RECAP: CALIBRATING CHANGE-POINT TESTS WITH NO-CHANGE REFERENCE SEQUENCES

Anonymous authors
Paper under double-blind review

ABSTRACT

Testing an ordered sequence for a distributional change is difficult when a useful statistic has an unknown null distribution. We address this problem by using historical sequences known to contain no change as a reference sample. We compute the same statistic for the test sequence and each reference sequence, then use the resulting statistics to form a randomized conformal p-value. If these statistics are exchangeable under the null, the p-value controls Type I error in finite samples. Each statistic may be computed from its own sequence, from the pooled sample of the reference sequences and the test sequence, or from that pooled sample together with optional sequences labelled as containing a change. We also establish finite-sample Type I error control under specified forms of heterogeneity across sequences. A multiscale extension detects multiple changes while retaining the same validity guarantee. Numerical studies demonstrate that our method yields empirical Type I error close to the nominal level and comparable power. We apply the method to detect activity transitions in wearable-sensor data and screen industrial screw-driving operations for anomalies.

1 INTRODUCTION

Change-point detection asks whether and where the data-generating distribution changes along an ordered sequence. It is used in epidemic surveillance, cancer genomics and primatology (Dehning et al., 2020; Chiang et al., 2009; Sandel et al., 2026). Recent work has broadened change-point analysis, with methods for high-dimensional data (Zhang et al., 2010; Cho & Fryzlewicz, 2015; Jirak, 2015; Wang & Samworth, 2018; Liu et al., 2020; Zhang et al., 2022; Wang & Feng, 2023; Xu et al., 2024) and nonparametric methods (Zou et al., 2014; Chen & Zhang, 2015; Arlot et al., 2019; McGonigle & Cho, 2025).

Flexible learning-based methods can detect complex distributional changes (Londschien et al., 2023; Li et al., 2024; Kanrar et al., 2025; Li et al., 2026), but the null distribution of a learned statistic can be difficult to derive. Analytic calibration is usually tied to a specified model or asymptotic regime, while resampling a learned statistic may require refitting the model for each resample (Londschien et al., 2023). Li et al. (2024) take a supervised learning perspective on change-point detection, formulating it as binary classification of sequences labelled as containing or not containing a change. This approach requires labelled training sequences from both classes; in practice, sequences containing a change can be rare and costly to label, so the training sample may be imbalanced (Anastasiou & Cribben, 2024).

Historical sequences known to contain no change may instead be abundant in many applications. In repeated batch production, past successful batches form a reference distribution for monitoring new batches (Nomikos & MacGregor, 1995). In wearable monitoring, long baseline records provide a reference for detecting later physiological changes (Mishra et al., 2020). We propose ReCaP (reference-calibrated p-values) to test for a distributional change within a test sequence using historical reference sequences. We summarize evidence of a change within each sequence by a real-valued *detection score*, with larger values indicating stronger evidence. We apply the same scoring procedure to each reference sequence and the test sequence, then compare the test score with the reference scores to form a randomized conformal p-value. The same calibration rule applies to different detection scores, including cumulative sum (CUSUM) statistics, scores computed with random forests, and fitted probabilities from neural networks.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

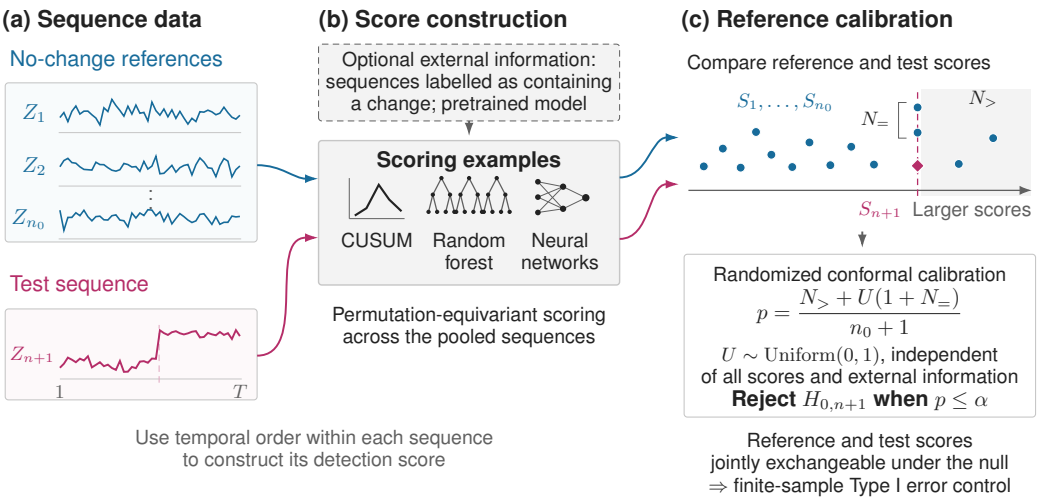

Figure 1: Schematic overview of ReCaP. Blue denotes no-change references and their scores; magenta denotes the test sequence and its score. Scoring examples include CUSUM statistics, classifier log-likelihood ratio scores computed with random forests, and fitted probabilities from neural networks. $N_>$ and $N_=$ count reference scores strictly greater than and equal to the test score, respectively.

Our contributions are as follows.

- We develop ReCaP, a framework for calibrating change-point tests with reference sequences. When the reference scores and the test score are exchangeable under the null, the p-value is uniform and the resulting test controls Type I error in finite samples. We allow observations within each sequence to be dependent over time.
- We provide three practical score constructions that use increasing amounts of information: each sequence alone, the pooled sample of the reference sequences and the test sequence, and that pooled sample together with optional sequences labelled as containing a change. The latter two constructions must treat all sequences in the pooled sample symmetrically. We also construct multiscale scores by maximizing over a common collection of intervals.
- We establish finite-sample Type I error control in two settings with heterogeneity across sequences. One permits sequence-specific location shifts and positive coordinatewise rescalings, so the sequences need not have the same distribution. The other permits nonexchangeable reference and test scores and uses an adjusted cutoff under a Kullback–Leibler condition.
- We extend ReCaP to multiple test sequences by using the detection scores from the reference and test sequences as inputs to the multiple testing procedure of Marandon et al. (2024). This gives finite-sample false discovery rate control under the stated exchangeability condition.

Relation to existing methods. Li et al. (2024) study out-of-sample misclassification error, whereas ReCaP tests the no-change hypothesis by comparing the test score with reference scores; sequences labelled as containing a change are optional. Conformal methods for outlier and novelty detection use a reference sample to construct a p-value for each test observation (e.g., Bates et al., 2023; Marandon et al., 2024). ReCaP applies this construction at the sequence level and uses temporal order to construct a detection score for a distributional change within each sequence. Like ReCaP, Cui et al. (2026) test for a distributional change within an ordered sequence. They compute a score for each observation, rank the scores within the sequence and calibrate an aggregate of the ranks by permutation. This uses only the test sequence but requires the test statistic to be an aggregate of within-sequence ranks; ReCaP instead calibrates a sequence-level detection score against external reference scores and allows statistics outside this form.

2 REFERENCE CALIBRATION

2.1 TESTING PROBLEM AND DATA SOURCES

Suppose each sequence consists of T ordered observations, and let $Z_{n+1} = (Z_{n+1,1}, \ldots, Z_{n+1,T})$ denote the test sequence. We test the null hypothesis

$$H_{0,n+1}: \quad Z_{n+1,1}, \ldots, Z_{n+1,T} \text{ are identically distributed.}$$

The alternative is that the distribution of $Z_{n+1,t}$ changes with t. We allow observations within each sequence to be dependent over time. We also observe n historical sequences $Z_i = (Z_{i,1}, \ldots, Z_{i,T})$, $i = 1, \ldots, n$. The first n_0 form the no-change reference sample $\mathcal{D}_0 = \{Z_1, \ldots, Z_{n_0}\}$. The remaining sequences, if any, are labelled as containing a change and form $\mathcal{D}_1 = \{Z_{n_0+1}, \ldots, Z_n\}$.

2.2 RANDOMIZED CONFORMAL p-VALUE

Let $I = \{1, \ldots, n_0\} \cup \{n+1\}$ index the reference sequences and the test sequence, and let $\mathbf{Z} = (Z_i : i \in I)$ denote their pooled sample. Let $\mathcal{G}$ be the σ-field generated by $\mathcal{D}_1$ and any auxiliary information external to $\mathbf{Z}$ that is used by the scoring procedure, such as a pretrained model. Conditional on $\mathcal{G}$, the procedure produces a score vector $(S_i : i \in I)$ from $\mathbf{Z}$. It is permutation-equivariant if reordering $\mathbf{Z}$ produces the same reordering of the score vector. For a randomized procedure, this requirement applies to the conditional distribution of the score vector given $\mathbf{Z}$ and $\mathcal{G}$.

