A Fast and Efficient Change-point Detection Framework based on Approximate k-Nearest Neighbor Graphs

Yi-Wei Liu, *UC Davis,* and Hao Chen, *UC Davis,*

***Abstract*—Change-point analysis is thriving in this big data era to address problems arising in many fields where massive data sequences are collected to study complicated phenomena over time. It plays an important role in processing these data by segmenting a long sequence into homogeneous parts for follow-up studies. The task requires the method to be able to process large datasets quickly and deal with various types of changes for high-dimensional data. We propose a new approach making use of approximate k-nearest neighbor information from the observations, and derive an analytic formula to control the type I error. The time complexity of our proposed method is $O\left(dn(\log n + k\log d) + nk^2\right)$ for an n-length sequence of d-dimensional data. The test statistic we consider incorporates a useful pattern for moderate- to high- dimensional data so that the proposed method could detect various types of changes in the sequence. The new approach is also asymptotic distribution free, facilitating its usage for a broader community. We apply our method to fMRI datasets and Neuropixels datasets to illustrate its effectiveness.**

***Index Terms*—Change-point analysis; Graph-based edge-count statistic; Directed k-nearest neighbor graph.**

I. Introduction

With advances in technologies, scientists in many fields are collecting massive data for studying complex phenomena over time and/or space. Such data often involve sequences of high-dimensional measurements that cannot be analyzed through traditional approaches. Insights on such data often come from segmentation/change-point analysis, which divides the sequence into homogeneous temporal or spatial segments. They are crucial early steps in understanding the data and in detecting anomalous events. Change-point analysis has been extensively studied for univariate and low-dimensional data (see [1]–[5] for various aspects of classic change-point analysis). However, many modern applications require effective and fast change-point detection for high-dimensional data. For example, Neuropixels recordings [6], microarrays [7], healthcare data [8], etc.

Let the sequence of observations be $\{\mathbf{y}_t : t = 1, \ldots, n\}$, indexed by some meaningful order, such as time or location. Then the change-point detection problem can be formulated as testing the null hypothesis of homogeneity:

$$H_0 : \mathbf{y}_t \sim F_0,\, t = 1, \ldots, n, \tag{1}$$

against the alternative that there exists a change-point τ:

$$H_1 : \exists 1 \le \tau < n,\, \mathbf{y}_t \sim \begin{cases} F_0, & t \le \tau \\ F_1, & \text{otherwise} \end{cases} \tag{2}$$

Here, F_0 and F_1 are two different probability measures. When there are multiple change-points, wild binary segmentation [9], [10] or seeded binary segmentation [11] can be incorporated.

Recently, there were quite a number of progresses on parametric change-point detection. For example, [12] used the piecewise stationary time series factor models for multivariate observations, and assumed the changes are in their second-order structure. Reference [13] studied the change-point detection and localization problem in dynamic networks by assuming the entries of the adjacency matrices are from inhomogeneous Bernoulli models. Reference [14] considered the change-point detection problem in networks generated by a dynamic stochastic block model mechanism. Reference [15] addressed the change-point problem with missing values, and focused on detecting the covariance structure breaks in Gaussian graphical models. These methods work under certain parametric models. However, in many applications, we have little knowledge on F_0 and F_1.

In the context of nonparametric change-point detection for high-dimensional data, kernel-based methods were first explored [16], [17], and continued to be improved [18]. However, this kernel approach is difficult to use practically. As we will show in Section IV, this method is very sensitive to the choice of a tuning parameter and it is time-consuming to apply the method with a proper type I error control, where type I error is the event that a change-point is falsely detected when the sequence is actually homogeneous. Reference [19] proposed the scan B-statistic for kernel change-point detection, which is computationally efficient and has a fast formula for type I error control; however, it requires a large amount of reference data. In recent years, distance-based methods [20] and graph-based methods [21], [22] were proposed for high-dimensional change-point detection. The distance-based method (ecp) uses all pairwise distances among observations to find change-points, which could also be computationally heavy for large datasets because computing all pairwise distances needs $O(dn^2)$ time for d-dimensional data. In addition, there

This work was supported in part by the NSF Awards DMS-1513653 and DMS-1848579. *(Corresponding author: Hao Chen.)* The authors are with the Department of Statistics, University of California, Davis, Davis, CA, 95616 (e-mail: lywliu@ucdavis.edu; hxchen@ucdavis.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2022.3162120, provided by the authors.

is no fast analytic formula for type I error control, and thus one needs to draw random permutations to approximate the p-value. The graph-based methods [21], [22] utilize the information of a similarity graph constructed on observations to detect change-points. The authors also provided analytic formulas for type I error control, making them faster to run. In addition, the graph-based methods can detect more types of changes compared to ecp. The ecp method is very sensitive to changes in mean but its performance decays when the changes come in many other forms (see Section IV-E).

In this paper, we seek further improvement on graph-based methods, especially from an efficiency perspective. Existing graph-based methods for offline change-point detection utilize an undirected graph constructed among observations. Some common choices are the minimum spanning tree (MST), where all observations are connected with the total distance minimized; the minimum distance pairing (MDP), where the observations are partitioned into $n/2$ pairs with the total within-pair distance minimized; the undirected nearest neighbor (NN) graph, where each observation connects to its nearest neighbor; and their denser versions, k-MST, k-MDP, and undirected k-NN graphs. Take the k-MST for example, it is the union of the 1st, ..., kth MSTs, where the 1st MST is the MST, and the jth MST is a spanning tree connecting all observations such that the sum of the edges in the tree is minimized under the constraint that it does not contain any edge in the 1st, ..., $(j-1)$th MSTs. Among these graphs, k-MST is preferred as it in general has a higher power than others [21]. Nevertheless, it requires $O(dn^2)$ time to compute the distance matrix among n d-dimensional observations, so it takes at least $O(dn^2)$ time to construct the k-MST from the original data when the pairwise distances were not provided in the beginning, which is usually the case. This could be inefficient when either n or d is large.

