Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 23(11): 3241–3272.
DOI: 10.3934/mbe.2026112
Received: 12 July 2026
Revised: 02 September 2026
Accepted: 21 September 2026
Published: 03 November 2026

Research article

Change-in-velocity detection in multidimensional data

Linh Do[1,*], Dat Do[2], Keisha J. Cook[3], Yitao Shen[4] and Scott A. McKinley[5]

[1] Department of Biostatistics and Bioinformatics, Duke University School of Medicine, 2424 Erwin Road, Durham, NC 27705, USA
[2] Department of Statistics, University of Chicago, 5747 S. Ellis Avenue, Chicago, IL 60637, USA
[3] School of Mathematical and Statistical Sciences, Clemson University, O-110 Martin Hall, 220 Parkway Drive, Clemson, SC 29634, USA
[4] Biochemistry and Molecular Biology and the Huck Institutes of the Life Sciences, The Pennsylvania State University, University Park, PA 16802, USA
[5] Department of Mathematics, Tulane University, 6823 St. Charles Avenue, New Orleans, LA 70118, USA

* **Correspondence:** Email: thuylinh.do@duke.edu; Tel: +1-504-493-4897.

Supplementary

This document provides supplementary figures, additional simulation details, computational performance benchmarks, and a detailed balance verification supporting the main manuscript.

S.1. Additional figures

This section provides additional CPLASS output for simulated and experimental trajectories. For visualization, segments with inferred speed below 0.1μm/s are shaded pink and segments with inferred speed above 0.1μm/s are shaded green. This threshold is used only for visualization and not by the CPLASS segmentation algorithm.

S.1.1. Examples of CPLASS performance for experimental observations

In this section, we provided some real paths and output from CPLASS, Figure S1 shows four paths coming from different datasets, the first two panel (A) and (B) are paths from the lysosomal transport dataset in the periphery region of the cell, the last two panel (C) and (D) are paths from the quantum

dot transport datasets under two circumstances - one with a single Kinesin-1 motor (panel (C)) and one when the two motors Kin1 and DDB present at the same time. Figure S2 is another path in the group Kin1DDB. We can observe bidirectional, multi-state transport, possibly due to tug-of-war dynamics between the motors. For the Kin1 path, the motor-cargo complex appears to be always in active transport; the detected changes in this case are when it changes the speed or direction during the movement.

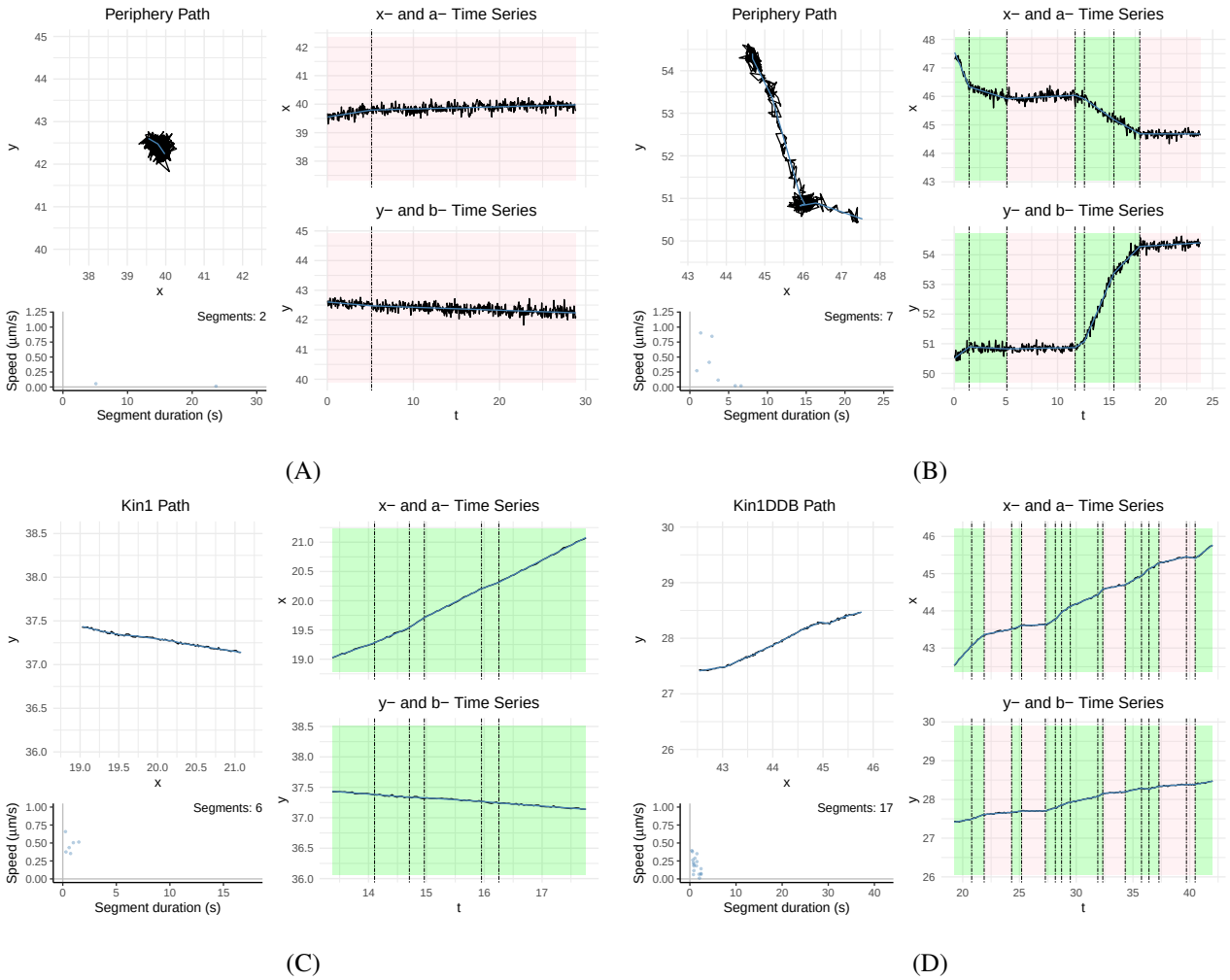


Figure S1. *CPLASS performance for intracellular transport trajectories* **Panel (A) and (B)** - *in vivo* lysosomal transport - examples of a non-active path (Panel A) and an active path (Panel B). Panel (C) and (D) - *in vitro* quantum dot transport - are examples of an active path with a kin1 motor (Panel C) and a path moving and pausing while the two motors kin1 and DDB are present at the same time.

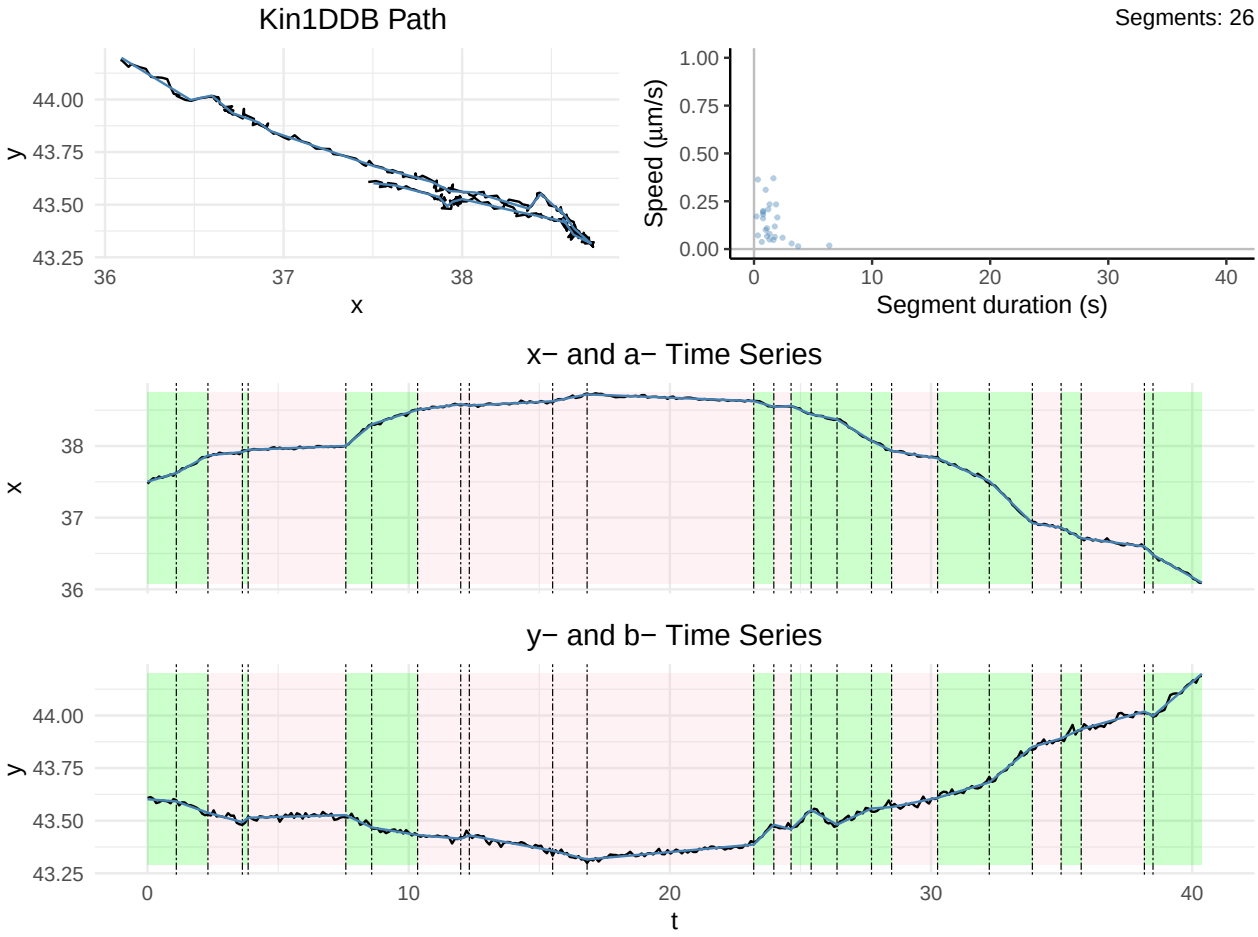
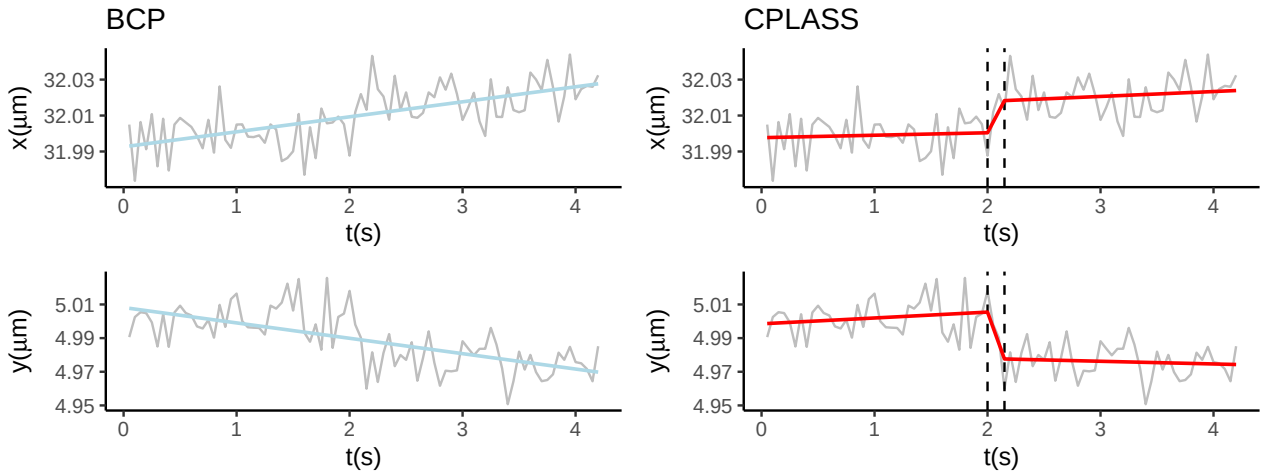