Define p by

$$p = \frac{1}{n_0 + 1}\left[\sum_{i=1}^{n_0} \mathbf{1}\{S_i > S_{n+1}\} + U\left\{1 + \sum_{i=1}^{n_0} \mathbf{1}\{S_i = S_{n+1}\}\right\}\right], \tag{1}$$

where $U \sim \mathrm{Uniform}(0,1)$ is independent of $\mathcal{G}$ and $(S_i : i \in I)$ and randomizes ties. Equation (1) is the standard randomized conformal p-value constructed from the test score and the reference scores (e.g., Vovk et al., 2003).

Proposition 1 (Finite-sample validity)**.** *If, under $H_{0,n+1}$, $(S_i : i \in I)$ is exchangeable conditional on $\mathcal{G}$, then, for every $\alpha \in [0,1]$, $\Pr(p \le \alpha \mid \mathcal{G}) = \alpha$ almost surely.*

Thus the test that rejects $H_{0,n+1}$ when $p \le \alpha$ has conditional, and hence unconditional, Type I error α. A simple sufficient condition is that, under $H_{0,n+1}$, the sequences Z_i, $i \in I$, are i.i.d. conditional on $\mathcal{G}$ and the scoring procedure is permutation-equivariant.

2.3 CONSTRUCTION OF DETECTION SCORES

The three examples below use increasing amounts of information: each sequence alone, the pooled sample $\mathbf{Z}$, and $\mathbf{Z}$ together with the optional sample $\mathcal{D}_1$.

Example 1 (Scores computed within each sequence)**.** *Let $Z_{i,t} \in \mathbb{R}^d$ and write $\bar{Z}_{i,a:b} = (b-a+1)^{-1}\sum_{u=a}^{b} Z_{i,u}$. For $q \in [1,\infty]$, $\|\cdot\|_q$ denotes the ℓ_q norm. A cumulative sum (CUSUM) score for a mean change is*

$$S_{i,q} = \max_{1 \le t < T} \left\| \{t(T-t)/T\}^{1/2} (\bar{Z}_{i,1:t} - \bar{Z}_{i,t+1:T}) \right\|_q, \qquad i \in I.$$

The choice $q = 2$ aggregates evidence across coordinates (Zhang et al., 2010), whereas $q = \infty$ takes the largest coordinatewise contrast (Jirak, 2015). For a fixed q, use $S_i = S_{i,q}$.

For a learning-based alternative, use the classifier log-likelihood ratio (CLR) score. For a pilot split $\tau \in \{1, \ldots, T-1\}$, such as $\tau = \lfloor T/2 \rfloor$, fit a classifier to the observations in Z_i, labelling those before and after τ as the two classes. Let $\widehat{p}^{\,\tau}_{i,1}$ and $\widehat{p}^{\,\tau}_{i,2}$ be the fitted probability functions for the pre- and post-split classes, respectively, and suppose that their values lie in $(0,1)$. The fixed-pilot CLR score, which may be used directly as S_i, is

$$S_{\mathrm{CLR}}(Z_i; \tau) = \max_{1 \le t < T} \left\{ \sum_{u \le t} \log \frac{T \widehat{p}^{\,\tau}_{i,1}(Z_{i,u})}{\tau} + \sum_{u > t} \log \frac{T \widehat{p}^{\,\tau}_{i,2}(Z_{i,u})}{T - \tau} \right\}.$$

To reduce sensitivity to the choice of τ, we recommend the variant of Londschien et al. (2023) that uses multiple pilot splits and denote its final CLR score by S_i; see Appendix B.1.

Example 2 (Scores computed from the pooled sample)**.** *The second construction combines several candidate scores after standardizing them over* $\mathbf{Z}$*. Let* $Q \subset [1, \infty]$ *be a finite set indexing the CUSUM scores* $S_{i,q}$ *in Example 1, with different values of* q *responding to different coordinate patterns (Liu et al., 2020; Zhang et al., 2022; Wang & Feng, 2023). For* $q \in Q$*, define* $\bar{S}_q = (n_0+1)^{-1} \sum_{i \in I} S_{i,q}$ *and* $v_q^2 = n_0^{-1} \sum_{i \in I} (S_{i,q} - \bar{S}_q)^2$*. If* $v_q = 0$ *for every* $q \in Q$*, set* $S_i = 0$ *for every* $i \in I$*; otherwise, define*

$$S_i = \max_{q \in Q:\, v_q > 0} (S_{i,q} - \bar{S}_q)/v_q, \qquad i \in I.$$

Because the pooled means and variances are unchanged by reorderings of $\mathbf{Z}$*, the score vector is permutation-equivariant. If* $\mathbf{Z}$ *is exchangeable conditional on* $\mathcal{G}$ *under* $H_{0,n+1}$*, Proposition 1 gives finite-sample validity.*

Example 3 (Scores using sequences labelled as containing a change)**.** *When* $\mathcal{D}_1$ *is available, fit a binary classifier by assigning label* 0 *to every sequence in* $\mathbf{Z}$ *and label* 1 *to every sequence in* $\mathcal{D}_1$*. For example, the classifier may be a neural network, as in Li et al. (2024). Let* $\widehat{\eta}(z)$ *denote the fitted probability of label* 1 *for a sequence* z*, and set*

$$S_i = \widehat{\eta}(Z_i), \qquad i \in I.$$

The classification procedure must be permutation-equivariant under reorderings of $\mathbf{Z}$*. If* $\mathbf{Z}$ *is exchangeable conditional on* $\mathcal{G}$ *under* $H_{0,n+1}$*, the scores are exchangeable conditional on* $\mathcal{G}$*, so Proposition 1 gives* $p \mid \mathcal{G} \sim \mathrm{Uniform}(0,1)$*. Class imbalance does not by itself affect this finite-sample validity conclusion.*

Algorithm 1 summarizes the procedure.

Algorithm 1: ReCaP: reference-calibrated change-point testing

Input: No-change references $\mathcal{D}_0 = \{Z_1, \ldots, Z_{n_0}\}$; test sequence Z_{n+1}; a permutation-equivariant scoring procedure $\mathcal{S}$; optional external information; level α.

Scoring: Apply $\mathcal{S}$ to $\mathbf{Z} = (Z_i : i \in I)$ using the available external information to obtain $(S_i : i \in I)$, where $I = \{1, \ldots, n_0\} \cup \{n+1\}$.

Calibration: Set $p \leftarrow \mathrm{RankCalibrate}(S_1, \ldots, S_{n_0}; S_{n+1})$.

Decision: Set $\delta \leftarrow \mathbf{1}\{p \le \alpha\}$.

Function $\mathrm{RankCalibrate}(s_1, \ldots, s_{n_0}; s_\star)$**:**

  Count $N_> \leftarrow \sum_{i=1}^{n_0} \mathbf{1}\{s_i > s_\star\}$ and $N_= \leftarrow \sum_{i=1}^{n_0} \mathbf{1}\{s_i = s_\star\}$.

  Draw $U \sim \mathrm{Uniform}(0,1)$ independently of $\mathcal{G}$ and all scores.

  return $\{N_> + U(1 + N_=)\}/(n_0+1)$.

Output: Randomized conformal p-value p and decision δ; reject $H_{0,n+1}$ if $\delta = 1$.

2.4 Multiscale reference calibration

A test based on a full-sequence score may have low power against a change confined to a short interval. Let $\mathcal{J} = \{I_1, \ldots, I_L\}$ be a nonempty finite collection of intervals in $\{1, \ldots, T\}$, common to all sequences and chosen independently of $\mathbf{Z}$ conditional on $\mathcal{G}$. For each interval I_ℓ, apply a scoring procedure from Section 2.3 to the corresponding subsequences of the references and the test sequence, with any external information used by that procedure, to obtain scores $(S_{\ell,i} : i \in I)$. Any standardization must treat the pooled sequences symmetrically. We define the multiscale score

$$S_i = \max_{1 \le \ell \le L} S_{\ell,i}.$$

We apply equation (1) to these multiscale scores. Given $\mathcal{G}$ and $\mathcal{J}$, the full procedure producing all interval scores, including its joint randomization, must be permutation-equivariant under reorderings of $\mathbf{Z}$. If $\mathbf{Z}$ is exchangeable conditional on $\mathcal{G}$ under $H_{0,n+1}$, the multiscale scores are also exchangeable conditional on $\mathcal{G}$, and the resulting p-value is conditionally uniform; see Proposition A1 in Appendix A.2. Section 3.2 evaluates the construction for epidemic mean changes.