Hence, we seek other ways to construct the similarity graph. There are fast existing algorithms to construct the directed approximate k-NN graph [23], where each observation finds k other nearby points that might not be the k closest ones. We use the kd-tree algorithm to search for approximate nearest neighbors. A kd-tree is a space-partitioning data structure for organizing points in a high-dimensional space, which is a binary tree constructed through splitting the points by the values on alternating coordiantes as the tree grows. It takes $O(dn \log n)$ time to preprocess a set of n points in $\mathbb{R}^d$ [24]. The nearest neighbor for any given query point can be searched efficiently with the kd-tree. To approximate the nearest neighbors, first traverse the tree to the leaf node that contains the query point, and then search for the nearest neighbors only in nearby areas. It requires only $O(d \log d + \log n)$ time to result in a good approximate nearest neighbor per query [25], so the total computational cost for obtaining a directed approximate k-NN graph with the kd-tree can be achieved at $O\left(dn(\log n + k \log d)\right)$. Simulation studies show that this new approach has power on par with the existing method on k-MST (Section IV-E), and the new approach is much faster (Section IV-B).

Since the existing offline graph-based change-point detection framework needs the graph to be an undirected graph, we further work out a framework that can deal with the directed approximate k-NN graph, i.e., all the following steps after the graph is constructed: the exact analytic formulas to compute the test statistic, the limiting distribution of the new statistic, and the analytic formula to supervise the false discovery rate efficiently. The time complexity of the method after the directed approximate k-NN graph is obtained is $O(nk^2)$. Thus, the overall time complexity of the new method is $O\left(dn(\log n + k \log d) + nk^2\right)$. We illustrate the new approach on the analyses of fMRI datasets and Neuropixels datasets (Section V). The former ones have very large dimensions and moderate sample sizes, whereas the latter ones feature very large sample sizes with moderate dimensions.

II. Proposed Statistic

Let $G = \{(i,j) : \mathbf{y}_j \text{ is among } \mathbf{y}_i\text{'s } k \text{ approximate nearest neighbors}\}$ be the directed approximate k-NN graph, $R_{G,1}(t)$ be the number of edges on G connecting observations both before t, and $R_{G,2}(t)$ be the number of edges on G connecting observations both after t:

$$R_{G,1}(t) = \sum_{(i,j)\in G} \mathbb{1}_{\{i\le t, j\le t\}},\ R_{G,2}(t) = \sum_{(i,j)\in G} \mathbb{1}_{\{i>t, j>t\}};$$

with $\mathbb{1}_A$ being the indicator function for event A. Here, we use the notations similar to those in [22]. A key difference is that the graph in [22] is undirected, whereas here G is directed. Fig. 1 illustrates the computation of $R_{G,1}(t)$ and $R_{G,2}(t)$ on a toy example. We will use these two quantities to construct the test statistics. The rationale is as follows: When all observations are from the same distribution, the distributions of $R_{G,1}(t)$ and $R_{G,2}(t)$ can be figured out under the permutation null distribution that places $1/n!$ probability on each of the $n!$ permutations of $\{\mathbf{y}_i : i = 1, \ldots, n\}$. With no further specification, we use pr, E, var, and cov to denote probability, expectation, variance, and covariance, respectively, under the permutation null distribution. When there is a change-point at τ, one typical outcome is that observations from the same distribution tend to form edges within themselves, making both $R_{G,1}(\tau)$ and $R_{G,2}(\tau)$ larger than their null expectations. Another common but somewhat counter-intuitive outcome is that observations from one distribution tend to connect within themselves, but observations from the other distribution tend not to connect within themselves, casuing one of $R_{G,1}(\tau)$ and $R_{G,2}(\tau)$ to be larger than its null expectation, and the other smaller than its null expectation. This happens commonly under moderate to high dimensions when the variances of the two distributions differ. The underlying reason is the curse of dimensionality (see [26] for detailed explanations on this phenomenon under the two-sample testing setting).

To cover both possible outcomes under the alternative, we focus on a max-type test statistic in the main context. Three other test statistics (original/weighted/generalized) are discussed in Supplement D.

For each candidate t of the true change-point τ, the max-type edge-count statistic is defined as

$$M(t) = \max(Z_w(t), |Z_{\text{diff}}(t)|); \tag{3}$$

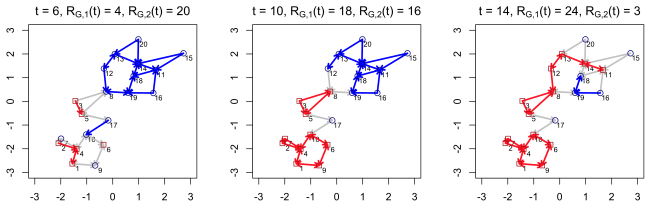


Fig. 1: The computation of $R_{G,1}(t)$ and $R_{G,2}(t)$ at three different values of t. Here $\mathbf{y}_1, \ldots, \mathbf{y}_{10} \overset{\text{i.i.d.}}{\sim} N((-0.5,-0.5)^T, \mathbb{I}_2)$, and $\mathbf{y}_{11}, \ldots, \mathbf{y}_{20} \overset{\text{i.i.d.}}{\sim} N((0.5,0.5)^T, \mathbb{I}_2)$, where $\mathbb{I}_2$ is the 2×2 identity matrix. The graph G here is the directed 2-NN on the Euclidean distance. Each t divides the observations into two groups: one group for observations before t (red squares) and the other group for observations after t (blue circles). Red edges connect observations before t and the number of red edges is $R_{G,1}(t)$; blue edges connect observations after t and the number of blue edges is $R_{G,2}(t)$. Notice that as t changes, the group identities change but the graph G does not change.

where

$$\begin{aligned} Z_w(t) &= \frac{R_w(t) - \mathsf{E}(R_w(t))}{\sqrt{\mathsf{var}(R_w(t))}}, \\ Z_{\text{diff}}(t) &= \frac{R_{\text{diff}}(t) - \mathsf{E}(R_{\text{diff}}(t))}{\sqrt{\mathsf{var}(R_{\text{diff}}(t))}}; \end{aligned}$$

with

$$\begin{aligned} R_w(t) &= \frac{n-t-1}{n-2} R_{G,1}(t) + \frac{t-1}{n-2} R_{G,2}(t), \\ R_{\text{diff}}(t) &= R_{G,1}(t) - R_{G,2}(t). \end{aligned}$$

The null hypothesis of homogeneity (1) is rejected if the test statistic

$$\max_{n_0 \le t \le n_1} M(t); \tag{4}$$

with n_0 and n_1 pre-specified, is larger than the critical value for a given significance level, which measures the strength of the evidence that must be presented in the sample to reject the null hypothesis, and has to be determined before conducting the experiment. Statistically, significance level is the probability of rejecting the null hypothesis when it is true, usually set to be 5% or 1%.

