANCE** DATASET. BLANK CELLS ARE METRICS NOT REPORTED IN THE SOURCE.

Method	F_1 **(tol.** M=5) **Trained / Default**	**AUC** η=8
TiVaCPD	0.45	– [12]
TiVaCPD-CovScore	0.34	– [12]
TiVaCPD-DistScore	0.32	– [12]
KL-CPD	0.13	– [12]
Roerich	0.40	– [12]
GraphTime	0.22	– [12]
TIRE	0.20	– [12]
BinSeg	0.597 / 0.097	– [19]
Microsoft SSA	0.583 / 0.279	– [19]
BOCPDMS	0.167 / 0.092	– [19]
mSSA (original)	0.404 / 0.124	– [19]
mSSA-MW	0.659 / 0.500	– [19]
HM-RNN	–	0.362 [15]
RCN	–	0.054 [15]
CNN	–	0.192 [15]
GGM	–	0.041 [15]
PRN-S	–	0.119 [15]
DWN	–	0.131 [15]
PRN	–	0.400 [15]
BETWEEN (ours)	0.60	.436

TABLE III
PUBLISHED CPD RESULTS ON THE HAR DATASET ([18], TOLERANCE M=5) UNLESS NOTED.

Method	Precision	Recall	F_1
TiVaCPD	0.72	0.48	0.58 [12]
TiVaCPD-CovScore	0.62	0.56	0.57 [12]
TiVaCPD-DistScore	0.50	0.35	0.40 [12]
KL-CPD	0.66	0.20	0.30 [12]
Roerich	0.69	0.11	0.18 [12]
GraphTime	0.04	0.96	0.08 [12]
TIRE	0.52	0.14	0.22 [12]
BETWEEN (ours) M=5)	0.56	0.72	0.63
BETWEEN (ours) M=1)	0.43	0.54	0.46

BETWEEN achieves the second highest absolute F_1 score on the Bee Dance dataset with $M = 5$ (see Table II), the highest AUC of any model at $\eta = 8$, and the highest F_1 score of any model on the HAR dataset with $M = 5$[3] (See Table III). Despite this, we have reason to believe that these results are stronger than they appear.

In [19], the authors propose an extension SSA models. Their use of singular value decomposition is a very clever way to avoid explicit parametric modeling and make the algorithm more adaptable and flexible; however, a more practical com-

[3]**Thresholded** F_1**:** Let $B = \{b_i\}_{i=1}^{N}$ and $\hat{B} = \{\hat{b}_j\}_{j=1}^{\hat{N}}$. For tolerance M,

$$\mathrm{TP} = \left|\{\hat{b} \in \hat{B} : \exists\, b \in B : |\hat{b} - b| \le M\}\right|, \quad \mathrm{FP} = |\hat{B}| - \mathrm{TP}, \quad \mathrm{FN} = |B| - \mathrm{TP},$$

$$F_1 = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}.$$

AUC: Let $T = \{0, 1, 2, 4, 8\}$ and $F_1(t)$ be the corresponding scores. Then

$$\mathrm{AUC} = \frac{1}{8-0} \sum_{i=1}^{|T|-1} (T_{i+1} - T_i)\, \frac{F_1(T_i) + F_1(T_{i+1})}{2}.$$

parison would be of BETWEEN and the default mSSA models – to achieve the results on the left side of the category, they've tuned the parameters, such as the size of the moving window and the number of principle components k, specifically for the ground truth of this dataset.

The necessity of using a tolerance of $M{=}5$ for BETWEEN's F_1 score for comparison purposes also somewhat understates the frequency of a positive prediction in close proximity to a changepoint. The usage of inflections to find appropriate cutting points for the time series means that BETWEEN might miss by a more few frames than without. As a result, the AUC at $\eta{=}8$ is .436, higher than any of the deep learning models surveyed, *even though those models trained directly on the bee dance data*. At that point, BETWEEN has F_1 of 0.731.

BETWEEN attains state-of-the-art results on HAR. The tolerance $M{=}5$ reflects a comparison quirk: prior work ran binary changepoint tests in 64-frame windows. Because BETWEEN predicts at the frame level, competing methods—unable to issue multiple positives within a true-positive window—show inflated precision. Nonetheless, BETWEEN has the best F1, and even its $M{=}5$ score exceeds all but one model class.

VII. Conclusions and Future Work

In this work, we introduce BETWEEN, a novel unsupervised change point detection framework for multivariate spatiotemporal sequences. Our approach emphasizes generality, computational efficiency, and high segmentation quality, addressing key challenges in unsupervised time-series analysis. BETWEEN requires no labeled supervision to identify segment boundaries, yet we have demonstrated broad applicability across unstructured spatiotemporal domains.

Through an inflection-based pruning strategy and caching, the algorithm significantly reduces the search space for potential change points, achieving sub-quadratic runtime complexity. By leveraging a hybrid gain function which combines dynamic time warping distances, signal energy change ratio, and Kullback-Leibler divergence for distributional shifts, the framework attains superior accuracy, outperforming state-of-the-art CPD algorithms and semantic segmentation methods on real world spatiotemporal datasets without training. These results indicate that the multi-criterion gain function can robustly capture a range of change signatures.