2.5 HETEROGENEITY ACROSS SEQUENCES

Location–scale-invariant scores. For each $i \in I$, suppose that $Z_{i,t} = \mu_i + D_i\varepsilon_{i,t}$, $t = 1, \ldots, T$, where $\mu_i \in \mathbb{R}^d$, D_i is a positive diagonal matrix and $\varepsilon_i = (\varepsilon_{i,1}, \ldots, \varepsilon_{i,T})$. Let ξ_i denote any randomization used when scoring sequence i; for a deterministic rule, take ξ_i to be a common fixed value. For a common scoring rule S, set $S_i = S(Z_i; \xi_i)$. We call S location–scale invariant if $S\big((\mu + D\varepsilon_1, \ldots, \mu + D\varepsilon_T); \xi\big) = S(\varepsilon; \xi)$ for every location vector μ, every positive diagonal matrix D, every sequence $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_T)$ and every value of ξ.

Theorem 1 (Validity under location–scale transformations)**.** *Under $H_{0,n+1}$, suppose that S is location–scale invariant and that $((\varepsilon_i, \xi_i) : i \in I)$ is exchangeable conditional on $\mathcal{G}$. Then, for every $\alpha \in [0, 1]$, the p-value in (1) satisfies*

$$\Pr(p \le \alpha \mid \mathcal{G}) = \alpha \quad \textit{almost surely}.$$

Remark 1. For example, if $Z_{i,t} = \mu_i + \varepsilon_{i,t}$, the CUSUM score in Example 1 is location invariant, so Theorem 1 applies whenever $(\varepsilon_i : i \in I)$ is exchangeable conditional on $\mathcal{G}$.

Corollary 1 (CLR scores computed with a random forest)**.** *Under the location–scale representation above, let S_i be either the fixed-pilot score $S_{\mathrm{CLR}}(Z_i; \tau)$ or the final multiple-pilot CLR score in Example 1, in both cases computed with the random-forest fitting rule of Londschien et al. (2023). Under $H_{0,n+1}$, if $((\varepsilon_i, \xi_i) : i \in I)$ is exchangeable conditional on $\mathcal{G}$, then, for every $\alpha \in [0, 1]$, the p-value in (1) satisfies $\Pr(p \le \alpha \mid \mathcal{G}) = \alpha$ almost surely.*

Remark 2. This random-forest fitting rule is invariant to positive rescaling of individual features (Londschien et al., 2023). Both CLR scores inherit this rescaling invariance and are also location invariant, so they are location–scale invariant; see Appendix A.4.

Nonexchangeable scores. If the reference scores and the test score are not exchangeable under the null, p need not be uniform. Motivated by work on conformal methods under distribution shift (Barber et al., 2023; Cauchois et al., 2024), we consider departures from uniformity in the lower tail, which can inflate Type I error. Under $H_{0,n+1}$, write $F_{\mathcal{G}}(t) = \Pr(p \le t \mid \mathcal{G})$. For $q \in [0, 1]$ and $t \in (0, 1)$, let $d_{\mathrm{KL}}(q, t) = q \log(q/t) + (1 - q) \log\{(1 - q)/(1 - t)\}$ denote the Kullback–Leibler divergence from $\mathrm{Bernoulli}(q)$ to $\mathrm{Bernoulli}(t)$, using the convention $0 \log 0 = 0$, and define $d^+_{\mathrm{KL}}(q, t) = d_{\mathrm{KL}}(q, t)\mathbf{1}\{q > t\}$.

Theorem 2 (Type I error control under a Kullback–Leibler bound)**.** *Let $\alpha \in (0, 1)$ and $\rho \ge 0$. Define $\alpha_\rho \in (0, \alpha]$ by*

$$d_{\mathrm{KL}}(\alpha, \alpha_\rho) = \rho.$$

If $\sup_{0<t\le\alpha} d^+_{\mathrm{KL}}\{F_{\mathcal{G}}(t), t\} \le \rho$ almost surely, then

$$\Pr(p \le \alpha_\rho \mid \mathcal{G}) \le \alpha \quad \textit{almost surely}.$$

Thus, when the stated lower-tail bound holds with ρ, rejecting $H_{0,n+1}$ when $p \le \alpha_\rho$ controls conditional Type I error at level α. The supremum quantifies departures $F_{\mathcal{G}}(t) > t$ from the uniform benchmark t over $0 < t \le \alpha$ using d_{KL}; it is zero if $F_{\mathcal{G}}(t) \le t$ throughout this range. The adjusted cutoff satisfies $\alpha_0 = \alpha$, decreases with ρ, and obeys $\alpha_\rho = \alpha - \sqrt{2\rho\alpha(1 - \alpha)} + O(\rho)$ as $\rho \downarrow 0$, showing that, for fixed α, the reduction $\alpha - \alpha_\rho$ is of order $\sqrt{\rho}$.

To obtain a heuristic cutoff, sample B short blocks from the test sequence. For each $b = 1, \ldots, B$, take the corresponding block from every reference sequence, apply the same permutation-equivariant scoring procedure to these blocks, and compute the randomized conformal p-value p_b using equation (1). Let $\widehat{F}(t) = B^{-1} \sum_{b=1}^{B} \mathbf{1}\{p_b \le t\}$ be the empirical distribution function of these block p-values, and set

$$\widehat{\rho} = \sup_{0<t\le\alpha} d^+_{\mathrm{KL}}\{\widehat{F}(t), t\}.$$

Use the rejection rule $p \le \alpha_{\widehat{\rho}}$ if $\widehat{F}(\alpha) > \alpha$, and $p \le \alpha$ otherwise. Appendix C.3 evaluates this heuristic under changes in scale, dependence, and tail behavior, reporting its effects on both empirical size and power.

2.6 Testing multiple sequences

Let $\mathcal{D}_{\text{test}} = \{Z_{n+1}, \ldots, Z_{n+m}\}$ contain m test sequences. For $j = 1, \ldots, m$, let $H_{0,n+j}$ be the hypothesis that the observations in Z_{n+j} are identically distributed. Marandon et al. (2024) develop AdaDetect for novelty detection with several test observations, constructing one p-value per test observation from a reference sample and then applying a multiple testing procedure. In our application, the unit of analysis is an ordered sequence: we first use temporal order within each reference and test sequence to construct a detection score S_i, as in Section 2.3, and then use these sequence-level scores as inputs to AdaDetect. Specifically, split $\mathcal{D}_0$ into a training sample $\mathcal{D}_{0,\text{tr}}$ and a calibration sample $\mathcal{D}_{0,\text{cal}} = \{Z_i : i \in \mathcal{C}\}$, where $|\mathcal{C}| = \ell$. Fit a probabilistic classifier $\widehat{g}$ to distinguish $\{S_i : Z_i \in \mathcal{D}_{0,\text{tr}}\}$ from $\{S_i : Z_i \in \mathcal{D}_{0,\text{cal}} \cup \mathcal{D}_{\text{test}}\}$. For each calibration or test sequence, let $A_i = \widehat{g}(S_i)$ be the fitted probability of membership in the pooled calibration and test sample. Break ties among these outputs by independent randomization.

The score construction and the fitting of $\widehat{g}$, including tuning, must be permutation-equivariant over $\mathcal{D}_{0,\text{cal}} \cup \mathcal{D}_{\text{test}}$. In this subsection, $\mathcal{G}$ also includes $\mathcal{D}_{0,\text{tr}}$ and any information outside this pooled sample used to construct S_i or fit $\widehat{g}$. For $j = 1, \ldots, m$, define the p-value for test sequence Z_{n+j} by $p_j = (1 + \sum_{i \in \mathcal{C}} \mathbf{1}\{A_i > A_{n+j}\})/(\ell + 1)$. Apply the Benjamini–Hochberg procedure (Benjamini & Hochberg, 1995) at level α to $p_1, \ldots, p_m$, and let R_α be its rejection set.

Proposition 2 (False discovery rate control)**.** *Let $\mathcal{H}_0 \subseteq \{1, \ldots, m\}$ index the true null hypotheses. Suppose that, conditional on $\mathcal{G}$ and $(A_{n+j} : j \notin \mathcal{H}_0)$, $(A_i : i \in \mathcal{C})$ and $(A_{n+j} : j \in \mathcal{H}_0)$ are jointly exchangeable. Then*

$$\text{FDR}(R_\alpha) = \text{E}\{|R_\alpha \cap \mathcal{H}_0|/(1 \vee |R_\alpha|)\} \leq \alpha.$$

Thus, the simultaneous testing procedure controls the false discovery rate at level α in finite samples under the stated exchangeability condition. The conclusion follows from a conditional application of Marandon et al. (2024, Theorem 3.4) to the classifier outputs A_i.