Here, the two components, $Z_w(t)$ and $|Z_{\text{diff}}(t)|$, capture the aforementioned two possible outcomes under the alternative. For better understanding, we illustrate the outcomes through a toy example (Fig. 2). When $\{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$ and $\{\mathbf{y}_{t+1}, \ldots, \mathbf{y}_n\}$ are from the same distribution, they are well mixed and $R_{G,1}(t)$ and $R_{G,2}(t)$ would be close to their null expectations (Fig. 2 (a)). When they are from different distributions, one common exhibition is that observations from the same distribution are more likely to be connected in G. Fig. 2 (b) plots a typical directed 2-NN graph under this alternative and we see that there are more edges connecting within each group. When this happens, $Z_w(t)$ is large. For moderate- to high- dimensional data, another exhibition of the graph is common under the alternative shown in Fig. 2 (c). Here, the dimension is $d = 100$, and the blue circles are from a distribution with a larger variance than that of the red squares. We see that $R_{G,1}(t)$ is much larger than its null expectation but $R_{G,2}(t)$ is much smaller than its null expectation (very few blue edges). This happens due to the curse of dimensionality. As the volume of a d-dimensional ball increases exponentially in d, the blue circles from a distribution with a larger variance are sparsely scattered and tend to find their nearest neighbors in red squares. The $Z_{\text{diff}}(t)$ part in our statistic is effective in capturing this pattern. The absolute value is to cover the two possible scenarios in opposite directions showcased in Fig. 2 (c) and (d).

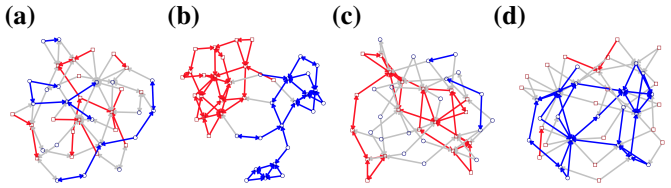


Fig. 2: Directed 2-NN graphs on 100-dimensional data visualized by the `ggnet2` function. Here, $\mathbf{y}_1, \ldots, \mathbf{y}_{20}$ (red squares) are randomly drawn from a 100-dimensional Gaussian distribution with zero mean and identity covariance matrix, and $\mathbf{y}_{21}, \ldots, \mathbf{y}_{40}$ (blue circles) are randomly drawn from $N_{100}(\boldsymbol{\mu}, a\mathbb{I}_{100})$ with (a) $\boldsymbol{\mu} = 0 \times \mathbf{1}_{100}, a = 1$; (b) $\boldsymbol{\mu} = 0.8 \times \mathbf{1}_{100}, a = 1$; (c) $\boldsymbol{\mu} = 0 \times \mathbf{1}_{100}, a = 1.4$; (d) $\boldsymbol{\mu} = 0 \times \mathbf{1}_{100}, a = 0.8$, where $\mathbf{1}_{100}$ is a length-100 vector whose elements are all one's. The same edge coloring scheme as in Fig. 1 is used here.

We next provide the exact analytic formulas for the expectation and variance of $(R_{G,1}(t), R_{G,2}(t))^T$ that are required to compute $M(t)$ so that we do not need to perform the time-consuming permutations to obtain them.

Theorem 1: The expectation, variance, and covariance of $R_{G,1}(t)$ and $R_{G,2}(t)$ under the permutation null distribution are:

$$\begin{aligned} &\mathsf{E}\left(R_{G,1}(t)\right) = nkp_1(t), \quad \mathsf{E}\left(R_{G,2}(t)\right) = nkq_1(t), \\ &\mathsf{var}\left(R_{G,1}(t)\right) = d_1p_1(t) + d_2p_2(t) + d_3p_3(t) - \left(nkp_1(t)\right)^2, \\ &\mathsf{var}\left(R_{G,2}(t)\right) = d_1q_1(t) + d_2q_2(t) + d_3q_3(t) - \left(nkq_1(t)\right)^2, \\ &\mathsf{cov}\left(R_{G,1}(t), R_{G,2}(t)\right) = d_3r(t) - \left(nkp_1(t)\right)\left(nkq_1(t)\right), \end{aligned}$$

where

$$\begin{aligned} p_1(t) &= \frac{t(t-1)}{n(n-1)}, \quad p_2(t) = \frac{t(t-1)(t-2)}{n(n-1)(n-2)}, \\ p_3(t) &= \frac{t(t-1)(t-2)(t-3)}{n(n-1)(n-2)(n-3)}, \\ q_1(t) &= \frac{(n-t)(n-t-1)}{n(n-1)}, \\ q_2(t) &= \frac{(n-t)(n-t-1)(n-t-2)}{n(n-1)(n-2)}, \\ q_3(t) &= \frac{(n-t)(n-t-1)(n-t-2)(n-t-3)}{n(n-1)(n-2)(n-3)}, \\ r(t) &= \frac{t(t-1)(n-t)(n-t-1)}{n(n-1)(n-2)(n-3)}, \end{aligned}$$

and $d_1 = c^{(1)} + c^{(2)}$, $d_2 = c^{(3)} + c^{(4)} + c^{(5)} + c^{(6)}$, $d_3 = c^{(7)}$, where $c^{(1)}, \ldots, c^{(7)}$ are quantities on the graph G, defined as:

$$c^{(1)} = nk, \quad c^{(2)} = \sum_{i=1}^{n} \sum_{j \in D_i} \mathbb{1}_{\{(i,j) \in G\}},$$

$$c^{(3)} = c^{(4)} = \sum_{i=1}^{n} \sum_{j \in D_i} (k - \mathbb{1}_{\{(i,j) \in G\}}), \quad c^{(5)} = nk(k-1),$$

$$c^{(6)} = \sum_{i=1}^{n} \left(|D_i|^2 - |D_i| \right), \quad c^{(7)} = (nk)^2 - \sum_{m=1}^{6} c^{(m)}.$$

Here, D_i is the set of indices of observations that point toward observation $\mathbf{y}_i$, and $|D_i|$ is the cardinality of set D_i, or the in-degree of observation $\mathbf{y}_i$.

Remark 1: The time complexity of computing $c^{(1)}, \ldots, c^{(7)}$ in Theorem 1 is $O(nk)$. One only has to construct a list of in-degrees for each observation in order to compute these seven quantities, which takes $O(nk)$ time.