Figure S2. CPLASS analysis of a kinesin-1/DDB trajectory exhibiting bidirectional transport. The green and pink shaded regions indicate segments that are estimated to have a speed of over or under 100 nm/s respectively.

S.1.2. BCP and CPLASS on detecting short segments

In this section, we provide an example where BCP struggles with detecting short segments as discussed in the main manuscript (Section 3.2) and Figure 8 (the corresponding figure in the main manuscript). With this issue, the motile segment was missed, and the whole path was labeled stationary. Meanwhile, CPLASS detected the short fast segment and labeled the middle segment active (see Figure S3).



Segment	1
Duration	4.2 s
Speed	0.0122 µm/s
State	Stationary

Segment	1	2	3
Duration	1.95 s	0.15 s	2.05 s
Speed	0.0037 µm/s	0.2207 µm/s	0.0032 µm/s
State	Stationary	Motile	Stationary

Figure S3. *Comparison of BCP and CPLASS for detection of a short active-transport segment.* The trajectory is one of the simulated paths analyzed in the main manuscript, observed at 20 Hz with $n = 84$ and $\sigma = 0.01$. Two true changepoints define three segments with durations $(2, 0.2, 2)s$ and speeds $(0, 0.17, 0)\mu$m/s. BCP fails to detect either changepoint, whereas CPLASS identifies both changepoints and estimates the duration and speed of the middle segment close to their true values.

S.1.3. CPLASS settings for the experimental analyses

All experimental trajectories were analyzed using the same stochastic-search settings, except that the physical reference speed was adapted to the application. Table S1 reports the complete settings. Here, λ_r is the rate parameter used by the independent proposal. For every trajectory, the selected multistart result was the valid fit with the largest value of the penalized criterion; ties were resolved by the fixed ordering of the predefined deterministic seeds.

Table S1. CPLASS settings used for the three experimental applications. The proposal-type probabilities are $(w_1, w_2, w_3, w_4) = (1/4, 1/8, 1/8, 1/2)$ for Types 1–4, respectively, for every dataset.

Dataset	λ_r	$T_{\max}$	Burn-in	Starts	γ	η	s_{cap}	$K_{\max}$
Lysosome	1/30	5000	500	16	1.01	1	$1\,\mu$m/s	30
Quantum dot	1/30	5000	500	16	1.01	1	$1\,\mu$m/s	30
EB1-GFP	1/30	5000	500	16	1.01	1	$0.25\,\mu$m/s	30

S.2. Simulation parameters used in the main manuscript

Table S2 summarizes the parameter values used to generate the 250 synthetic trajectories analyzed in the main manuscript. The simulation model was introduced by Cook et al. [1]. Here, we provide the

generative procedure used by the simulation code.

The latent trajectory is constructed as a sequence of stationary and motile segments. Let $J_m \in \{S, M\}$ denote the state of segment m, where S and M represent stationary and motile states, respectively. The state is updated only when a new segment begins, rather than at every observation time. The embedded segment-level Markov chain has transition matrix

$$P = \begin{pmatrix} p_{SS} & p_{SM} \\ p_{MS} & p_{MM} \end{pmatrix},$$

where p_{ab} is the probability that a segment in state a is followed by a segment in state b. For the Base parameter setting,

$$p_{SS} = 0, \qquad p_{SM} = 1, \qquad p_{MS} = 0.5, \qquad p_{MM} = 0.5.$$

Consequently, a stationary segment is always followed by a motile segment, whereas a motile segment is followed by either a stationary or another motile segment with equal probability.

For a stationary segment, the speed is zero and the segment duration is generated as

$$T_S \sim \text{Exponential}\left(\frac{1}{\bar{T}_S}\right),$$

where $\bar{T}_S = 5$ s is the mean stationary duration.

For a motile segment, the speed, expressed in nm/s, is generated according to

$$V \sim \text{Gamma}(\alpha,\ \text{rate} = \beta).$$

Conditional on the sampled speed $V = v$, the motile-segment duration is generated as

$$T_M \mid V = v \sim \text{Exponential}\left(\frac{v}{\bar{D}}\right),$$

where $\bar{D}$ is the mean distance traveled during a motile segment. Therefore,

$$\mathbb{E}(T_M \mid V = v) = \frac{\bar{D}}{v},$$

and the expected distance traveled during the segment, conditional on $V = v$, is $\bar{D}$.

When a motile segment begins, its direction reverses relative to the preceding direction with probability P_{reverse}, continues in the same direction with probability P_{continue}, and is otherwise drawn uniformly from $[0, 2\pi)$. Thus, under the Base setting, the probabilities of reversing, continuing, and selecting a new random direction are 0.3, 0.3, and 0.4, respectively.

The resulting segments define a continuous piecewise-linear latent anchor trajectory. After the latent trajectory has been generated, it is evaluated on a regular observation grid with spacing Δ. Independent Gaussian localization errors are then added to both spatial coordinates:

$$X_i = A_i + \sigma_{\text{cargo}}\epsilon_{xi}, \qquad Y_i = B_i + \sigma_{\text{cargo}}\epsilon_{yi},$$

where A_i and B_i are the latent anchor coordinates and

$$\epsilon_{xi}, \epsilon_{yi} \overset{\text{iid}}{\sim} N(0, 1).$$

The original simulation code uses $n_{\text{int}} = 200$ sampling intervals and includes both endpoints of the observation grid. Thus, at 25 Hz,

$$\Delta = 0.04 \text{ s}, \qquad t_f = n_{\text{int}}\Delta = 8 \text{ s},$$

and each saved trajectory contains $n_{\text{int}} + 1 = 201$ observed positions.

Table S2. Base parameters used to generate the simulated lysosome trajectories analyzed in the main manuscript. State-transition probabilities apply between latent segments, not between consecutive observation times. The latent trajectory is sampled at 25 Hz after the segments have been generated.

Parameter	Base	Description
N	250	Number of simulated trajectories
n_{int}	200	Number of sampling intervals
p_{SS}	0	Probability $S \to S$ between segments
p_{SM}	1	Probability $S \to M$ between segments
p_{MS}	0.5	Probability $M \to S$ between segments
p_{MM}	0.5	Probability $M \to M$ between segments
α	8	Shape parameter of the motile-speed distribution
β	0.02	Rate parameter of the motile-speed distribution
$\bar{D}$	300 nm	Mean distance traveled during a motile segment
$\bar{T}_S$	5 s	Mean stationary-segment duration
P_{reverse}	0.3	Probability of reversing direction
P_{continue}	0.3	Probability of continuing in the same direction
P_{random}	0.4	Probability of selecting a new uniform direction
σ_{cargo}	0.1 μm	Standard deviation of localization noise
Δ	0.04 s	Observation-grid spacing at 25 Hz
t_f	8 s	Final simulation time

The simulation algorithm is summarized as follows:

1. Initialize the segment state from the stationary distribution of the embedded segment-level Markov chain and generate an initial segment duration, speed, and direction.
2. Run the segment process for a 20 s burn-in period.
3. At each segment boundary, draw the next state using the transition matrix P.
4. Conditional on the new state, generate the segment duration and, for a motile segment, its speed and direction.
5. Repeat the segment-generation step until the latent trajectory covers the required final time t_f.
6. Construct the continuous piecewise-linear anchor trajectory and evaluate it on the regular observation grid.
7. Add independent Gaussian localization errors with standard deviation σ_{cargo} to both observed spatial coordinates.