3 Numerical studies

3.1 Gaussian mean changes

We first consider a high-dimensional Gaussian mean-change setting with $T = d = 200$. Let I_d be the $d \times d$ identity matrix. The observations are independent across sequences and time points. Each reference observation is distributed as $N_d(0, I_d)$. Under the null, each test observation is also distributed as $N_d(0, I_d)$. Under the alternative, the test observations at $t \leq T/2$ and $t > T/2$ have distributions $N_d(0, I_d)$ and $N_d(\delta, I_d)$, respectively, where only the first k entries of $\delta = (\delta_1, \ldots, \delta_d)^\mathsf{T}$ are nonzero and $\|\delta\|_2^2 = \Delta \in \{1, 2\}$. For $k = 1$, $\delta_1 = \Delta^{1/2}$. For $k > 1$, $\delta_1 = (0.1\Delta)^{1/2}$ and $\delta_j = \{0.9\Delta/(k-1)\}^{1/2}$ for $j = 2, \ldots, k$.

We compare Max, which takes the largest absolute standardized coordinatewise CUSUM over coordinates and candidate change points away from the sequence boundaries (Wang & Feng, 2023), and Sum, which sums the squared standardized coordinatewise CUSUMs over all coordinates and candidate change points (Wang et al., 2019). Comb combines their asymptotic p-values by Fisher's method (Wang & Feng, 2023); see Appendix C.1. For ReCaP, we use the score in Example 2 with $Q = \{\infty\}$, $Q = \{2\}$ and $Q = \{2, \infty\}$, and call the resulting tests ReCaP-Max, ReCaP-Sum and ReCaP-Comb, respectively. We apply all six tests at level 0.05; each ReCaP test uses $n_0 = 200$ reference sequences. Rejection probabilities are estimated from 1,000 Monte Carlo replications.

Table A1 in Appendix C.1 shows that the empirical sizes of the six tests are generally close to the nominal level 0.05 when all observations are mutually independent and distributed as $N_d(0, I_d)$. Figure 2 shows that Max and ReCaP-Max are more powerful than Sum and ReCaP-Sum for sparse changes, whereas Sum and ReCaP-Sum are more powerful for dense changes. ReCaP-Max and ReCaP-Sum have power close to Max and Sum, respectively, while ReCaP-Comb is close to the more powerful of ReCaP-Max and ReCaP-Sum.

We next change only the temporal dependence within each sequence, using a Gaussian AR(1) error process. Independently for each $i \in I$, draw $\varepsilon_{i,1}, \eta_{i,2}, \ldots, \eta_{i,T} \overset{\text{i.i.d.}}{\sim} N_d(0, I_d)$ and set $\varepsilon_{i,t} = 0.1\varepsilon_{i,t-1} + \sqrt{1 - 0.1^2}\,\eta_{i,t}$, $t = 2, \ldots, T$. Set $Z_{i,t} = \varepsilon_{i,t}$ for $i \leq n_0$ and $Z_{n+1,t} = \varepsilon_{n+1,t} +$

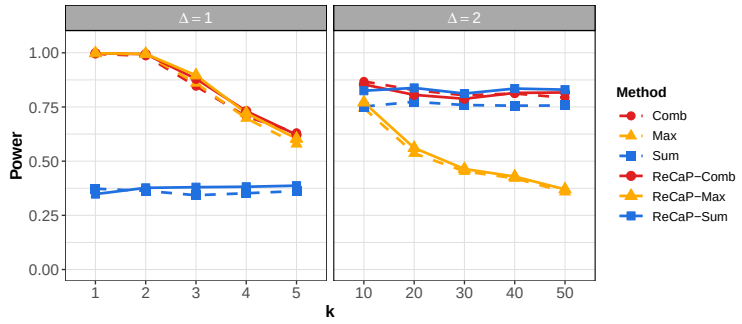

Figure 2: Empirical power as a function of the number k of affected coordinates when the Gaussian observations are mutually independent.

$\mathbf{1}(t > T/2)\delta$, with $\delta = 0$ under the null and the mean-change vectors specified above under the alternatives. When $\delta = 0$, the empirical sizes are 0.384, 1.000, and 1.000 for Max, Sum, and Comb, respectively, while those for ReCaP-Max, ReCaP-Sum, and ReCaP-Comb are 0.049, 0.062, and 0.059, respectively. The empirical sizes of the ReCaP tests are close to 0.05, whereas those of Max, Sum and Comb are much larger in this experiment. See Table A2 in Appendix C.1 for rejection probabilities under the mean-change alternatives with Gaussian AR(1) errors.

3.2 Multiscale detection of epidemic mean changes

We evaluate the multiscale construction in Section 2.4 for epidemic mean changes: the mean shifts over a subinterval and then returns to baseline. We use 50-dimensional Gaussian sequences of length $T = 200$ and $n_0 = 200$ no-change references, with observations independent over time. The alternatives combine dense shifts affecting all coordinates or sparse shifts affecting two coordinates with long or short change intervals, occupying 40% or 16% of the sequence, respectively. We compare ReCaP-Sum, ReCaP-Max and ReCaP-Comb with their multiscale versions. The latter use a common family of 40 intervals, selected independently of the data and including the full sequence. Each rejection probability is based on $2{,}000$ Monte Carlo replications at level 0.05. Appendix C.2 gives the shift magnitudes, interval sampling scheme and score standardization details.

Table 1 reports empirical sizes between 0.038 and 0.045, below the nominal level 0.05. In the settings considered, the multiscale versions generally have higher empirical power than their full-sequence counterparts. For both the original and multiscale procedures, ReCaP-Sum has higher empirical power than ReCaP-Max for dense shifts, whereas the reverse holds for sparse shifts.

Table 1: Empirical size and power for epidemic mean changes in Gaussian sequences, based on $2{,}000$ Monte Carlo replications.

Method	Size	Long interval (40%)		Short interval (16%)	
		Dense	Sparse	Dense	Sparse
ReCaP-Sum	0.043	0.533	0.338	0.156	0.137
Multiscale ReCaP-Sum	0.043	0.844	0.544	0.822	0.683
ReCaP-Max	0.045	0.139	0.490	0.093	0.156
Multiscale ReCaP-Max	0.038	0.122	0.784	0.134	0.886
ReCaP-Comb	0.043	0.405	0.481	0.129	0.160
Multiscale ReCaP-Comb	0.039	0.668	0.785	0.661	0.886

3.3 Real data

3.3.1 HASC activity transitions

We analyze acceleration records from six participants (person101–person106) in the Human Activity Sensing Consortium (HASC) Challenge 2011 corpus (Kawaguchi et al., 2011). Following Li

et al. (2024), we form length-700 sequences from six activities: stay, walk, jog, skip, stair up and stair down. A sequence spanning an activity transition is labelled as containing a change; all other sequences are labelled as containing no change.

In each of 100 repetitions, we draw a test sample $\mathcal{D}_{\text{test}}$ of $m = 100$ sequences, 50 labelled as containing a change and 50 labelled as containing no change, and a reference sample $\mathcal{D}_0$ of $n_0 = 200$ no-change sequences. Using these reference and test samples, we compare two pairs of procedures: random forest (RF) with ReCaP-RF and deep neural network (DNN) with ReCaP-DNN; the DNN pair uses additional sequences labelled as containing a change, as specified below.

RF and ReCaP-RF. Both methods use the within-sequence CLR score in Example 1 (with multiple pilot splits), computed with the default random-forest implementation in the Python package `changeforest` (Londschien et al., 2023). RF uses the package's default pseudo-permutation rule, whereas ReCaP-RF applies equation (1) to the resulting reference and test scores.

DNN and ReCaP-DNN. Both methods pretrain a neural network to learn a sequence representation, freeze its feature-extraction layers, and then fit only the final classification layer. Each length-700 acceleration sequence is augmented with its three elementwise-squared channels, standardized channelwise, and passed through a residual convolutional encoder. Global average pooling and fully connected layers produce a 30-dimensional representation. For each $r \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, we pretrain a separate network based on Li et al. (2024) using 2,000 labelled sequences outside $\mathcal{D}_0 \cup \mathcal{D}_{\text{test}}$, of which $2{,}000r$ are labelled as containing a change. The resulting frozen feature extractor supplies the representation used by both methods. We also draw a sample $\mathcal{D}_1(r)$ of sequences labelled as containing a change, disjoint from $\mathcal{D}_0$ and $\mathcal{D}_{\text{test}}$, with $|\mathcal{D}_1(r)| = \text{round}\{(n_0+1)r/(1-r)\}$. Thus r is the proportion of pretraining sequences labelled as containing a change and, through $|\mathcal{D}_1(r)|$, controls the amount of such labelled information used to fit the final layer. DNN fits the final layer once to distinguish $\mathcal{D}_0$ from $\mathcal{D}_1(r)$, whereas for each test sequence Z_{n+j}, ReCaP-DNN fits a separate final layer to distinguish $\mathcal{D}_0 \cup \{Z_{n+j}\}$ from $\mathcal{D}_1(r)$. DNN declares a test sequence to contain a change when its fitted probability of containing a change exceeds 0.5. ReCaP-DNN instead calibrates the fitted probability for Z_{n+j} against the reference probabilities from the corresponding test-specific fit using equation (1). See Appendix B.2 for implementation details.