Theorem 1 can be proved by combinatorial analysis. The expectations can be obtained easily by the linearity of expectation. For the variances and the covariance, we have to figure out the numbers of the seven possible configurations of pairs of edges as plotted in Fig. 3. The quantities $c^{(1)}, \ldots, c^{(7)}$ in Theorem 1 correspond respectively to the numbers of the seven configurations on a directed approximate k-NN graph. For such graphs, the out-degree of every observation node is a constant k, while the in-degree could vary from node to node, which requires one to scan through every edge to obtain the information. A detailed proof of Theorem 1 is in Supplement A.1.

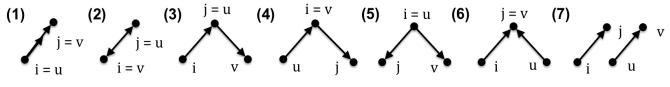


Fig. 3: Seven possible configurations of two edges $(i, j), (u, v)$ randomly chosen with replacement from a directed graph: (1) two edges degenerate into one ($(i, j) = (u, v)$); (2) two opposite edges (the two end nodes point to each other); (3)-(6) four different configurations with the two edges sharing one node; (7) two edges without any node sharing.

This max-type edge-count statistic $M(t)$ in (3) is well-defined under very mild conditions (Theorem 2). The proof is in Supplement A.2.

Theorem 2: The max-type edge-count statistic $\{M(t)\}_{t=1,\ldots,n-1}$ on a directed approximate k-NN graph is well-defined when $n \geq 5$ and not every observation has the same in-degree (i.e., there exists an i, $1 \leq i \leq n$, such that $|D_i| \neq k$).

The conditions in Theorem 2 ensure that the variances of $R_w(t)$ and $R_{\text{diff}}(t)$ are not zero. If all the observations have the same in-degree k, then $R_{\text{diff}}(t)$ is a constant. As the distribution of in-degrees could vary, we examine the most extreme case of which on the directed k-NN graph. Under the worst scenario, $n \geq 5$ ensures that the variance of $R_w(t)$ is positive.

III. Analytic type I error control

Given the max-type edge-count statistic, the next question is how large does the critical value need to be to constitute sufficient evidence against the null hypothesis of homogeneity (1). This is usually achieved by computing the p-value, which is defined as the probability of observing a more extreme or as extreme test statistic when the null hypothesis is ture. Here, the p-value is defined under the permutation null distribution of the max-type edge-count statisitc. As a relatively large value is the evidence for potential change-points, we are concerned with the tail probability of the test statistics under H_0:

$$\mathsf{pr}\Big(\max_{n_0 \leq t \leq n_1} M(t) > b \Big) \tag{5}$$

For a small n, the probability (5) can be obtained directly by permutation. However, when n is large, doing permutations could be very time-consuming. Hence, we derive analytic formulas to approximate the probability based on the asymptotic properties of the test statistic (6). We first work out the limiting distributions of $\{Z_w([nu]^1) : 0 < u < 1\}$ and $\{Z_{\text{diff}}([nu]) : 0 < u < 1\}$ jointly. On a directed graph, we use $e = (e_-, e_+)$ to denote an edge connected from e_- to e_+. Let $A_e = G_{e_-} \cup G_{e_+}$ be the subgraph in G that connect to either node e_- or node e_+, and $B_e = \cup_{e^* \in A_e} A_{e^*}$ be the subgraph in G that connect to any edge in A_e. In the following, we write $a_n = O(b_n)$ when a_n has the same order as b_n, and write $a_n = o(b_n)$ when a_n has order smaller than b_n.

Theorem 3: For a directed k-NN graph, if $k = O(n^\beta)$, $\beta < 0.25$, $\sum_{e \in G} |A_e||B_e| = o(n^{1.5(\beta+1)})$, $\sum_{e \in G} |A_e|^2 = o(n^{\beta+1.5})$, and $\sum_{i=1}^{n} |D_i|^2 - k^2 n = O(\sum_{i=1}^{n} |D_i|^2)$, as $n \to \infty$, $\{Z_w([nu]) : 0 < u < 1\}$ and $\{Z_{\text{diff}}([nu]) : 0 < u < 1\}$ converge to independent Gaussian processes in finite dimensional distributions.

The covariance functions of the limiting processes $\{Z_w([nu]) : 0 < u < 1\}$ and $\{Z_{\text{diff}}([nu]) : 0 < u < 1\}$ are provided in Supplement B.

The complete proof for Theorem 3 is in Supplement A.3. The key idea of the proof is to decouple the dependency resulted from the permutation null distribution and the dependency caused by the graph. More specifically, there are weak dependencies caused by the permutation as one observation appear at one time cannot appear at another time under permutation. To solve this, we take a step back and work on the bootstrap null distribution in which the probability of an observation appearing at one time does not affect by whether it appears at other time(s) or not. Thus, we could focus on dealing with the dependency caused by the graph, and the Stein's method is used to deal with the dependency. The bootstrap null distribution is then connected to the permutation null distribution by conditioning. Based on Theorem 3, the probability (5) can be approximated by

$$\begin{aligned} &\mathsf{pr}\Big(\max_{n_0 \leq t \leq n_1} M(t) > b \Big) \\ &\approx 1 - \mathsf{pr}\Big(\max_{n_0 \leq t \leq n_1} Z_w(t) < b \Big) \mathsf{pr}\Big(\max_{n_0 \leq t \leq n_1} |Z_{\text{diff}}(t)| < b \Big) \end{aligned} \tag{6}$$

[1]For a scalar x, we use $[x]$ to denote the largest integer no greater than x.

$$= 1 - \left(1 - \mathsf{pr}\Big(\max_{n_0 \le t \le n_1} Z_w(t) > b\Big)\right) \times \left(1 - \mathsf{pr}\Big(\max_{n_0 \le t \le n_1} |Z_{\text{diff}}(t)| > b\Big)\right).$$