This procedure produces synthetic trajectories together with their true segment states, changepoint times, durations, speeds, directions, and latent anchor positions. These stored quantities provide the ground truth used to evaluate the segmentation and motion-inference methods.

S.3. Checking the detailed balance condition in the MH algorithm

Recall that our main algorithm generates a Markov chain with stationary distribution $\pi(r) \propto \exp\{\Phi(r)\}$ as a stochastic search method for the changepoint configuration $r \in \mathcal{R}_n$. This is done via the Metropolis–Hastings (MH) algorithm with four proposal types described in the main manuscript. Our goal in this section is to verify that the resulting transition kernel satisfies detailed balance, a standard sufficient condition ensuring that π is stationary for the MH chain [2, Chapter 7].

Condition 1 (Detailed balance with respect to π)**.** *Let $P(r' \mid r)$ be a Markov transition kernel on $\mathcal{R}_n$; that is,*

$$P(r' \mid r) \geq 0 \quad \textit{and} \quad \sum_{r' \in \mathcal{R}_n} P(r' \mid r) = 1 \qquad \textit{for every } r \in \mathcal{R}_n.$$

The kernel P is said to satisfy the detailed balance condition *with respect to the target distribution π if*

$$\pi(r)P(r' \mid r) = \pi(r')P(r \mid r') \qquad \textit{for every } r, r' \in \mathcal{R}_n. \tag{S.1}$$

At each iteration, our algorithm proposes one of the four new types with probability w_1, w_2, w_3 and w_4. We first show that it suffices to prove Condition 1 for each type (Lemma 1). We then provide the proof for each proposal type separately (Proposition 1). For notational convenience, let q_ℓ denote the proposal probability conditional on selecting proposal Type ℓ and let P_ℓ denote the complete MH transition kernel after the acceptance/rejection step, for $\ell = 1, \ldots, 4$.

Lemma 1 (Fixed mixtures preserve detailed balance)**.** *Let $P_1, \ldots, P_L$ be Markov transition kernels on $\mathcal{R}_n$, and let $w_1, \ldots, w_L$ be constants satisfying $w_\ell \geq 0$ and $\sum_{\ell=1}^{L} w_\ell = 1$. If every P_ℓ satisfies detailed balance with respect to π, then the mixture*

$$P(r' \mid r) = \sum_{\ell=1}^{L} w_\ell P_\ell(r' \mid r)$$

also satisfies detailed balance with respect to π.

Proof. The mixture P is a Markov kernel because it is nonnegative and, for every $r \in \mathcal{R}_n$,

$$\sum_{r' \in \mathcal{R}_n} P(r' \mid r) = \sum_{\ell=1}^{L} w_\ell \sum_{r' \in \mathcal{R}_n} P_\ell(r' \mid r) = \sum_{\ell=1}^{L} w_\ell = 1.$$

Moreover, for every $r, r' \in \mathcal{R}_n$,

$$\begin{aligned} \pi(r)P(r' \mid r) &= \sum_{\ell=1}^{L} w_\ell \pi(r) P_\ell(r' \mid r) \\ &= \sum_{\ell=1}^{L} w_\ell \pi(r') P_\ell(r \mid r') \\ &= \pi(r')P(r \mid r'). \end{aligned}$$

Thus P satisfies Condition 1. □

Proposition 1 (Reversibility of the type-specific MH kernels)**.** *For each proposal type $\ell \in \{1, 2, 3, 4\}$, the complete MH kernel P_ℓ satisfies detailed balance with respect to π.*

The proof is organized into four subsections, corresponding to the four proposal types. For a proposed state $r^{\text{prop}} \neq r^{\text{cur}}$, the MH acceptance probability conditional on selecting proposal Type ℓ is

$$\alpha_\ell(r^{\text{prop}}|r^{\text{cur}}) = \min\left(1, \frac{\pi(r^{\text{prop}})q_\ell(r^{\text{cur}}|r^{\text{prop}})}{\pi(r^{\text{cur}})q_\ell(r^{\text{prop}}|r^{\text{cur}})}\right).$$

Thus, for $r^{\text{prop}} \neq r^{\text{cur}}$,

$$P_\ell(r^{\text{prop}}|r^{\text{cur}}) = q_\ell(r^{\text{prop}}|r^{\text{cur}})\alpha_\ell(r^{\text{prop}}|r^{\text{cur}}).$$

For each proposal type ℓ, it is enough to verify the two following facts.

1. For every $r \in \mathcal{R}_n$, $q_\ell(\cdot \mid r)$ is a probability distribution on $\mathcal{R}_n$. Any unavailable or inadmissible outcome is assigned to the self-transition $q_\ell(r \mid r)$.
2. For every pair of distinct admissible states r, r', the stated values of $q_\ell(r' \mid r)$ and $q_\ell(r \mid r')$ are the actual forward and reverse proposal probabilities. Moreover,

$$q_\ell(r' \mid r) > 0 \quad \Longleftrightarrow \quad q_\ell(r \mid r') > 0.$$

Once these facts are verified, detailed balance follows from the standard MH identity

$$\pi(r)P_\ell(r' \mid r) = \min\{\pi(r)q_\ell(r' \mid r),\, \pi(r')q_\ell(r \mid r')\}, \qquad r' \neq r, \tag{S.2}$$

as the right-hand side is symmetric in r and r'. When $r' = r$, the detailed-balance identity is tautological. See, for example, [2, Chapter 7].

For the CPLASS proposals, these requirements have the following concrete content. Type 1 requires the probability of each Bernoulli vector and the self-transition mass from inadmissible draws. Type 2 requires the exact numbers of occupied and unoccupied candidate locations, together with the birth/death boundary cases. Type 3 requires the exact two-stage insertion rule, the exact set of removable ordered-neighbor pairs, normalization of the complete proposal kernel, and all unavailable-direction self-transitions. Type 4 requires the exact numbers of occupied and unoccupied locations and the shift boundary cases. The Type 3 verification is the most involved and is given explicitly in Section S.3.3 below.

S.3.1. Type 1: q_{new} (Independent changepoint vector proposal)

Proof of Proposition 1; Type 1. Type 1 draws each component independently from a Bernoulli(p_λ) distribution, where $p_\lambda = 1 - \exp(-\lambda\Delta)$. Because the Bernoulli law is a probability distribution on $\{0, 1\}^N$, mapping every inadmissible draw to the current state produces a probability distribution on $\mathcal{R}_n$. This verifies Fact 1.

For two distinct admissible states, the actual forward and reverse proposal probabilities are

$$q_{\text{new}}(r^{\text{prop}}|r^{\text{cur}}) = p_\lambda^{|r^{\text{prop}}|}(1 - p_\lambda)^{N-|r^{\text{prop}}|},$$
$$q_{\text{new}}(r^{\text{cur}}|r^{\text{prop}}) = p_\lambda^{|r^{\text{cur}}|}(1 - p_\lambda)^{N-|r^{\text{cur}}|}.$$

These are the probabilities generated by the proposal rule and, because $0 < p_\lambda < 1$, both are positive. This verifies Fact 2.

Therefore,

$$\alpha_{\text{new}}(r^{\text{prop}}|r^{\text{cur}}) = \min\left(1,\ \frac{\pi(r^{\text{prop}})}{\pi(r^{\text{cur}})}\left(\frac{1-e^{-\lambda\Delta}}{e^{-\lambda\Delta}}\right)^{|r^{\text{cur}}|-|r^{\text{prop}}|}\right).$$

□

S.3.2. Type 2: q_{bd} (Birth/death proposal)

Proof of Proposition 1; Type 2. Birth and death are each selected with probability 1/2, after which the location is sampled uniformly from the appropriate occupied or unoccupied set. An unavailable or inadmissible outcome is assigned to the current state, so the total proposal probability remains one. This verifies Fact 1.

For two distinct admissible states, the actual proposal probability is

$$q_{\text{bd}}(r^{\text{prop}} \mid r^{\text{cur}}) = \begin{cases} \dfrac{1}{2|r^{\text{cur}}|}, & \text{if one changepoint is deleted,} \\ \dfrac{1}{2(N-|r^{\text{cur}}|)}, & \text{if one changepoint is added,} \\ 0, & \text{otherwise.} \end{cases}$$

Adding a changepoint is reversed by deleting that same changepoint, and conversely. Hence the corresponding forward and reverse probabilities are both positive and are obtained from the displayed formula by interchanging r^{cur} and r^{prop}. This verifies Fact 2. The resulting acceptance probability is

$$\alpha_{\text{bd}}(r^{\text{prop}} \mid r^{\text{cur}}) = \min\left\{1, \frac{\pi(r^{\text{prop}})}{\pi(r^{\text{cur}})}\left\{\begin{array}{ll} \dfrac{|r^{\text{cur}}|}{N-|r^{\text{prop}}|}, & \text{for deletion,} \\ \dfrac{N-|r^{\text{cur}}|}{|r^{\text{prop}}|}, & \text{for birth.} \end{array}\right\}\right.$$

□

S.3.3. Type 3: q_{bd_2} (Segment insertion/deletion proposal)

Proof of Proposition 1; Type 3. The Type 3 proposal adds or removes two changepoints at the same time.

Because this proposal is asymmetric, we explicitly derive the probability of proposing each particular state.