We apply ReCaP-RF and ReCaP-DNN at the nominal level 0.05. Table 2 shows average false positive rates (FPRs) close to this level for both procedures, whereas RF and DNN have much larger FPRs in some settings. ReCaP-RF has a lower true positive rate (TPR) than RF but much lower false positive and misclassification error rates, and DNN and ReCaP-DNN have similar TPRs and MERs for $r \leq 0.7$. At $r = 0.9$, DNN labels essentially every test sequence as containing a change, whereas ReCaP-DNN has a much lower MER.

Table 2: Average false positive rate (FPR), true positive rate (TPR) and misclassification error rate (MER) over 100 repetitions of the HASC analysis.

Measure	RF	ReCaP-RF	$r = 0.1$		$r = 0.3$		$r = 0.5$		$r = 0.7$		$r = 0.9$	
			DNN	ReCaP-DNN	DNN	ReCaP-DNN	DNN	ReCaP-DNN	DNN	ReCaP-DNN	DNN	ReCaP-DNN
FPR	0.988	0.054	0.002	0.048	0.035	0.048	0.062	0.055	0.051	0.052	1.000	0.055
TPR	1.000	0.583	0.854	0.896	0.878	0.886	0.685	0.668	0.876	0.871	1.000	0.774
MER	0.494	0.236	0.074	0.076	0.078	0.081	0.189	0.193	0.087	0.090	0.500	0.141

3.3.2 Anomaly screening of industrial screw-driving operations

We further evaluate ReCaP on the S04 subset of the Industrial Screw Driving Dataset Collection (West & Deuse, 2025). Each record is one automated operation with torque, angle, and gradient measurements across four tightening stages. We retain the 4,618 operations containing all four stages. Operations labelled as normal with an OK workpiece result form the normal class, while operations under faulty scenarios form the faulty class. We use normal operations as references to calibrate operation-level anomaly scores, rather than testing for a distributional change within an operation.

We align the four stages to obtain 1400×3 trajectories. In each of 100 repetitions, 200 auxiliary normal operations are used for robust pointwise and channelwise standardization and network training; a disjoint set of $n_0 = 200$ normal operations forms $\mathcal{D}_0$; and the test set contains 50 normal and 50 faulty operations. The faulty training sample $\mathcal{D}_1(r)$ follows the sample-size rule used in the HASC experiment, so r controls the amount of labelled faulty training data. We train a residual convolutional network using the auxiliary normal operations and $\mathcal{D}_1(r)$, with the architecture used for HASC. DNN uses the network's fitted sigmoid probabilities for classification. ReCaP-DNN freezes its feature extractor and, for each test operation, refits the final classification layer to distinguish that operation and the references from $\mathcal{D}_1(r)$. It then calibrates the fitted probabilities using equation (1). Appendix B.3 gives the preprocessing, sampling and training details.

Table 3 shows that DNN's average FPR increases with r, from 0.015 to 1.000. ReCaP-DNN has average FPRs between 0.050 and 0.052 across the values of r considered, while its average TPR increases with r.

Table 3: Average false positive rate (FPR), true positive rate (TPR), and misclassification error rate (MER) over 100 repetitions of the S04 analysis.

Measure	$r = 0.1$		$r = 0.3$		$r = 0.5$		$r = 0.7$		$r = 0.9$	
	DNN	ReCaP-DNN	DNN	ReCaP-DNN	DNN	ReCaP-DNN	DNN	ReCaP-DNN	DNN	ReCaP-DNN
FPR	0.015	0.050	0.347	0.051	0.952	0.052	0.968	0.050	1.000	0.051
TPR	0.032	0.197	0.413	0.289	0.959	0.322	0.984	0.327	1.000	0.414
MER	0.491	0.426	0.467	0.381	0.496	0.365	0.492	0.362	0.500	0.319

4 Discussion and Limitations

ReCaP uses historical no-change sequences to calibrate classical and learned detection scores without requiring an analytic null distribution. Its power depends on how well the detection score distinguishes the changes of interest.

First, the block-based cutoff adjustment is heuristic. Further work is needed to establish conditions for its validity and characterize its power. Second, extending ReCaP to estimate change-point locations and quantify uncertainty in those locations is a natural next step. Third, online monitoring is important for timely change detection. Extending reference calibration to this setting would require error control across repeated tests.

AI use statement

OpenAI ChatGPT was used to assist with manuscript organization, language editing, and LaTeX formatting. AI-assisted suggestions were reviewed by the authors. The authors take responsibility for the final content of this work, including its text, mathematical claims, proofs, citations, code, and reported empirical results.

Ethics statement

This work develops statistical methodology and evaluates it using simulated data and two publicly available datasets. No new human participants were recruited and no new personal data were collected. The HASC data contain wearable-sensor recordings from human participants; we use only the released acceleration signals and activity labels, make no attempt to re-identify participants, and report aggregate results. The screw-driving data contain industrial process measurements rather than personal data. Deployment of the proposed method in safety-critical monitoring settings would require application-specific validation and appropriate human oversight.

Reproducibility statement

The theoretical assumptions and proofs are presented in Section 2 and Appendix A. The simulation models, comparison statistics, and settings for the multiscale and heterogeneity experiments are

described in Section 3 and Appendix C. These descriptions include sample sizes, parameter settings, multiscale interval sampling, and numbers of Monte Carlo replications. The real-data analyses use publicly available datasets. Section 3.3 and Appendix B describe data construction, preprocessing, sampling, model fitting, and evaluation, as well as the CLR and neural-network implementations.

REFERENCES

Andreas Anastasiou and Ivor Cribben. Ivor Cribben and Anastasiou Andreas's contribution to the Discussion of 'the Discussion Meeting on probabilistic and statistical aspects of machine learning'. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(2):313–314, 2024. doi: 10.1093/jrsssb/qkad149.

Sylvain Arlot, Alain Celisse, and Zaid Harchaoui. A kernel multiple change-point algorithm via model selection. *Journal of Machine Learning Research*, 20(162):1–56, 2019.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023. doi: 10.1214/23-AOS2276.

Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for outliers with conformal p-values. *The Annals of Statistics*, 51(1):149–178, 2023. doi: 10.1214/22-AOS2244.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.

Maxime Cauchois, Suyash Gupta, Alnur Ali, and John C. Duchi. Robust validation: Confident predictions even when distributions shift. *Journal of the American Statistical Association*, 119 (548):3033–3044, 2024. doi: 10.1080/01621459.2023.2298037.

Hao Chen and Nancy R. Zhang. Graph-based change-point detection. *The Annals of Statistics*, 43 (1):139–176, 2015. doi: 10.1214/14-AOS1269.

Derek Y. Chiang, Gad Getz, David B. Jaffe, Michael J. T. O'Kelly, Xiaojun Zhao, Scott L. Carter, Carsten Russ, Chad Nusbaum, Matthew Meyerson, and Eric S. Lander. High-resolution mapping of copy-number alterations with massively parallel sequencing. *Nature Methods*, 6:99–103, 2009. doi: 10.1038/nmeth.1276.

Haeran Cho and Piotr Fryzlewicz. Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 77(2):475–507, 2015. doi: 10.1111/rssb.12079.

Xiaolong Cui, Haoyu Geng, Guanghui Wang, Zhaojun Wang, and Changliang Zou. ART: distribution-free and model-agnostic changepoint detection with finite-sample guarantees. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 88(4):1278–1302, 2026. doi: 10.1093/jrsssb/qkag037.

Jonas Dehning, Johannes Zierenberg, F Paul Spitzner, Michael Wibral, Joao Pinheiro Neto, Michael Wilczek, and Viola Priesemann. Inferring change points in the spread of COVID-19 reveals the effectiveness of interventions. *Science*, 369(6500):eabb9789, 2020. doi: 10.1126/science.abb9789.