The two probabilities in (6) can be computed similarly as in [22]:

$$\mathsf{pr}\Big(\max_{n_0 \le t \le n_1} Z_w(t) > b\Big) \approx b\phi(b) \int_{n_0}^{n_1} S_w(t) C_w(t) \nu\big(\sqrt{2b^2 C_w(t)}\big) dt \tag{7}$$

$$\mathsf{pr}\Big(\max_{n_0 \le t \le n_1} |Z_{\text{diff}}(t)| > b\Big) \approx 2b\phi(b) \int_{n_0}^{n_1} S_{\text{diff}}(t) C_{\text{diff}}(t) \nu\big(\sqrt{2b^2 C_{\text{diff}}(t)}\big) dt \tag{8}$$

where the function $\nu(\cdot)$ can be estimated numerically as $\nu(x) \approx \frac{(2/x)(\Phi(x/2)-0.5)}{(x/2)\Phi(x/2)+\phi(x/2)}$ with $\phi(\cdot)$ and $\Phi(\cdot)$ being the probability density function and cumulative distribution function of the standard normal distribution, respectively; $C_w(t)$, $C_{\text{diff}}(t)$ the partial derivative of the covariance function of the process; $S_w(t)$, $S_{\text{diff}}(t)$ the time-dependent skewness correction terms, i.e., for $j = w, \text{diff}$,

$$C_j(t) = \lim_{s \nearrow t} \frac{\partial \rho_j(s,t)}{\partial s}, \quad \rho_j(s,t) = \mathsf{cov}(Z_j(s), Z_j(t)),$$

$$S_j(t) = \frac{\exp\left(\frac{1}{2}(b - \hat{\theta}_{b,j}(t))^2 + \frac{1}{6}\gamma_j(t)\hat{\theta}^3_{b,j}(t)\right))}{\sqrt{1 + \gamma_j(t)\hat{\theta}_{b,j}(t)}};$$

where

$$\gamma_j(t) = \mathsf{E}\left(Z_j^3(t)\right), \quad \hat{\theta}_{b,j}(t) = (-1 + \sqrt{1 + 2b\gamma_j(t)})/\gamma_j(t).$$

Moreover, in the above expressions, $C_w(t)$ and $C_{\text{diff}}(t)$ can be derived and simplified to be (details in Supplement B)

$$C_w(t) = \frac{n(n-1)(2t^2/n - 2t + 1)}{2t(n-t)(t^2 - nt + n - 1)}, \quad C_{\text{diff}}(t) = \frac{n}{2t(n-t)}.$$

To compute $S_w(t)$ and $S_{\text{diff}}(t)$, we need the third moments of $Z_w(t)$ and $Z_{\text{diff}}(t)$, respectively. Comparing to that under undirected graphs, the computation here is much more complicated. For an undirected graph, there are 8 possible configurations (Fig. 4). However, for a directed graph, there are 24 possible configurations (Fig. 5). With brute force, it would take $O(|G|^3)$ time to compute the numbers of those configurations for a generic directed graph, which is very computationally expensive. To tackle this problem, we work out efficient formulas that can provide the results in $O(nk^2)$ time for directed k-NN graphs.

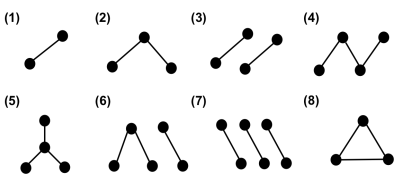


Fig. 4: Eight possible configurations of three edges randomly chosen with replacement from an undirected graph.

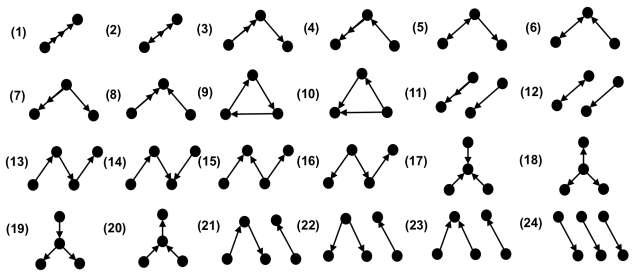


Fig. 5: Twenty-four possible configurations of three edges randomly chosen with replacement from a directed graph.

Let $G^{(m)}$ be the set of pairs of edges in G having the mth configuration as shown in Fig. 3, $m = 1, \ldots, 7$. Let $N^{(l)}$ be the number of occurrence for each of the configurations illustrated in Fig. 5, $l = 1, \ldots, 24$, then $\sum_{l=1}^{24} N^{(l)} = |G|^3$. We can obtain $N^{(l)}$'s with effort:

$$\begin{aligned}
N^{(1)} &= nk, \\
N^{(2)} &= 3c^{(2)}, \\
N^{(3)} &= 3c^{(3)}, \\
N^{(4)} &= 3c^{(6)}, \\
N^{(5)} &= 6c^{(2)}(k-1), \\
N^{(6)} &= 3 \sum_{(i,j),(u,v) \in G^{(2)}} (|D_i| + |D_j| - 2), \\
N^{(7)} &= 3c^{(5)}, \\
N^{(8)} &= 3c^{(3)}, \\
N^{(9)} &= 2 \sum_{(i,j),(u,v) \in G^{(3)}} \mathbb{1}_{\{(v,i) \in G\}}, \\
N^{(10)} &= 6 \sum_{(i,j),(u,v) \in G^{(3)}} \mathbb{1}_{\{(i,v) \in G\}}, \\
N^{(11)} &= 3c^{(7)}, \\
N^{(12)} &= 3c^{(2)}(nk-2) - (N^{(5)} + N^{(6)}), \\
N^{(13)} &= 6kc^{(3)} - (N^{(6)} + 3N^{(9)}), \\
N^{(14)} &= 6 \sum_{(i,j),(u,v) \in G^{(3)}} (|D_v| - 1) - N^{(10)}, \\
N^{(15)} &= 6(k-1)c^{(6)} - N^{(10)}, \\
N^{(16)} &= 6kc^{(5)} - (N^{(5)} + N^{(10)}), \\
N^{(17)} &= 6 \sum_{i=1}^{n} \binom{|D_i|}{3}, \\
N^{(18)} &= 6n \binom{k}{3}, \\
N^{(19)} &= 3 \sum_{(i,j),(u,v) \in G^{(5)}} |D_i| - N^{(5)}, \\
N^{(20)} &= 3kc^{(6)} - N^{(6)}, \\
N^{(21)} &= 6c^{(3)}(nk-2) - \Big(N^{(5)} + N^{(6)} + N^{(10)} + N^{(14)} \\
&\quad + N^{(16)} + 3N^{(9)} + 2N^{(13)} + 2N^{(19)} + 2N^{(20)}\Big),
\end{aligned}$$

$$N^{(22)} = 3c^{(5)}(nk-2) - \left(N^{(5)} + N^{(10)} + N^{(15)} + N^{(16)} + N^{(19)} + 3N^{(18)}\right),$$
$$N^{(23)} = 3c^{(6)}(nk-2) - \left(N^{(6)} + N^{(10)} + N^{(14)} + N^{(15)} + N^{(20)} + 3N^{(17)}\right),$$
$$N^{(24)} = (nk)^3 - \sum_{l=1}^{23} N^{(l)}.$$

In the above formulas, it takes at most $O(nk^2)$ time to compute $c^{(2)}, c^{(3)}, c^{(5)}$, and $c^{(6)}$, and there are at most nk^2 elements in the sets $G^{(2)}, G^{(3)}$, and $G^{(5)}$, so the numbers of the 24 configurations can be calculated within $O(nk^2)$ time. Indeed, as the rest of the computation is relatively straightforward (see Supplement C), the whole analytic p-value approximation procedure for a directed k-NN graph can also be done within $O(nk^2)$.