Let $M_1 < \cdots < M_{|r|}$ denote the ordered changepoint locations of state r, with $M_0 = 1$ and $M_{|r|+1} = n$ denoting the trajectory boundaries.

For each segment j, define

$$B_j(r) = \left\{\{s, s'\} : M_{j-1} < s < s' < M_j,\ r^{+(s,s')} \in \mathcal{R}_n\right\},$$

and define the collection of segments that can accommodate an admissible two-changepoint insertion by

$$E_B(r) = \{j : B_j(r) \neq \varnothing\}.$$

Conditional on selecting Type 3, the algorithm first chooses the paired insertion direction or the paired deletion direction, each with probability $1/2$. For insertion, it samples a segment uniformly from $E_B(r)$ and then a pair uniformly from that segment's set $B_j(r)$.

For deletion, define

$$D(r) = \left\{\{M_j, M_{j+1}\} : j = 1, \ldots, |r| - 1,\ r^{-(M_j, M_{j+1})} \in \mathcal{R}_n\right\}.$$

Here, "consecutive" means adjacent changepoints in the ordered changepoint list $(M_1, \ldots, M_{|r|})$, not changepoints at consecutive observation indices.

For deletion, it samples one pair uniformly from $D(r)$.

If the selected direction is unavailable, its probability $1/2$ is assigned to the self-transition rather than being redistributed. Thus

$$q_{\mathrm{bd}_2}(r \mid r) = \frac{1}{2}\mathbf{1}\{E_B(r) = \varnothing\} + \frac{1}{2}\mathbf{1}\{D(r) = \varnothing\}. \tag{S.3}$$

When available, the uniformly sampled insertion and deletion directions each have total probability $1/2$. Consequently,

$$\begin{aligned}\sum_{r^{\mathrm{prop}} \in \mathcal{R}_n} q_{\mathrm{bd}_2}(r^{\mathrm{prop}}|r) &= \frac{1}{2}\mathbf{1}\{E_B(r) \neq \varnothing\} + \frac{1}{2}\mathbf{1}\{D(r) \neq \varnothing\} + q_{\mathrm{bd}_2}(r \mid r)\\ &= 1.\end{aligned}$$

This verifies Fact 1.

For an insertion of $\{s, s'\} \in B_j(r^{\mathrm{cur}})$, the inserted changepoints are adjacent in the ordered list of r^{prop}, and

$$\begin{aligned}q_{\mathrm{bd}_2}(r^{\mathrm{prop}} \mid r^{\mathrm{cur}}) &= \frac{1}{2|E_B(r^{\mathrm{cur}})||B_j(r^{\mathrm{cur}})|},\\ q_{\mathrm{bd}_2}(r^{\mathrm{cur}} \mid r^{\mathrm{prop}}) &= \frac{1}{2|D(r^{\mathrm{prop}})|}.\end{aligned}$$

The first quantity is the probability of proposing this particular state, not the aggregate probability of inserting some pair into segment j. For a deletion from $D(r^{\mathrm{cur}})$, let j^* denote the merged segment of r^{prop} containing the deleted pair. Then

$$\begin{aligned}q_{\mathrm{bd}_2}(r^{\mathrm{prop}} \mid r^{\mathrm{cur}}) &= \frac{1}{2|D(r^{\mathrm{cur}})|},\\ q_{\mathrm{bd}_2}(r^{\mathrm{cur}} \mid r^{\mathrm{prop}}) &= \frac{1}{2|E_B(r^{\mathrm{prop}})||B_{j^*}(r^{\mathrm{prop}})|}.\end{aligned}$$

These are the actual probabilities of the two-stage sampling rules, and each move is reversed by deleting or reinserting the same pair. This verifies Fact 2. The resulting acceptance probability is

$$\alpha_{\mathrm{bd}_2}(r^{\mathrm{prop}} \mid r^{\mathrm{cur}}) = \min\left\{1, \frac{\pi(r^{\mathrm{prop}})}{\pi(r^{\mathrm{cur}})}\begin{cases}\dfrac{|E_B(r^{\mathrm{cur}})||B_j(r^{\mathrm{cur}})|}{|D(r^{\mathrm{prop}})|}, & \text{for insertion,}\\ \dfrac{|D(r^{\mathrm{cur}})|}{|E_B(r^{\mathrm{prop}})||B_{j^*}(r^{\mathrm{prop}})|}, & \text{for deletion.}\end{cases}\right\}.$$

□

S.3.4. Type 4: q_{shift} (Shift proposal)

Proof of Proposition 1; Type 4. Type 4 samples one occupied and one unoccupied location uniformly. An unavailable or inadmissible outcome is assigned to the current state, so the resulting proposal probabilities sum to one. This verifies Fact 1.

For two distinct admissible states connected by a shift,

$$q_{\text{shift}}(r^{\text{prop}}|r^{\text{cur}}) = \frac{1}{|r^{\text{cur}}|(N - |r^{\text{cur}}|)}.$$

The reverse shift exchanges the same two locations. Since $|r^{\text{prop}}| = |r^{\text{cur}}|$,

$$q_{\text{shift}}(r^{\text{cur}}|r^{\text{prop}}) = \frac{1}{|r^{\text{prop}}|(N - |r^{\text{prop}}|)} = q_{\text{shift}}(r^{\text{prop}}|r^{\text{cur}}).$$

Thus the displayed quantities are the actual forward and reverse proposal probabilities and are both positive. This verifies Fact 2. Consequently,

$$\alpha_{\text{shift}}(r^{\text{prop}}|r^{\text{cur}}) = \min\left\{1, \frac{\pi(r^{\text{prop}})}{\pi(r^{\text{cur}})}\right\}.$$

□

Completion of the detailed-balance verification.

Completion of proof of Proposition 1. The four preceding subsections give the correct forward and reverse probabilities for every admissible move and show how unavailable or inadmissible moves are assigned to self-transitions. For each type, a move between two distinct states can be reversed by a move of the same type. Therefore, Equation (S.2) establishes detailed balance for every distinct pair of states. When the two states are the same, the detailed-balance identity is tautological. Hence each P_ℓ, $\ell = 1, \ldots, 4$, is a Markov kernel satisfying Condition 1. □

S.4. Sensitivity to bounded noise and isolated positional outliers

The use of a Gaussian distribution for the fluctuation about the phenomenological anchor is itself a phenomenological choice. In the case of intracellular cargo, the motor-cargo tether has a finite extension. Meanwhile, random departures from piecewise-linear paths result from variations in motor speed, curvature in the microtubules, and observation error in tracking cargo positions. Rather than attempt to model these either separately or in concert, we summarize them with the Gaussian model for the noise.

It is reasonable to ask what happens when this noise model is misspecified. To address this, we conducted a targeted simulation study to evaluate two departures from the Gaussian noise model: (1) noise with compact support and (2) isolated positional outliers. We stress that this analysis is intended to examine two natural forms of model misspecification and not to establish robustness to all possible non-Gaussian distributions. In particular, if an underlying physical model entails heavy-tailed displacement distributions or persistent false location readings, we expect that the statistical likelihood model would need to be adjusted.

In our misspecification study, we generated trajectories of length $n = 53$, sampled at 20 Hz over 2.65 s. Under the null model, the latent trajectory was stationary. Under the alternative, the latent trajectory was stationary until $t = 0.90$ s, moved diagonally for 0.90 s at a speed of $0.50\,\mu$m/s, and then became stationary. The two velocity components during the motile segment were $0.50/\sqrt{2}$ and $-0.50/\sqrt{2}\,\mu$m/s.

We considered three observation-error conditions. In the Gaussian condition, the coordinate errors were independent $N(0, \sigma^2)$ variables with $\sigma = 0.01\,\mu$m. In the bounded condition, the coordinate errors were independent Uniform$(-\sqrt{3}\sigma, \sqrt{3}\sigma)$ which has compact support and variance σ^2. In the outlier condition, Gaussian errors were generated as above and one observation was displaced in a uniformly random two-dimensional direction by a radial distance of 5σ. The outlier was placed away from the trajectory endpoints and, under the alternative, away from the true changepoints.

Fifty trajectories were generated for each combination of truth model and error condition. Each trajectory was fitted with $\gamma = 1.01$, $s_{\text{cap}} = 1\,\mu$m/s, 16 deterministic starts, $K_{\max} = 30$, and the remaining default CPLASS settings. We considered both $\eta = 0$, without the speed penalty, and $\eta = 1$, with the speed penalty. This produced 300 unique trajectories and 600 CPLASS fits.

A true changepoint was considered recovered when an inferred changepoint occurred within two sampling indices (0.10 s) of it. An outlier-associated pair was defined as at least two inferred changepoints within two sampling indices of the outlying observation. Exact changepoint count refers only to selecting the correct number of changepoints; recovery additionally requires proximity to the true changepoint locations.

Under Gaussian and bounded errors, CPLASS produced one false-positive trajectory among the 50 null trajectories in each condition. Under the alternative, both true changepoints were recovered in all 50 trajectories under both error distributions. Exactly two changepoints were selected in 44 of 50 Gaussian trajectories and 47 of 50 bounded-error trajectories. Thus, the compactly supported error distribution did not reduce recovery or increase the observed false-positive frequency relative to the Gaussian working model.

After adding one isolated 5σ outlier, CPLASS produced at least one false changepoint in 3 of 50 null trajectories. Under the alternative, both true changes were still recovered in all 50 trajectories, although exactly two changepoints were selected in 41 of 50 trajectories. An outlier-associated changepoint pair occurred in 3 of 50 null trajectories and 6 of 50 alternative trajectories. These results illustrate that an isolated positional anomaly can occasionally be represented as a short inferred regime. Please see Figures S4 and S5 .