We plan to extend this work to more rigorously balance the DTW-Energy product and KL penalty, potentially with a GradNorm-like scheme. We also aim to integrate deep learning—particularly variational autoencoders—into the segmentation pipeline to capture higher-level temporal features and improve cross-dimensional weighting. This should enhance detection of abstract or context-dependent changepoints and reduce false positives. Even without algorithmic changes, we will examine how the gain function's two components interact to guide principled hyperparameter tuning.

Ultimately, we envision BETWEEN as an unsupervised "tokenization" for spatiotemporal data, producing coherent segments serving as input to downstream sequence models. This abstraction could enable more scalable and interpretable modeling for tasks like activity recognition and prediction. BETWEEN's success derives from demonstrated gains to downstream learning. Together, these directions position BETWEEN as a foundation for semantically-informed, scalable sequence modeling across real-world spatiotemporal data.

References

[1] P. Yang, G. Dumont, and J. Ansermino, "Adaptive change detection in heart rate trend monitoring in anesthetized children," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 11, pp. 2211–2219, 2006.

[2] X. He, M.-O. Pun, C.-C. J. Kuo, and Y. Zhao, "A change-point detection approach to power quality monitoring in smart grids," Mitsubishi Electric Research Laboratories, Cambridge, Massachusetts, Tech. Rep. TR2010-054, July 2010. [Online]. Available: https://merl.com/publications/docs/TR2010-054.pdf

[3] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Proceedings of the Seventeenth Conference on Innovative Applications of Artificial Intelligence (IAAI-05)*. AAAI Press, 2005, pp. 1541–1546. [Online]. Available: https://cdn.aaai.org/AAAI/2005/IAAI05-013.pdf

[4] A. J. Scott and M. Knott, "A cluster analysis method for grouping means in the analysis of variance," *Biometrics*, pp. 507–512, 1974.

[5] Z. Harchaoui and C. L'evy-Leduc, "Multiple change-point estimation with a total variation penalty," *Journal of the American Statistical Association*, vol. 105, no. 492, pp. 1480–1493, 2010.

[6] F. Desobry, M. Davy, and C. Doncarli, "An online kernel change detection algorithm," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2961–2974, 2005.

[7] R. P. Adams and D. J. MacKay, "Bayesian online changepoint detection," *arXiv preprint arXiv:0710.3742*, 2007.

[8] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1590–1598, 2012.

[9] D. Workinn, "Improving change point detection using self-supervised vaes: A study on distance metrics and hyperparameters in time series analysis," 2023.

[10] T. Shiraishi, D. Miwa, V. N. Le Duy, and I. Takeuchi, "Selective inference for change point detection by recurrent neural network," *Neural Computation*, vol. 37, no. 1, pp. 160–192, 12 2024. [Online]. Available: https://doi.org/10.1162/neco_a_01724

[11] Q. Martinez, C. Chen, J. Xia, and H. Bahai, "Sequence-to-sequence change-point detection in single-particle trajectories via recurrent neural network for measuring self-diffusion," *Transport in Porous Media*, vol. 147, pp. 679–701, 2023. [Online]. Available: https://bura.brunel.ac.uk/bitstream/2438/26261/3/FullText.pdf

[12] K. Garg, J. Yu, T. Behrouzi, S. Tonekaboni, and A. Goldenberg, "Dynamic interpretable change point detection," *arXiv preprint arXiv:2211.03991*, 2022.

[13] Z. Atashgahi, D. C. Mocanu, R. Veldhuis, and M. Pechenizkiy, "Memory-free online change-point detection: A novel neural network approach," *arXiv preprint arXiv:2207.03932*, 2022.

[14] J. Knoblauch and T. Damoulas, "Spatio-temporal bayesian on-line changepoint detection with model selection," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2718–2727.

[15] C. Truong, L. Oudre, and N. Vayatis, "A review of change point detection methods," *CoRR*, vol. abs/1801.00718, 2018. [Online]. Available: http://arxiv.org/abs/1801.00718

[16] R. Maidstone, T. Hocking, G. Rigaill, and P. Fearnhead, "On optimal multiple changepoint algorithms for large data," 2014. [Online]. Available: https://arxiv.org/abs/1409.1842

[17] A. Serre, D. Chételat, and A. Lodi, "Change point detection by cross-entropy maximization," *CoRR*, vol. abs/2009.01358, 2020. [Online]. Available: https://arxiv.org/abs/2009.01358

[18] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz *et al.*, "A public domain dataset for human activity recognition using smartphones." in *Esann*, vol. 3, no. 1, 2013, pp. 3–4.

[19] A. Alanqary, A. Alomar, and D. Shah, "Change point detection via multivariate singular spectrum analysis," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 23 218–23 230. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/c348616cd8a86ee661c7c98800678fad-Paper.pdf