Now, let's check the performance of (6) through simulation studies.

TABLE I: Critical values for the statistic $\max_{n_0 \le t \le n_1} M(t)$ based on 3-NN's graph at $\alpha = 0.05$.

		Critical Values							
		$n_0 = 100$		$n_0 = 75$		$n_0 = 50$		$n_0 = 25$	
	d	Ana	Per	Ana	Per	Ana	Per	Ana	Per
(C1)	10	3.26	3.26	3.31	3.35	3.39	3.43	3.52	3.60
	100	3.29	3.29	3.36	3.40	3.45	3.52	3.62	3.78
	1,000	3.31	3.31	3.40	3.42	3.51	3.60	3.70	3.94
(C2)	10	3.26	3.26	3.32	3.34	3.39	3.44	3.51	3.60
	100	3.30	3.30	3.35	3.39	3.44	3.51	3.60	3.79
	1,000	3.30	3.30	3.46	3.54	3.59	3.75	3.80	4.27
(C3)	10	3.27	3.28	3.32	3.33	3.40	3.41	3.52	3.58
	100	3.28	3.28	3.35	3.37	3.43	3.48	3.58	3.70
	1,000	3.36	3.41	3.45	3.58	3.58	3.81	3.72	4.07

Table I shows the performance of the asymptotic p-value approximation of the max-type edge-count statistic (6) under different settings. We examine three different distributions (multivariate Gaussian (C1), multivariate t_5 (C2), and multivariate log-normal (C3) distributions) with different data dimensions ($d = 10, 100, 1000$). In Table I, column "Per" is the critical value obtained from doing $10,000$ permutations. This can be deemed as close to the true critical value. Column "Ana" presents the analytical critical values given by plugging (6) with (7) and (8). Here, the length of the sequence is $n = 1000$, and we present the results in four different choices of n_0 with $n_1 = n - n_0$. When $n_0 = 100$ or 75, the analytical approximation works quite well across all distributions and dimensions. As n_0 decreases ($n_0 = 50$ or 25), the analytical critical values become less precise. This is expected as the asymptotic distribution needs both groups to have $O(n)$ observations. When n_0 is small, one group could give a smaller order of observations than the other group. On the other hand, the accuracy of the analytical critical values is less dependent on the distribution of the data.

IV. Numerical results

A. Simulation setting and notations

We compare our proposed test with three state-of-the-art methods: graph-based method with the max-type edge-count statistic on 5-MST (the recommended setting in [22], denoted by "5-MST" in the following), the distance-based method (ecp) [20], and the kernel-based method [18]. For our method, we use the kd-tree algorithm [23] to approximate the directed 5-NN graph (d-a5NN). In the following, we focus on the Euclidean distance. There are implementations for other Minkowski distances in the ANN library (http://www.cs.umd.edu/~mount/ANN/). We use the directed approximate 5-NN graph as it contains a similar number of edges to the 5-MST to make the comparison with the existing graph-based method fair. The proposed method is denoted by "New" in the following.

B. Computational efficiency

First, we compare the computational cost of these methods through 10 simulation runs. In each simulation, the observations are generated i.i.d. from a multivariate Gaussian distribution with dimension $d = 500$. The results are presented in Table II. Among all the four methods, our proposed method is the fastest, whereas the kernel method is the slowest to run. For the graph-based methods, in particular, our proposed method on d-a5NN is more than 5 times faster than the method in [22] on 5-MST for $n = 2,000$ or above.

TABLE II: Runtime comparison: Average time cost in seconds (standard deviation) from 100 simulation runs for each choice of n (10 runs for the cells having average runtime greater than 1k seconds). The environment where the experiments are conduected: CPU: Intel(R) Xeon(R) CPU E5-2690 0 @ 2.90GHz / RAM: DDR3 @ 1600MHz / OS: Scientific Linux 6.10 / 2.6.32 Linux.

n	New	5-MST	ecp	kernel
$1,000$	0.8 (0.01)	5.2 (0.5)	13 (1.8)	683 (17)
$2,000$	2.7 (0.01)	21 (3.2)	52 (6.7)	10,224
$5,000$	17 (0.1)	157 (8.3)	482 (87)	>10,000
$10,000$	76 (1.2)	689 (27)	2,073 (387)	-
$20,000$	321 (3.7)	2,193 (189)	6,528 (1,276)	-
$30,000$	726 (5.9)	4,757 (106)	>10,000	-

C. Empirical size

Here, we check the empirical size of these methods under three significance levels ($\alpha = 0.10, 0.05$, and 0.01). Observations are generated i.i.d. from a d-dimensional multivariate Gaussian distribution with no change-point. The results are summerized in Table III, where the p-value of the graph-based methods is obtained through their corresponding analytical formulas, and those for the ecp are based on 999 random permutations. We only report those for $d = 25$ here, as the results are very similar for other choices of d (see Supplement

E). Both the graph-based methods and ecp could control the type I error well. However, for the kernel method, there is no direct mean to control type I error. We use the `'kcpa'` function in the R package `'ecp'`. In this function, the empirical size is supervised by a tuning parameter C. The larger the C is, the less likely the null is rejected. Unfortunately, it is not straightforward to link the tuning parameter C with the empirical size. As illustrated in Table IV, the relation between C and the empirical size depends heavily on the dimension of the observations.

TABLE III: Fractions of simulation runs (out of 10,000 simulations) that the null hypothesis is rejected when there is no change-point in the sequence ($n = 1,000$). Graph-based methods and ecp at level α.