The reported false-positive, true-change recovery, exact-count, extra-count, and outlier-pair frequencies were identical for $\eta = 0$ and $\eta = 1$ and are therefore reported once. The precise inferred changepoint locations were not necessarily identical between the two penalty settings. Thus, the speed penalty should not be interpreted as protection against isolated observation anomalies; its purpose is to discourage persistently high fitted segment speeds.

S.5. Computational performance

To characterize the computational performance of CPLASS, we benchmarked the current R package implementation using simulated trajectories. Each runtime benchmark was repeated five times, and median elapsed times together with interquartile ranges are reported. CPLASS uses Metropolis–Hastings

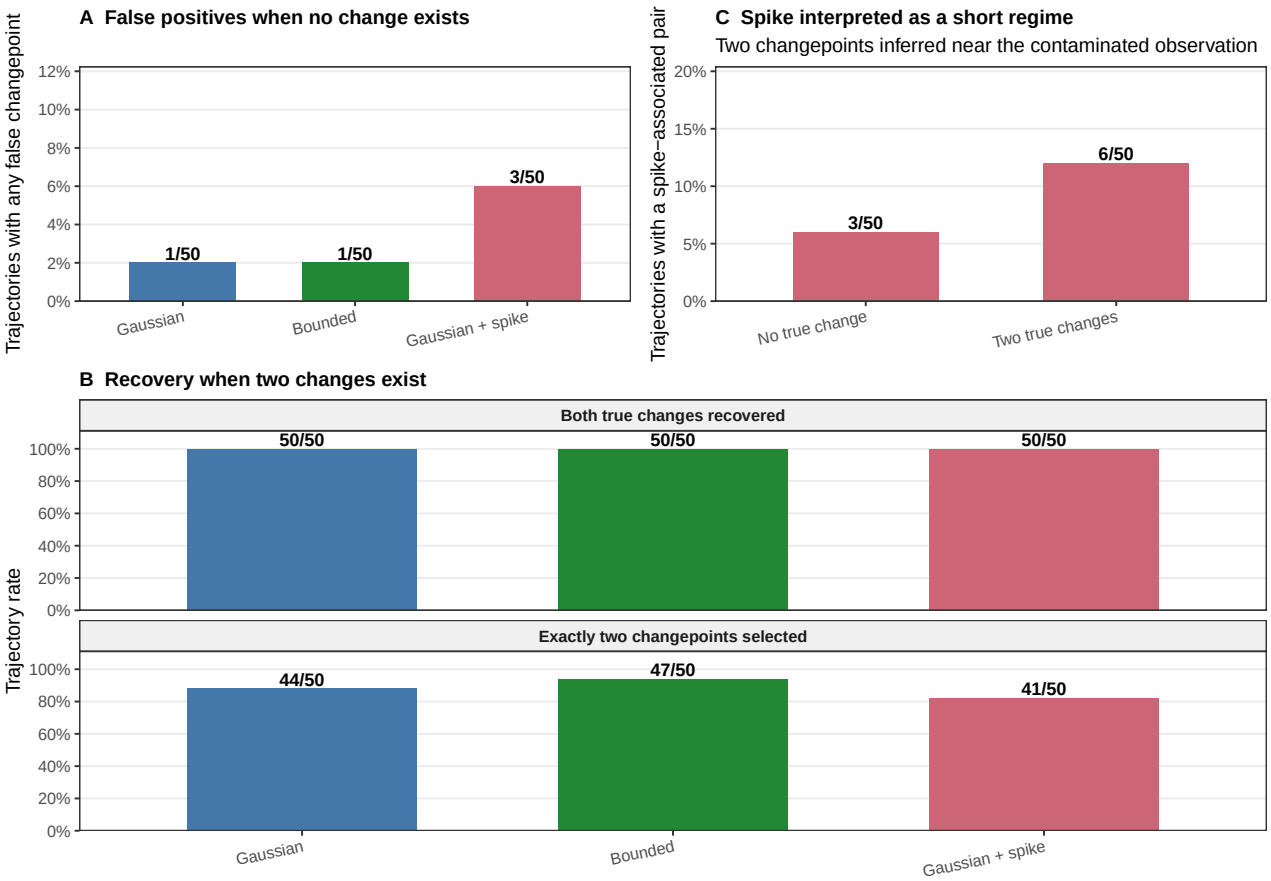


Figure S4. *Sensitivity of CPLASS to bounded observation errors and an isolated positional outlier.* Each condition contains 50 trajectories. **(A)** Trajectories with at least one false changepoint under the null. **(B)** Recovery of both true changepoints and selection of exactly two changepoints under the alternative. **(C)** Trajectories in which at least two changepoints were inferred within two sampling indices of the outlier. Bar labels give exact counts. The displayed outcomes were identical for $\eta = 0$ and $\eta = 1$ and are shown once.

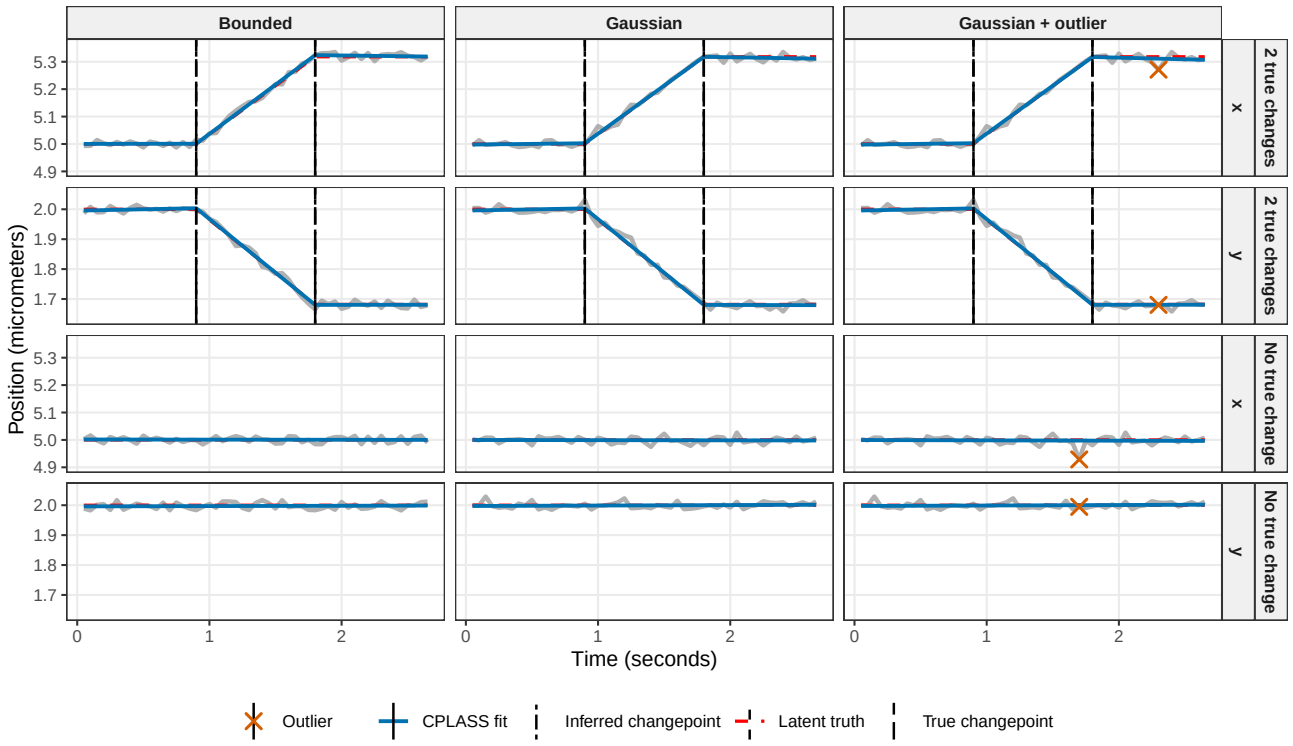


Figure S5. *Representative CPLASS fits under the three observation-error conditions.* The first two rows show the x and y coordinates under the alternative, and the last two rows show the corresponding coordinates under the null. The representative replicate was fixed before inspecting fit quality. Gray curves show observations, red dashed curves show the latent generating trajectories, and blue curves show CPLASS fits. Orange crosses mark the isolated outliers. Black long-dashed and double-dashed vertical lines mark true and inferred changepoints, respectively.

proposals as a stochastic search for the changepoint configuration maximizing the penalized criterion; the resulting chain is not used for posterior inference.

For a trajectory containing n observations in dimension d and a candidate model containing k segments, criterion evaluation is dominated by the continuous piecewise-linear least-squares calculation. With the practitioner-specified bound $K_{\max}$ fixed, the per-iteration cost is approximately linear in n and d, up to matrix-factorization and proposal costs. A search using S starts and at most I iterations therefore has approximate cost $O(S\,I\,d\,n)$ for fixed $K_{\max}$. The independent starts can be evaluated in parallel. The current R implementation is specialized to two-dimensional trajectories; consequently, the empirical benchmarks below vary n but not d.

For single-chain analyses, runtime increased gradually with trajectory length and with the maximum number of stochastic-search iterations (Supplementary Figure S6). For synthetic trajectories containing between 100 and 1200 observations, a 5000-iteration CPLASS run required a median of 0.73–1.97 seconds, whereas a 1000-iteration run required 0.15–0.44 seconds.