Moritz Jirak. Uniform change point tests in high dimension. *The Annals of Statistics*, 43(6):2451–2483, 2015. doi: 10.1214/15-AOS1347.

Rohit Kanrar, Feiyu Jiang, and Zhanrui Cai. Model-free change-point detection using AUC of a classifier. *Journal of Machine Learning Research*, 26(190):1–50, 2025.

Nobuo Kawaguchi, Ying Yang, Tianhui Yang, Nobuhiro Ogawa, Yohei Iwasaki, Katsuhiko Kaji, Tsutomu Terada, Kazuya Murao, Sozo Inoue, Yoshihiro Kawahara, Yasuyuki Sumi, and Nobuhiko Nishio. HASC2011corpus: Towards the common ground of human activity recognition. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, pp. 571–572. Association for Computing Machinery, 2011. doi: 10.1145/2030112.2030218.

Jie Li, Paul Fearnhead, Piotr Fryzlewicz, and Tengyao Wang. Automatic change-point detection in time series via deep learning. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(2):273–285, 2024. doi: 10.1093/jrsssb/qkae004.

Yunchen Li, Guanghui Wang, Shuntuo Xu, and Zhou Yu. Integral probability metric-guided CUSUM-Net for nonparametric changepoint detection. *Biometrika*, 113(3):asag046, 2026. doi: 10.1093/biomet/asag046.

Bin Liu, Cheng Zhou, Xinsheng Zhang, and Yufeng Liu. A unified data-adaptive framework for high dimensional change point detection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(4):933–963, 2020. doi: 10.1111/rssb.12375.

Malte Londschien, Peter Bühlmann, and Solt Kovács. Random forests for change point detection. *Journal of Machine Learning Research*, 24(216):1–45, 2023.

Ariane Marandon, Lihua Lei, David Mary, and Etienne Roquain. Adaptive novelty detection with false discovery rate guarantee. *The Annals of Statistics*, 52(1):157–183, 2024. doi: 10.1214/23-AOS2338.

Euan T. McGonigle and Haeran Cho. Nonparametric data segmentation in multivariate time series via joint characteristic functions. *Biometrika*, 112(2):asaf024, 2025. doi: 10.1093/biomet/asaf024.

Tejaswini Mishra, Meng Wang, Ahmed A. Metwally, Gireesh K. Bogu, Andrew W. Brooks, Amir Bahmani, Arash Alavi, Alessandra Celli, Emily Higgs, Orit Dagan-Rosenfeld, Bethany Fay, Susan Kirkpatrick, Ryan Kellogg, Michelle Gibson, Tao Wang, Erika M. Hunting, Petra Mamic, Ariel B. Ganz, Benjamin Rolnik, Xiao Li, and Michael P. Snyder. Pre-symptomatic detection of COVID-19 from smartwatch data. *Nature Biomedical Engineering*, 4:1208–1220, 2020. doi: 10.1038/s41551-020-00640-6.

Paul Nomikos and John F. MacGregor. Multivariate SPC charts for monitoring batch processes. *Technometrics*, 37(1):41–59, 1995. doi: 10.1080/00401706.1995.10485888.

Aaron A. Sandel, Yixuan He, Junpeng Ren, Yik Lun Kei, Kevin C. Lee, Isabelle R. Clark, Rachna B. Reddy, Jacob D. Negrey, Charles Birungi, Blessing A. Apamaku, Diana Kanweri, Davis Kalunga, Christopher Aliganyira, Sebastián Ramírez-Amaya, Phionah Nakayima, Raymond Katumba, Brian Kamugyisha, Daniela Acosta-Florez, Bas van Boekholt, Godfrey Mbabazi, Erone Akamumpa, Sharifah Namaganda, Alfred Tumusiime, Samuel Angedakin, Gesine Reinert, Oscar Madrid-Padilla, Mihai Cucuringu, David Wipf, Kevin E. Langergraber, David P. Watts, and John C. Mitani. Lethal conflict after group fission in wild chimpanzees. *Science*, 392(6794):216–220, 2026. doi: 10.1126/science.adz4944.

Vladimir Vovk, Ilia Nouretdinov, and Alex Gammerman. Testing exchangeability on-line. In Tom Fawcett and Nina Mishra (eds.), *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 768–775. AAAI Press, 2003.

Guanghui Wang and Long Feng. Computationally efficient and data-adaptive changepoint inference in high dimension. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(3):936–958, 2023. doi: 10.1093/jrsssb/qkad048.

Tengyao Wang and Richard J. Samworth. High dimensional change point estimation via sparse projection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80(1):57–83, 2018. doi: 10.1111/rssb.12243.

Yunlong Wang, Changliang Zou, Zhaojun Wang, and Guosheng Yin. Multiple change-points detection in high dimension. *Random Matrices: Theory and Applications*, 8(4):1950014, 2019. doi: 10.1142/S201032631950014X.

Nikolai West and Jochen Deuse. Industrial screw driving dataset collection: Time series data for process monitoring and anomaly detection. Zenodo, 2025. Version v1.2.2, doi: 10.5281/zenodo.15393134.

Haotian Xu, Daren Wang, Zifeng Zhao, and Yi Yu. Change-point inference in high-dimensional regression models under temporal dependence. *The Annals of Statistics*, 52(3):999–1026, 2024. doi: 10.1214/24-AOS2380.

Nancy R. Zhang, David O. Siegmund, Hanlee Ji, and Jun Z. Li. Detecting simultaneous change-points in multiple sequences. *Biometrika*, 97(3):631–645, 2010. doi: 10.1093/biomet/asq025.

Yangfan Zhang, Runmin Wang, and Xiaofeng Shao. Adaptive inference for change points in high-dimensional data. *Journal of the American Statistical Association*, 117(540):1751–1762, 2022. doi: 10.1080/01621459.2021.1884562.

Changliang Zou, Guosheng Yin, Long Feng, and Zhaojun Wang. Nonparametric maximum likelihood approach to multiple change-point problems. *The Annals of Statistics*, 42(3):970–1002, 2014. doi: 10.1214/14-AOS1210.

Appendix A gives the proofs and states the multiscale validity result. Appendix B describes the CLR score computation and the neural-network implementations for the HASC and screw-driving analyses. Appendix C provides simulation details and additional results.

A Theoretical results and proofs

A.1 Proof of Proposition 1

Proof. Conditional on $\mathcal{G}$, equation (1) is the standard randomized conformal p-value based on the exchangeable scores $(S_i : i \in I)$. Hence Vovk et al. (2003, Theorem 1) gives $p \mid \mathcal{G} \sim \mathrm{Uniform}(0,1)$. □

A.2 Validity of multiscale calibration

We use the common nonempty finite interval family $\mathcal{J} = \{I_1, \ldots, I_L\}$ and multiscale scores S_i defined in Section 2.4. The following result allows a deterministic interval family or one chosen independently of $\mathbf{Z}$ conditional on $\mathcal{G}$.

Proposition A1 (Validity with multiple intervals)**.** *Suppose that, under $H_{0,n+1}$, $\mathbf{Z}$ is exchangeable conditional on $\mathcal{G}$, and that $\mathcal{J}$ is independent of $\mathbf{Z}$ conditional on $\mathcal{G}$. Given $\mathcal{G}$ and $\mathcal{J}$, suppose that the full procedure producing the real-valued score array $(S_{\ell,i} : 1 \le \ell \le L,\ i \in I)$ is permutation-equivariant under reorderings of $\mathbf{Z}$. For a randomized procedure, this condition applies to the joint conditional distribution of the array given $\mathbf{Z}$, $\mathcal{G}$ and $\mathcal{J}$. Then $(S_i : i \in I)$ is exchangeable conditional on $\mathcal{G}$, and the p-value in equation (1), computed from these multiscale scores, is uniform on $[0,1]$ conditional on $\mathcal{G}$.*

Proof. Conditional on $\mathcal{G}$ and $\mathcal{J}$, the pooled sequences remain exchangeable. Permutation equivariance of the joint scoring procedure therefore makes $(S_{1,i}, \ldots, S_{L,i})$, $i \in I$, exchangeable. Taking their maxima and then integrating over $\mathcal{J}$ gives exchangeability of $(S_i : i \in I)$ conditional on $\mathcal{G}$. Proposition 1 gives the conclusion. □

A.3 Proof of Theorem 1

Proof. Location–scale invariance gives $S_i = S(Z_i; \xi_i) = S(\varepsilon_i; \xi_i)$ for every $i \in I$. Applying the common map $(\varepsilon, \xi) \mapsto S(\varepsilon; \xi)$ to the conditionally exchangeable pairs $((\varepsilon_i, \xi_i) : i \in I)$ yields conditionally exchangeable scores. Proposition 1 gives the result. □