Method	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
New	0.100	0.051	0.011
5-MST	0.096	0.051	0.012
ecp	0.098	0.050	0.011

TABLE IV: Fractions of simulation runs (out of 10,000 simulations) that the null hypothesis is rejected when there is no change-point in the sequence ($n = 1,000$). Kernel method with tuning parameter C under three different dimensions.

kernel method	$C = 4.8$	$C = 5.2$	$C = 5.6$
$d = 25$	0.943	0.821	0.631
$d = 30$	0.477	0.265	0.134
$d = 35$	0.094	0.036	0.009

D. Type II error analysis

To get an idea of the performance of the proposed method, we compare the probability of making the type II error, which is the event that the null hypothesis is not rejected when it is false, for the three methods can could control the type I error under some common parametric families. In particular, we consider six scenarios with each coordinate randomly generated from (1) Chi-square distributions with a change in the degree of freedom, (2) Weibull distribution with a change in the scale parameter while the shape parameter is fixed, (3) & (4) Gamma distributions with a change in one of the two parameters, respectively, while the other parameter is fixed, and (5) & (6) Beta distributions with a change in one of the two parameters, respectively, while the other parameter is fixed. In each simulation run, the length of the sequence is $n = 1,000$, and the change-point is at a quarter of the sequence $\tau = 250$.

- Setting 1 (Chi-square distribution): $F_0 = \chi^2_{\nu_0}$; $F_1 = \chi^2_{\nu_1}$.
- Setting 2 (Weibull distribution): $F_0 = \text{Weibull}(\lambda_0, k_0)$; $F_1 = \text{Weibull}(\lambda_1, k_0)$. Shape parameter fixed at $k_0 = 1$.
- Setting 3 (Gamma distribution-a): $F_0 = \text{Gamma}(\alpha_0, \beta_0)$; $F_1 = \text{Gamma}(\alpha_1, \beta_0)$. Scale parameter fixed at $\beta_0 = 1$.
- Setting 4 (Gamma distribution-b): $F_0 = \text{Gamma}(\alpha_0, \beta_0)$; $F_1 = \text{Gamma}(\alpha_0, \beta_1)$. Shape parameter fixed at $\alpha_0 = 1$.
- Setting 5 (Beta distribution-a): $F_0 = \text{Beta}(\alpha_0, \beta_0)$; $F_1 = \text{Gamma}(\alpha_1, \beta_0)$. Second parameter fixed at $\beta_0 = 0.5$.
- Setting 6 (Beta distribution-b): $F_0 = \text{Beta}(\alpha_0, \beta_0)$; $F_1 = \text{Gamma}(\alpha_0, \beta_1)$. First parameter fixed at $\alpha_0 = 0.5$.

TABLE V: Type II error: Fractions of times (out of 100) the null hypothesis is not rejected under $\alpha = 0.05$ for various data dimensions and sizes of change.

S1: Chi-square (d.f. ν change, $\nu_0 = 3$)

d	25	100	500	1000	2000
ν_1	3.27	3.20	3.15	3.12	3.09
New	**.16**	**.32**	**.13**	.11	.13
5-MST	.19	.37	**.13**	**.10**	**.11**
ecp	.95	.95	.95	.95	.96

S2: Weibull (scale λ change, $\lambda_0 = 1, k_0 = 1$)

d	25	100	500	1000	2000
λ_1	1.8	2.4	3.2	4.8	6.2
New	**.19**	**.17**	**.11**	**.18**	**.23**
5-MST	.24	.19	.13	.19	**.23**
ecp	.93	.97	.94	.93	.97

S3: Gamma (shape change, $\alpha_0 = 1, \beta_0 = 1$)

d	25	100	500	1000	2000
α_1	1.09	1.08	1.05	1.04	1.03
New	**.12**	**.15**	.34	.28	.33
5-MST	.19	.18	**.29**	**.22**	**.28**
ecp	.92	.91	.89	.95	.91

S4: Gamma (scale change, $\alpha_0 = 1, \beta_0 = 1$)

d	25	100	500	1000	2000
β_1	1.050	1.040	1.030	1.025	1.020
New	**.32**	**.29**	.25	.11	.12
5-MST	.36	.36	**.24**	**.08**	**.07**
ecp	.95	.93	.92	.90	.89

S5: Beta (shape 1 change, $\alpha_0 = 0.5, \beta_0 = 0.5$)

d	25	100	500	1000	2000
α_1	0.590	0.550	0.530	0.520	0.512
New	.23	**.19**	**.05**	**.03**	**.19**
5-MST	**.21**	.23	.06	.05	.25
ecp	.97	.96	.94	.96	.95

S6: Beta (shape 2 change, $\alpha_0 = 0.5, \beta_0 = 0.5$)

d	25	100	500	1000	2000
β_1	0.590	0.550	0.530	0.520	0.512
New	.25	**.14**	**.04**	**.05**	**.16**
5-MST	**.24**	.18	.06	.09	.24
ecp	.98	.93	.96	.92	.91

The results are shown in Table V. For each dimension, the alternatives are chosen so that the type II error is not too small to be comparable. We see that the type II error of the new test in on the small end for data from different distribution families.

E. Power comparison

Here, we compare the power of the proposed method to the other two methods. Power is the probability of rejecting the null hypothesis when it is false, i.e., the probability of not making the type II error. We consider six different scenarios. They are chosen to cover a variety of change types.

Scenarios 1-4 emphasize on the Gaussian distribution and cover changes in mean and variance, as well as different parts of the covariance matrix. Scenarios 5 and 6 cover asymmetric distributions and fat-tailed distributions. In the following, a and b are constants. N_d denotes a d-dimensional multivariate Gaussian distribution, $\mathbf{0}_d$ and $\mathbf{1}_d$ denote length-d vectors of all zeros and one's, respectively, $\mathbb{I}_d$ denotes a $d\times d$ identity matrix, and $\mathbf{\Sigma}$ denotes the covariance matrix with $\mathbf{\Sigma}_{ij} = 0.6^{|i-j|}$, where $\mathbf{\Sigma}_{ij}$ is the element of the ith row and the jth column of $\mathbf{\Sigma}$. The L_2 norm of the mean vector in F_1 is given by $||\Delta||_2$ in Table VI. In each simulation run, the length of the sequence is $n = 1,000$, and the change-point is at a quarter of the sequence $\tau = 250$.