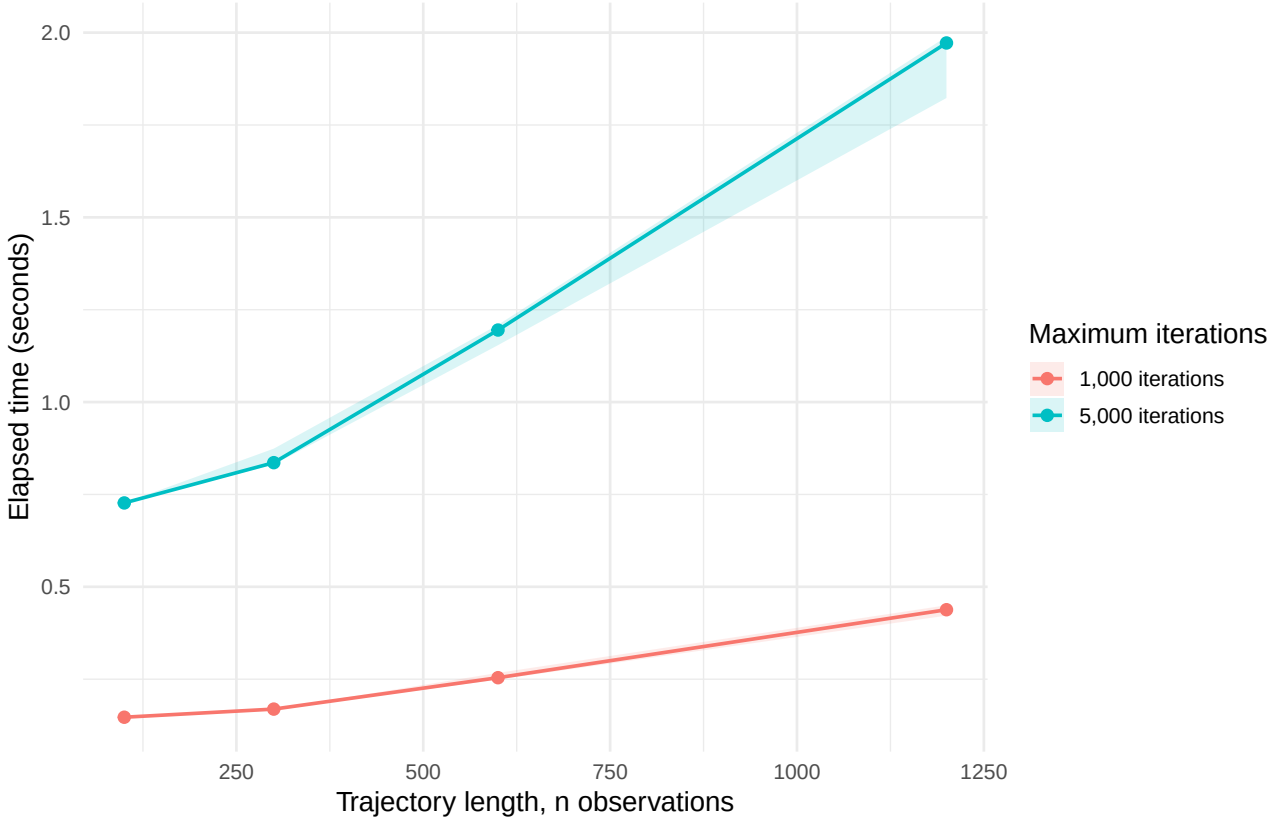


Figure S6. *Single-chain runtime as a function of trajectory length.* Synthetic trajectories containing 100–1200 observations were analyzed using maximum search lengths of either 1000 or 5000 iterations. Points show median elapsed times across five independent runs, and shaded regions show the interquartile ranges.

We next evaluated the multistart implementation using a synthetic trajectory containing 600 observations, with a maximum of 3000 iterations per start (Supplementary Figure S7). Parallel execution substantially reduced wall-clock time. Ten starts required a median of 5.48 seconds using one CPU core

and 1.61 seconds using seven CPU cores, corresponding to an approximately 3.4-fold speedup.

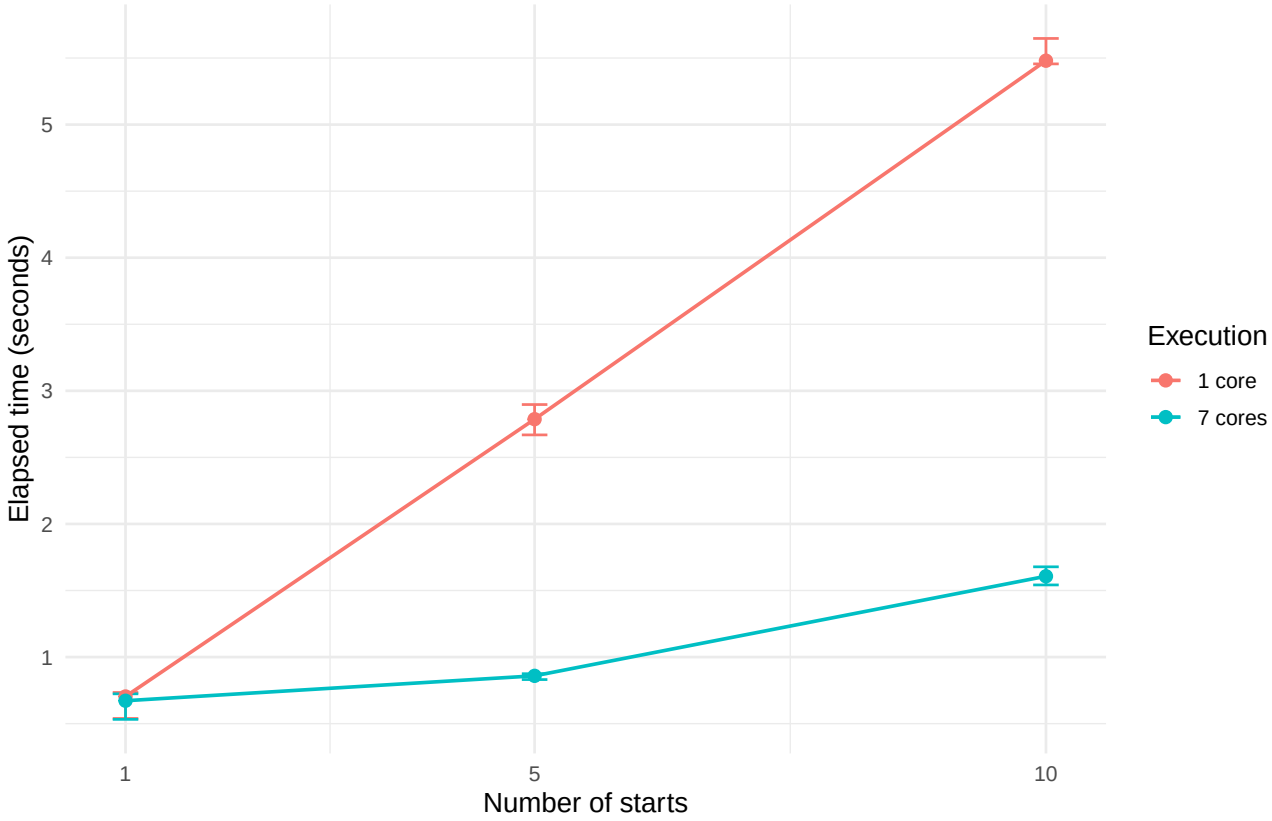


Figure S7. *Runtime of the CPLASS multistart procedure.* A synthetic trajectory containing 600 observations was analyzed using a maximum of 3000 search iterations per start. Results are shown for serial execution on one CPU core and parallel execution on seven CPU cores. Points show median elapsed times across five repeated runs, and error bars show interquartile ranges. Parallel execution provided approximately 3.2-fold and 3.4-fold speedups for five and ten starts, respectively.

We additionally benchmarked the production configuration: 16 deterministic starts, a maximum of 5000 iterations per start, and automatic early stopping. We considered $n \in \{53, 103, 203, 403\}$ under both a stationary null model and an alternative containing two changepoints, with five independently simulated trajectories per condition.

For the two-changepoint trajectories, median runtime increased from 10.78 seconds at $n = 53$ to 17.78 seconds at $n = 403$. For stationary trajectories, the corresponding medians were 3.12 and 4.07 seconds. All 16 starts produced valid fits for every trajectory, and automatic early stopping occurred in all 40 fits. The highest-scoring multistart solution selected the correct number of segments in all 40 trajectories (Supplementary Table S3).

Table S3. *Runtime and multistart diagnostics for the production benchmark.* Each condition contains five trajectories fitted using 16 starts. Runtime is reported as median [interquartile range] in seconds. Agreement indicates the number of trajectories for which all 16 starts selected the same number of segments. Correct best fit indicates whether the highest-scoring multistart solution selected the true segment count.

n	Model	Runtime (s)	Agreement	Correct best fit
53	Null	3.12 [3.09, 3.16]	5/5	5/5
53	Alternative	10.78 [10.75, 10.84]	4/5	5/5
103	Null	2.97 [2.95, 3.06]	5/5	5/5
103	Alternative	11.86 [11.69, 11.88]	4/5	5/5
203	Null	3.43 [3.23, 3.45]	5/5	5/5
203	Alternative	14.48 [13.90, 14.79]	3/5	5/5
403	Null	4.07 [4.02, 4.21]	5/5	5/5
403	Alternative	17.78 [17.48, 18.08]	0/5	5/5

Because CPLASS employs the chain for stochastic optimization rather than posterior sampling, conventional posterior-mixing diagnostics such as effective sample size and $\widehat{R}$ are not applicable. We instead examined the accepted-state criterion, its running maximum, the iteration of the final improvement, and agreement across independent starts.