A.4 Proof of Corollary 1

Proof. Fix the scoring randomization ξ. The random forest of Londschien et al. (2023, Section 3.3) is invariant to positive rescaling of each feature. A coordinate shift also preserves every split membership because it preserves feature order and translates each midpoint threshold by the same amount. The fitted class probabilities are therefore unchanged by any positive coordinatewise affine transformation. The CLR gains are unchanged for every pilot split, as are the fixed-pilot score and the final multiple-pilot score. Thus the score is location–scale invariant, and Theorem 1 gives the result. □

A.5 Proof of Theorem 2

Proof. For fixed α, $t \mapsto d_{\mathrm{KL}}(\alpha, t)$ is continuous and strictly decreasing from infinity to zero on $(0, \alpha]$, because its derivative is $(t-\alpha)/\{t(1-t)\}$. Thus α_ρ exists and is unique. Set $t = \alpha_\rho$ and $q = F_{\mathcal{G}}(t)$. If $q \le t$, then $q \le \alpha$. If $q > t$, the assumed bound gives $d_{\mathrm{KL}}(q, t) \le \rho = d_{\mathrm{KL}}(\alpha, t)$. Since $q \mapsto d_{\mathrm{KL}}(q, t)$ is strictly increasing on $(t, 1]$, again $q \le \alpha$. Therefore $\Pr(p \le \alpha_\rho \mid \mathcal{G}) \le \alpha$. Finally, with $\delta = \alpha - \alpha_\rho$, the expansion $d_{\mathrm{KL}}(\alpha, \alpha - \delta) = \delta^2/\{2\alpha(1-\alpha)\} + O(\delta^3)$ gives the stated small-ρ expansion. □

B Implementation details

B.1 CLR scores with multiple pilot splits

For sequence Z_i, let $G_i(t;\tau)$ denote the CLR gain at candidate split t when the classifier is fitted at pilot split τ, as in Example 1. Following Londschien et al. (2023), we first use the three pilot splits $\mathcal{P} = \{\lfloor T/4 \rfloor, \lfloor T/2 \rfloor, \lfloor 3T/4 \rfloor\}$. For the candidate-split set $\mathcal{T} \subset \{1, \dots, T-1\}$, choose the refinement split

$$\tau_i^\star \in \arg\max_{t \in \mathcal{T}} \max_{\tau \in \mathcal{P}} G_i(t;\tau).$$

We then fit a random forest using $\tau_i^\star$ as the pilot split. Let $\widehat{p}^{\star}_{i,k}(Z_{i,u})$ denote its out-of-bag probability for class $k \in \{1,2\}$, and set $\pi_{i,u} = \{\tau_i^\star - \mathbf{1}(u \le \tau_i^\star)\}/(T-1), \eta = \exp(-6)$. The final gain curve is

$$G_i^\star(t) = \sum_{u \le t} \log\left\{(1-\eta)\frac{\widehat{p}^{\star}_{i,1}(Z_{i,u})}{\pi_{i,u}} + \eta\right\} + \sum_{u > t} \log\left\{(1-\eta)\frac{\widehat{p}^{\star}_{i,2}(Z_{i,u})}{1-\pi_{i,u}} + \eta\right\}, \quad t \in \mathcal{T},$$

and the multiple-pilot CLR score used in the main text is

$$S_i = \max_{t \in \mathcal{T}} G_i^\star(t).$$

B.2 HASC neural-network implementation

This subsection gives implementation details for the neural-network methods in the HASC analysis (Section 3.3.1). After the channelwise standardization described there, each 700×6 array is transposed to form a 6×700 network input.

We adapt the deep residual convolutional neural network architecture employed by Li et al. (2024) for the HASC data to binary classification. Each residual block comprises two convolutional layers, both followed by batch normalization and ReLU activation. The stack of residual blocks is followed by global average pooling and fully connected layers with 50, 40, and 30 units. The convolutional layers use 8 filters with kernel size $(3, 25)$, and the dropout rate is 0.3. The network is pretrained on the labelled sample described in Section 3.3.1 using unweighted sparse categorical cross-entropy and Adam with an initial learning rate of 10^{-3} and inverse-time decay. The batch size is 32, 20% of the pretraining sample is held out for validation, and training is run for 30 epochs. We remove the output layer and freeze the feature extractor; its 30-unit layer supplies the representation $e(Z) \in \mathbb{R}^{30}$.

For each final-layer fit specified in Section 3.3.1, the representation coordinates are standardized. We use ℓ_2-penalized logistic regression with $C = 1$, the `lbfgs` solver, and 1000 iterations. For ReCaP-DNN, conditional on the frozen feature extractor and $\mathcal{D}_1(r)$, this fitting procedure treats the pooled reference sequences and test sequence symmetrically. The resulting score construction is permutation-equivariant under reorderings of this pooled sample, as required in Section 2.2.

B.3 Screw-driving implementation

This subsection supplements the S04 application in Section 3.3.2. The Industrial Screw Driving Dataset Collection is publicly available from Zenodo (West & Deuse, 2025) (`https://doi.org/10.5281/zenodo.15393134`).

Record selection. We retain operations containing the four tightening steps Finding, Thread forming, Pre-tightening and Final tightening. Normal operations have `scenario_condition` set to `normal` and `workpiece_result` set to `OK`. Faulty operations have `scenario_condition` set to `faulty`.

Preprocessing and sampling. The four stages are linearly resampled to 100, 650, 250 and 400 points, respectively, giving aligned 1400×3 trajectories. The auxiliary, reference and test sample sizes are given in Section 3.3.2. For each of the 100 repetitions and each $r \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, operations are sampled without replacement, with workpiece-level separation enforced across samples. No workpiece contributes operations to more than one of the auxiliary normal training, faulty training, reference, and test samples. The auxiliary normal sample $\mathcal{D}_{0,\text{aux}}$ is used to robustly standardize

the trajectories at each time point and channel. The resulting standardized trajectory $R_i \in \mathbb{R}^{1400\times 3}$ is used as the neural-network input, and the same standardization is applied to all remaining operations. The faulty training sample $\mathcal{D}_1(r)$ has size $\mathrm{round}\{(n_0+1)r/(1-r)\}$, as in Section 3.3.1. Thus r controls the number of faulty training operations; it is not their exact proportion in the network's training sample.

DNN and ReCaP-DNN. DNN and ReCaP-DNN classify whole operations as normal or faulty. We use the residual convolutional architecture in Appendix B.2. For each r, the network is trained on $\mathcal{D}_{0,\mathrm{aux}}$ and $\mathcal{D}_1(r)$, with label 0 assigned to normal operations and label 1 to faulty operations. We train for up to 20 epochs, with 20% of the training sample held out for validation; training stops if the validation loss does not improve for three consecutive epochs. DNN uses the fitted sigmoid probabilities from this network for classification. ReCaP-DNN freezes the feature extractor and uses its 30-dimensional representation. For each test operation, it fits a logistic final layer to distinguish that operation and the references from $\mathcal{D}_1(r)$, using the fitting settings in Appendix B.2, and then applies equation (1) to the fitted probabilities.

The results and their interpretation are reported in Section 3.3.2 and Table 3.

C Simulation details and additional results

C.1 Gaussian mean changes

C.1.1 Comparison statistics

For $Z = (Z_1, \ldots, Z_T)$, with $Z_t = (Z_{t1}, \ldots, Z_{td})^\mathsf{T}$, define the standardized CUSUM statistic

$$C_j(t) = \left\{ \frac{t}{T}\left(1-\frac{t}{T}\right)\right\}^{-1/2} \frac{\sum_{u=1}^{t} Z_{uj} - (t/T)\sum_{u=1}^{T} Z_{uj}}{\sqrt{T}\,\widehat{\sigma}_j},$$

where $\widehat{\sigma}_j^2 = \{2(T-1)\}^{-1}\sum_{u=2}^{T}(Z_{uj}-Z_{u-1,j})^2$. With $c = 0.2$, we use

$$M_{T,d} = \max_{\lfloor cT\rfloor \le t \le T-\lfloor cT\rfloor} \max_{1\le j\le d} |C_j(t)|, \qquad S_{T,d} = \sum_{t=1}^{T-1}\sum_{j=1}^{d} C_j^2(t).$$

For Max, Wang & Feng (2023) show that, after normalization, $M_{T,d}$ has a standard Gumbel limit under the null. In particular, writing $G(x) = \exp\{-\exp(-x)\}$ for the standard Gumbel distribution function, the corresponding asymptotic p-value is

$$p_{\mathrm{Max}} = 1 - G\left\{A(d\log h_T)M_{T,d} - D(d\log h_T)\right\},$$

where $A(x) = (2\log x)^{1/2}$ and $D(x) = 2\log x + \frac{1}{2}\log\log x - \frac{1}{2}\log\pi$; h_T is determined by the boundary parameter $\lfloor cT\rfloor$ as in Wang & Feng (2023).

For Sum, Wang et al. (2019) establish

$$\frac{S_{T,d} - (T+2)d}{V_{T,d}^{1/2}} \implies N(0,1),$$

where $V_{T,d}$ is the variance estimator used in their procedure. We therefore compute

$$p_{\mathrm{Sum}} = 1 - \Phi\left\{\frac{S_{T,d}-(T+2)d}{V_{T,d}^{1/2}}\right\},$$

with Φ denoting the standard normal distribution function.

Finally, Wang & Feng (2023) show that the Max and Sum statistics are asymptotically independent under the null. Their Fisher combination, denoted Comb, therefore uses

$$T_{\mathrm{Comb}} = -2\{\log(p_{\mathrm{Max}}) + \log(p_{\mathrm{Sum}})\}, \qquad T_{\mathrm{Comb}} \implies \chi^2_4,$$

and $p_{\mathrm{Comb}} = 1 - F_{\chi^2_4}(T_{\mathrm{Comb}})$.

C.1.2 Additional results

Table A1 reports the empirical sizes from the independent Gaussian experiment in Section 3.1.

Table A1: Empirical sizes under the independent Gaussian null, based on 1,000 Monte Carlo replications.

Method	Max	Sum	Comb	ReCaP-Max	ReCaP-Sum	ReCaP-Comb
Empirical size	0.046	0.058	0.068	0.044	0.057	0.039

Table A2 reports rejection probabilities under the Gaussian AR(1) alternatives in Section 3.1. Because Max, Sum and Comb have empirical sizes above the nominal level in this setting, these results do not provide size-matched power comparisons.

Table A2: Empirical rejection probabilities under Gaussian AR(1) mean-change alternatives, based on 1,000 Monte Carlo replications.

Δ	k	Max	Sum	Comb	ReCaP-Max	ReCaP-Sum	ReCaP-Comb
1	1	0.997	1.000	1.000	0.975	0.350	0.972
1	2	0.991	1.000	1.000	0.955	0.355	0.946
1	3	0.971	1.000	1.000	0.743	0.356	0.751
1	4	0.924	1.000	1.000	0.575	0.380	0.610
1	5	0.863	1.000	1.000	0.413	0.357	0.439
2	10	0.972	1.000	1.000	0.660	0.809	0.863
2	20	0.902	1.000	1.000	0.348	0.780	0.798
2	30	0.835	1.000	1.000	0.281	0.775	0.791
2	40	0.803	1.000	1.000	0.246	0.768	0.782
2	50	0.794	1.000	1.000	0.236	0.784	0.787

C.2 Experimental design for epidemic mean changes

We evaluate the multiscale procedures under epidemic mean changes in Gaussian sequences. We set $T = 200$, $p = 50$, $n_0 = 200$, and $\alpha = 0.05$, and observations are independent over time. Under the null, $Z_t \sim N_p(0, I_p)$ for all t. Under the alternatives,

$$Z_t \sim \begin{cases} N_p(0, I_p), & t \le aT, \\ N_p(\delta, I_p), & aT < t \le bT, \\ N_p(0, I_p), & t > bT. \end{cases}$$

Thus the mean changes at aT and returns to zero at bT. We use four settings:

$$\begin{aligned} &\text{Dense, long:} && (a,b) = (0.30, 0.70), && \delta = 0.18\,\mathbf{1}_p, \\ &\text{Sparse, long:} && (a,b) = (0.30, 0.70), && \delta = 0.75\,(e_1 + e_2), \\ &\text{Dense, short:} && (a,b) = (0.42, 0.58), && \delta = 0.23\,\mathbf{1}_p, \\ &\text{Sparse, short:} && (a,b) = (0.42, 0.58), && \delta = 1.05\,(e_1 + e_2), \end{aligned}$$

where $\mathbf{1}_p$ is the vector of ones and e_j is the jth standard basis vector. The long- and short-interval settings use different shift magnitudes, so their comparison does not isolate the effect of interval length.

We compare ReCaP-Sum, ReCaP-Max and ReCaP-Comb with their multiscale versions. For each reference sample, we generate a common family of 40 intervals before sampling the sequences: the full interval $(0, T]$ and 39 random intervals whose endpoints are sampled uniformly from $\{0, \dots, T\}$, subject to a minimum length of 20. The same intervals are used for the reference and test sequences. Candidate split points leave at least 10% of each interval on either side. On each interval, the scores are standardized over the pooled reference and test sequences before the maximum is taken. For ReCaP-Comb, the maximum is also taken over $q \in \{2, \infty\}$. Each reported rejection probability is based on 2,000 Monte Carlo replications.

Table 1 reports the results, which are discussed in Section 3.2.

C.3 Numerical study of the heuristic cutoff

We evaluate the heuristic cutoff in Section 2.5 when the test and reference sequences have different distributions, using the $q = 2$ CUSUM score and the fixed-pilot CLR score.

C.3.1 Simulation settings

The reference observations are independent with $Z_{i,t} \sim N_p(0, I_p)$, where $n_0 = 200$, $T = 200$, $p = 10$, and $\alpha = 0.05$. We consider four null models for the test sequence:

$$\begin{aligned}
\text{Scale:} \quad & Z_t \sim N_p(0, D^2), \quad D = \operatorname{diag}(d_1, \ldots, d_p), \\
& d_j = 1 + \frac{0.1(j-1)}{p-1}; \\
\text{Serial dependence:} \quad & Z_1 \sim N_p(0, I_p), \\
& Z_t = 0.025 Z_{t-1} + \sqrt{1 - 0.025^2}\, \varepsilon_t, \quad \varepsilon_t \overset{\text{i.i.d.}}{\sim} N_p(0, I_p); \\
\text{Spatial dependence:} \quad & Z_t \sim N_p(0, \Sigma), \quad \Sigma_{jj} = 1, \quad \Sigma_{jk} = 0.5 \quad (j \neq k); \\
\text{Heavy tails:} \quad & Z_{t,j} \overset{\text{i.i.d.}}{\sim} t_3/\sqrt{3}.
\end{aligned}$$

In each null setting, the observations in the test sequence are identically distributed over time, but the joint distribution of the sequence differs from that of a reference sequence. Under the alternative, we add $\delta = 0.4(e_1 + e_2 + e_3)$ to the mean after $T/2$. Each reported rejection probability is based on 250 Monte Carlo replications.

We use the $q = 2$ CUSUM score and the fixed-pilot CLR score from Example 1. For CUSUM, the block construction in Section 2.5 uses $B = 100$ blocks of length $\ell = 30$. For CLR, we use $B = 50$ blocks of length $\ell = 100$. The pilot split is at the midpoint of the sequence being scored: $T/2$ for the full sequence and $\ell/2$ for each block. The random forest has 25 trees, maximum depth 6, and minimum leaf size 2, and the CLR gain is evaluated over 20 candidate split points. We do not include the scale setting for CLR because the CLR score computed with a random forest is invariant to positive coordinatewise rescaling by Corollary 1 and Remark 2.

C.3.2 Results

Table A3 compares rejection probabilities using the nominal cutoff with those using the heuristic adjustment in Section 2.5. For CUSUM, the adjusted empirical sizes range from 0.012 to 0.056, compared with 0.076 to 0.472 without adjustment. The reduction in size can entail substantial power loss: under spatial dependence, CUSUM power decreases from 0.852 to 0.036.

Table A3: Empirical rejection probabilities when reference and test sequences have different distributions, based on 250 Monte Carlo replications. "Adj." denotes the heuristic cutoff adjustment.

		Null		Alternative	
Score	Setting	Size	Adj. size	Power	Adj. power
CUSUM	Scale	0.076	0.024	0.864	0.552
	Serial dependence	0.080	0.052	0.840	0.676
	Spatial dependence	0.472	0.012	0.852	0.036
	Heavy tails	0.156	0.056	0.860	0.448
CLR	Serial dependence	0.080	0.052	0.628	0.268
	Spatial dependence	0.072	0.048	0.660	0.388
	Heavy tails	0.036	0.016	0.972	0.228

For CLR, the adjusted empirical sizes are close to the nominal level under serial and spatial dependence. In the heavy-tailed setting, the unadjusted empirical size is already below the nominal level and decreases further after adjustment. The adjustment reduces empirical power in all three settings. These empirical findings do not establish validity of the heuristic cutoff.