Supplementary Figure S8 shows the criterion stabilization for one fixed representative trajectory from each benchmark condition. In every case, the final best criterion value was found before the last 20% of post-burn-in iterations. Longer trajectories containing two changepoints generally required more iterations to reach their final best score.

All 16 starts selected the same number of segments for every stationary trajectory. Under the alternative, all starts agreed for 4 of 5 trajectories at $n = 53$, 4 of 5 at $n = 103$, 3 of 5 at $n = 203$, and none at $n = 403$. Thus, some individual starts terminated at different lower-scoring local solutions, particularly for longer trajectories. Nevertheless, after retaining the highest-scoring solution across the 16 starts, CPLASS selected the correct three-segment model for all 20 alternative trajectories. This result illustrates the practical value of the multistart procedure.

S.6. Implementation details for the comparison methods

All methods were evaluated using the same frozen trajectories described in Section 3.3. Representative alternative trajectories from the three simulation regimes are shown in Supplementary Figure S9.

CPLASS was applied jointly to the two spatial coordinates using a common segmentation. CPOP, NOT, first-order trend filtering, and PLANT were applied separately to the x- and y-coordinate traces. PLANT is formulated for scalar time traces and was applied by Sosa-Costa et al. [3] to coordinate traces obtained from cell-tracking experiments; it does not impose a joint common segmentation across spatial coordinates. Although the framework of Yin et al. [4] includes a multidimensional extension, the publicly released serial executable that we were able to reproduce, `Single-DV`, accepts one scalar trajectory at a time. We therefore applied this executable separately to the two spatial coordinates.

For every coordinate-wise analysis, the final inferred changepoint set was the exact union

$$\widehat{\mathcal{M}} = \widehat{\mathcal{M}}_x \cup \widehat{\mathcal{M}}_y.$$

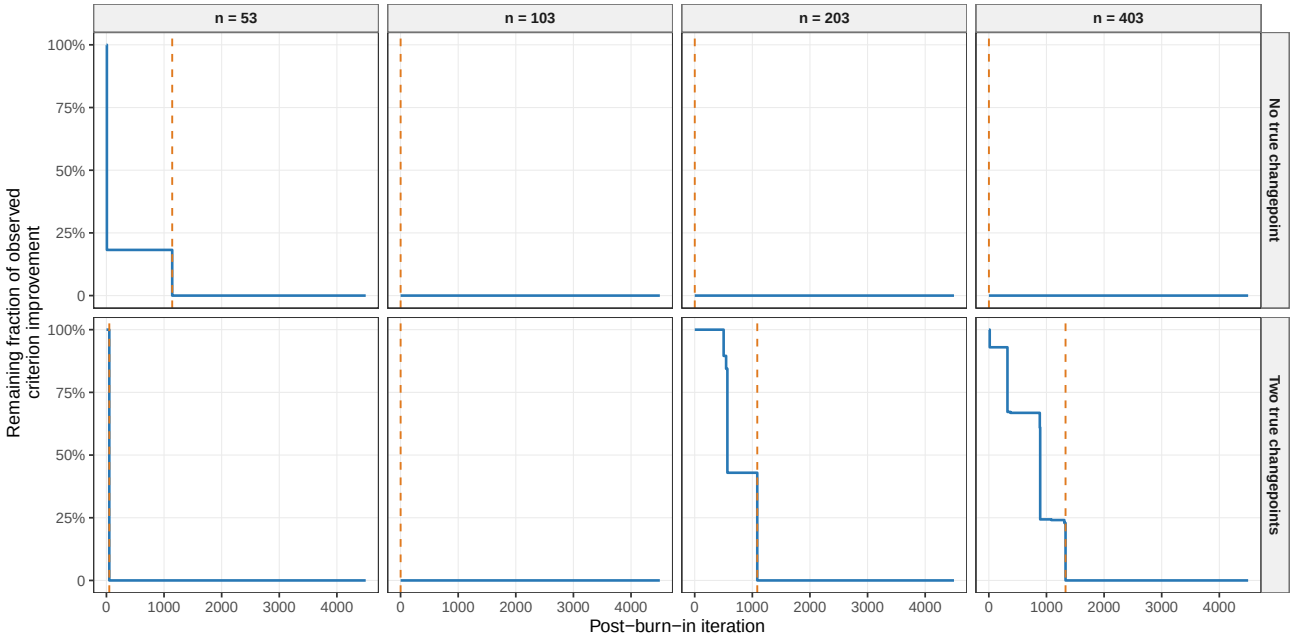


Figure S8. *Criterion stabilization for representative CPLASS stochastic-search chains.* Curves show the fraction of the total observed criterion improvement that remained at each post-burn-in iteration; zero indicates that the final best score obtained during the run had been reached. Orange dashed lines mark the final improvement. A curve at zero from the first displayed iteration indicates that the initial post-burn-in configuration remained the best. Columns correspond to trajectory length, with stationary trajectories in the top row and two-changepoint trajectories in the bottom row. In every condition, the running maximum remained unchanged throughout at least the final 20% of iterations.

No proximity-based merging or other post-processing was applied, because doing so would introduce an additional rule and tuning parameter not specified by the released implementations.

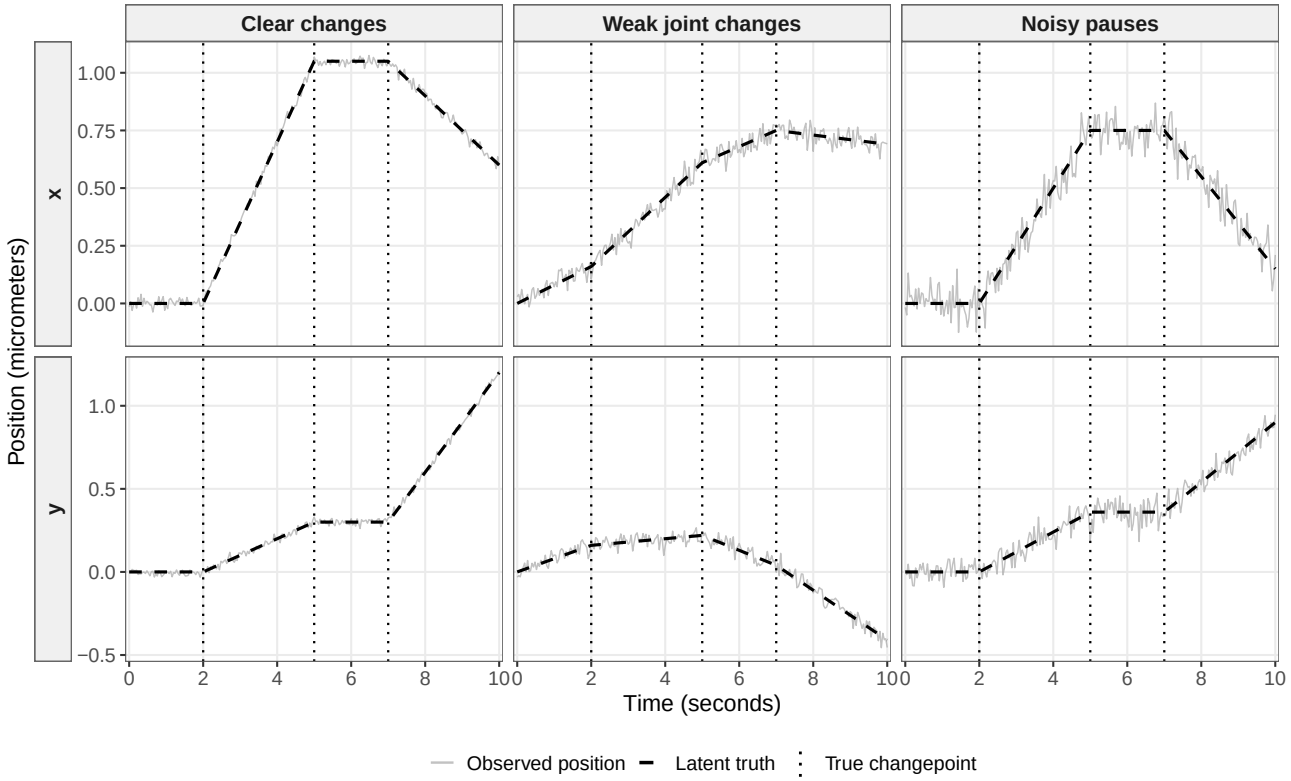


Figure S9. *Representative trajectories from the method-comparison simulation regimes.* Columns show the clear-change, weak-joint, and noisy-pause regimes, and rows show the x- and y-coordinates. Light-gray curves show the observed positions, black dashed curves show the continuous piecewise-linear generating trajectories, and black dotted vertical lines mark the generating changepoints at 2, 5, and 7 s. Replicate 1, selected by replicate number without reference to method performance, is shown for each regime. All trajectories were observed at 25 Hz for 10 s; their velocity and observation-error parameters are reported in Table 1 of the main manuscript.

Table S4. Software implementations and settings used in the method-comparison experiment.

Method	Implementation	Settings
CPLASS	Version 0.1.1; commit `d350fab`; archived tag `reviewer2-core-locked-v1`	Joint two-dimensional fit; $\gamma = 1.01$, $\eta = 1$, $s_{\text{cap}} = 1$, 16 deterministic starts, 5000 maximum iterations, 500 burn-in iterations, $K_{\max} = 30$, a minimum changepoint gap of one observation, and automatic stopping patience.

Continued on the next page

Method	Implementation	Settings
CPOP	R package `cpop`, version 1.0.9	Applied separately to x and y. Package defaults were used: penalty $2 \log n$, coordinate-specific default noise estimation, no minimum segment-length restriction, and exact pruning. Coordinate-wise changepoints were combined by exact union.
NOT	R package `not`, version 1.6	Applied separately to x and y using $M = 10{,}000$ random intervals, the continuous piecewise-linear contrast `pcwsLinContMean`, information-criterion selection, and the strengthened Schwarz information criterion penalty. Coordinate-wise changepoints were combined by exact union.
Trend filtering	R package `genlasso`, version 1.6.1	First-order trend filtering was applied separately to x and y. The tuning parameter was selected independently for each coordinate using five-fold cross-validation and the one-standard-error rule. Slope changes larger than the numerical tolerance 10^{-8} were retained, and coordinate-wise changepoints were combined by exact union.
PLANT	Authors' MATLAB implementation; commit `657ed550`; MATLAB R2025a	PLANT accepts a two-column time–response trace, in which the first column is time and the second is the scalar position being segmented. We therefore applied PLANT separately to the x- and y-coordinate traces using minimum segment length $N_{\min} = 3$, confidence level 0.95, and the known generating value of σ. Batch copies differed from the public source only through removal of graphical progress-window commands. Coordinate-wise changepoints were combined by exact union.
Yin et al.	Authors' serial `Single-DV` implementation; tag `v1.0-CPD`; commit `fa38b105`	Although the published framework includes a multidimensional extension, the released `Single-DV` executable used here accepts one scalar trajectory at a time. We therefore applied it separately to the x- and y-coordinate traces using sampling interval $\Delta = 0.04$ s and significance level $\alpha = 0.05$. The single-process branch did not contain the required `critical_values.h` header; the corresponding header from the authors' parallel branch was supplied without modifying the detection code. Coordinate-wise changepoints were combined by exact union.

For all methods, localization performance was evaluated using maximum one-to-one matching between inferred and generating changepoints. The primary analysis allowed a localization difference of at most five observations (0.20 s); a stricter analysis using two observations (0.08 s) gave the same qualitative conclusions. Type I error and the inferred-to-true changepoint-count ratio do not depend on the localization tolerance.

Because the implementations span R, MATLAB, and compiled C and PLANT was executed on separate hardware, recorded execution times are not interpreted as a formal speed ranking. Computational scaling and production-style runtime of the current CPLASS implementation are reported separately in

Supplementary Section S.5.

S.7. Sensitivity to the speed-regularization coefficient

The dimensionless speed regularizer is

$$\mathrm{pen}_{\mathrm{speed}}(r) = \eta \sum_{j=1}^{K(r)} h\left(\frac{\widehat{s}_{j,r}}{s_{\mathrm{cap}}} - 1\right),$$

where $s_{\mathrm{cap}} > 0$ is a physically specified reference speed and $\eta \geq 0$ controls the strength of the regularization. Throughout this analysis, we used $s_{\mathrm{cap}} = 1\,\mu\mathrm{m/s}$ and considered

$$\eta \in \{0, 0.25, 0.5, 1, 2, 4\}.$$

The value s_{cap} is a soft reference threshold rather than a hard upper bound: fitted speeds above it remain admissible when sufficiently supported by the likelihood.

We first examined the low-speed null and alternative simulations used in the γ-sensitivity analysis. These simulations included path lengths $n = 53$ and $n = 203$, with 200 fixed trajectories per hypothesis and path length. For $\eta \in \{0.25, 0.5, 2, 4\}$, none of the 3,200 selected fits contained a segment speed above s_{cap}. The speed-penalty contribution to every selected fit was therefore zero, and the detection results were unchanged across these values of η. The corresponding analyses at $\eta = 0$ and $\eta = 1$, obtained from the $\gamma = 1.01$ experiment, showed the same behavior. Thus, the regularizer has no practical effect in settings where the fitted segment speeds remain below the reference speed.

We next examined a setting in which the regularizer was active. Among the 250 trajectories from the Cook simulation analyzed in Section 3.1.2, we identified the fixed subset of 26 trajectories whose selected unpenalized fit ($\eta = 0$) contained at least one segment speed above s_{cap}. We refitted each of these trajectories at every value of η using the same CPLASS settings: $\gamma = 1.01$, 16 deterministic starts, 5,000 maximum iterations, 500 burn-in iterations, $K_{\max} = 30$, a minimum changepoint gap of one observation, and automatic stopping patience.

Increasing η progressively reduced both the frequency and magnitude of excessive fitted speeds (Figure S10). At $\eta = 0$, all 26 selected fits exceeded s_{cap}, the median trajectory-level maximum speed was $2.42\,\mu$m/s, and the overall maximum was $5.49\,\mu$m/s. At $\eta = 1$, 13 of the 26 fits exceeded s_{cap}, the median maximum speed decreased to $1.02\,\mu$m/s, and the overall maximum decreased to $2.36\,\mu$m/s. Increasing η to 2 or 4 produced smaller additional reductions in fitted speed and altered more inferred changepoint locations relative to the unpenalized analysis.

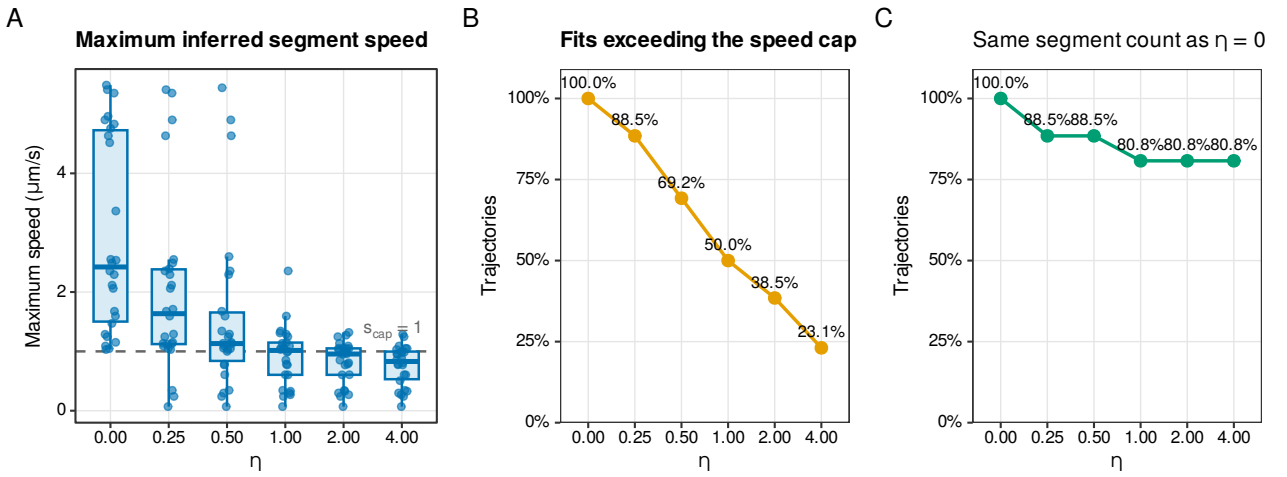


Figure S10. *Sensitivity of CPLASS to the dimensionless speed-regularization coefficient η.* Results are shown for the fixed subset of 26 Cook-simulation trajectories whose selected unpenalized fit contained at least one segment speed exceeding $s_{\text{cap}} = 1\,\mu$m/s. **(A)** Distribution of the maximum inferred segment speed for each value of η. Points represent individual trajectories, boxes show the interquartile range and median, and the horizontal dashed line marks s_{cap}. **(B)** Percentage of trajectories whose selected fit contains at least one segment speed exceeding s_{cap}. **(C)** Percentage of trajectories for which the selected number of segments agrees with the corresponding unpenalized fit at $\eta = 0$. Increasing η progressively reduces extreme fitted speeds, while the majority of trajectories retain the same segment count.

Because this subset was selected specifically for pathological unpenalized speeds, it is used to evaluate the behavior of the regularizer rather than to estimate general detection accuracy. The results support $\eta = 1$ as a moderate default: it substantially reduced extreme fitted speeds when the regularizer was active, while stronger values provided diminishing additional reductions and changed more changepoint locations. We emphasize that η remains a user-adjustable coefficient rather than a universally optimal constant.

Finally, the consistency argument continues to hold after adding the optional dimensionless speed regularizer. Under the conditions of the consistency result, and for fixed η, the speed-regularization term is asymptotically dominated by the strengthened Schwarz information-criterion component $(\log n)^{\gamma}\rho$, with $\gamma > 1$, and therefore does not alter selection consistency. Setting $\eta = 0$ recovers the sSIC-only criterion, whereas fixed $\eta > 0$ allows scientifically informed speed regularization to influence finite-sample segmentation without changing the large-sample conclusion.

References

1. K. J. Cook, N. Rayens, L. Do, C. K. Payne, S. A. McKinley, Considering experimental frame rates and robust segmentation analysis of piecewise-linear microparticle trajectories, *Math. Biosci. Eng.*, **22** (2025), 2595–2626. https://doi.org/10.3934/mbe.2025095
2. C. P. Robert, G. Casella, *Monte Carlo Statistical Methods*, Springer Texts in Statistics. Springer, New York, NY, 2nd edition, 2004. https://doi.org/10.1007/978-1-4757-4145-2
3. A. Sosa-Costa, I. K. Piechocka, L. Gardini, F. S. Pavone, M. Capitanio, M. F. Garcia-Parajo, et al., PLANT: A method for detecting changes of slope in noisy trajectories, *Biophys. J.*, **114** (2018), 2044–2051. https://doi.org/10.1016/j.bpj.2018.04.006
4. S. Yin, N. Song, H. Yang, Detection of velocity and diffusion coefficient change points in single-particle trajectories, *Biophys. J.*, **115** (2018), 217–229. https://doi.org/10.1016/j.bpj.2017.11.008



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)