- Scenario 1 (MG: mean and variance): $F_0 = N_d(\mathbf{0}_d, \mathbf{\Sigma})$; $F_1 = N_d(a \times \mathbf{1}_d, b\mathbf{\Sigma})$.
- Scenario 2 (MG: 5-coordinate): $F_0 = N_d(\mathbf{0}_d, \mathbb{I}_d)$; $F_1 = N_d((a\times\mathbf{1}_5, \mathbf{0}_{d-5})^T, \text{diag}((b\times\mathbf{1}_5, \mathbf{1}_{d-5})^T))$, where $\text{diag}(u)$ is a diagonal matrix with its diagonal vector u.
- Scenario 3 (MG: Diagonal): $F_0 = N_d(\mathbf{0}_d, \mathbb{I}_d)$; $F_1 = N_d(\mathbf{0}_d, b\mathbb{I}_d)$.
- Scenario 4 (MG: Off-diagonal): $F_0 = N_d(\mathbf{0}_d, \mathbf{\Sigma})$; $F_1 = N_d(\mathbf{0}_d, \mathbf{\Sigma'})$, where $\mathbf{\Sigma'}_{ij} = \rho^{|i-j|}$.
- Scenario 5 (Chi-square distribution): $F_0 = \mathbf{\Sigma}^{\frac{1}{2}}\mathbf{u}^{\chi^2_{3,c}}$; $F_1 = (b\mathbf{\Sigma})^{\frac{1}{2}}\mathbf{u}^{\chi^2_{3,c}} + a \times \mathbf{1}_d$. Here, $\mathbf{u}^{\chi^2_{3,c}}$ is a length-d vector with each component i.i.d. from the centered χ^2_3 distribution.
- Scenario 6 (t-distribution): $F_0 = \mathbf{\Sigma}^{\frac{1}{2}}\mathbf{u}^{t_5}$; $F_1 = (b\mathbf{\Sigma})^{\frac{1}{2}}\mathbf{u}^{t_5} + a \times \mathbf{1}_d$. Here, $\mathbf{u}^{t_5}$ is a length-d vector with each component i.i.d. from the t_5-distribution.

From Table VI, we can see that our proposed test has good power under a wide range of alternatives. It is the best or on par with the best in these simulation studies. In sharp contrast, the ecp method suffers from the curse of dimensionality, and could have low power when the change contain sources other than the mean shift.

TABLE VI: Power comparison: Numbers of times (out of 100) the null hypothesis is rejected under $\alpha = 0.05$ for various data dimensions and sizes of change.

S1: MG (Mean and Variance)

d	25	100	500	1000	2000
$\|\|\Delta\|\|_2$	0.10	0.20	0.45	0.63	0.89
b	1.10	1.06	1.03	1.02	1.02
New	**75**	**76**	**65**	**58**	**83**
5-MST	71	66	60	56	**83**
ecp	4	8	10	13	15

S2: MG (5-coordinate)

d	25	100	500	1000	2000
$\|\|\Delta\|\|_2$	0.20	0.40	0.67	0.63	0.89
b	1.8	2.4	3.2	4.8	6.2
New	**94**	**84**	**63**	64	69
5-MST	92	83	60	**65**	**71**
ecp	17	36	33	22	34

S3: MG (Diagonal)

d	25	100	500	1000	2000
b	1.10	1.06	1.03	1.02	1.02
New	**69**	**78**	87	**90**	**99**
5-MST	60	77	**88**	88	**99**
ecp	3	7	4	2	4

S4: MG (Off-diagonal)

d	25	100	500	1000	2000
ρ	0.53	0.50	0.48	0.47	0.46
New	**74**	**88**	**89**	**88**	**89**
5-MST	68	83	87	87	88
ecp	4	9	8	4	3

S5: Chi-square distribution

d	25	100	500	1000	2000
$\|\|\Delta\|\|_2$	0.05	0.10	0.22	0.32	0.45
b	1.10	1.08	1.05	1.04	1.03
New	**71**	**81**	**85**	**85**	**89**
5-MST	66	80	**85**	83	**89**
ecp	4	10	3	7	2

S6: t-distribution

d	25	100	500	1000	2000
$\|\|\Delta\|\|_2$	0.20	0.40	0.67	0.63	0.89
b	1.14	1.08	1.05	1.04	1.03
New	**85**	**75**	**73**	86	85
5-MST	81	73	70	**87**	**87**
ecp	8	15	18	11	11

F. Types of changes the new method can detect

Here, we check the types of changes the proposed method can detect. We investigate five types of changes: mean, variance, covariance, skewness and kurtosis. All experiments are conducted under the setting with $n = 1,000$, $\tau = 250$ and $d = 1,000$. Below describes in details the five scenrios:

- Change in mean: Before the change, all coordinates are from independent standard Gaussian distributions. After the change, the L_2-norm of the mean vector is specified by the x-axis in Fig. 6. We study the power for changes in all coordinates ($dc = 1,000$), one coordinate ($dc = 1$), and some subsets of the coordinates ($dc = 200, 50, 10$). We can see from Fig. 6 that our test has good power to mean change regardless of the number of coordinates that has a mean shift.
- Change in variance: Before the change, all coordinates are from independent standard Gaussian distributions. After the change, the determinant of the variance-covariance matrix becomes a^{1000}, where a is specified by the x-axis in Fig. 7. We study the power for changes in all coordinates ($dc = 1,000$), one coordinate ($dc = 1$), and some subsets of the coordinates ($dc = 200, 50, 10$). We can see from Fig. 7 that our test is generally sensitive to variance change in all cases, and the power is higher when the change comes in fewer coordinates.
- Change in covariance: Before the change, the observations are from multivariate Gaussian distribution with mean zero and variance-covariance matrix $\mathbf{\Sigma}_{ij} = 0.6^{|i-j|}$, denoted as $\rho_0 = 0.6$. After the change, the variance-covariance matrix becomes $\mathbf{\Sigma}_{ij} = \rho_1^{|i-j|}$ and new correlation coefficent is defined as $\rho_1 = 0.6 - \Delta\rho$. The ten values of the $\Delta\rho$'s used in the experiment

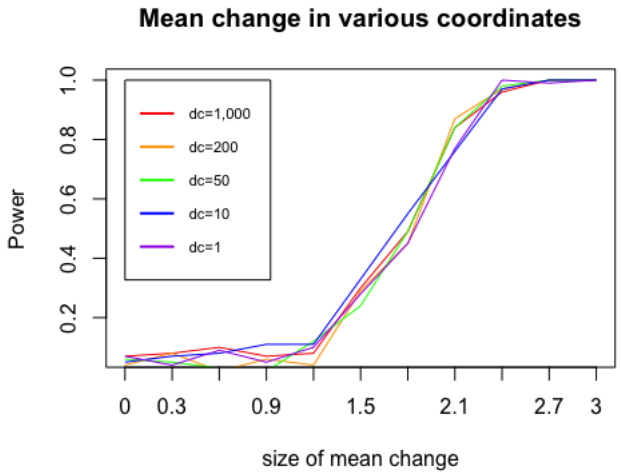


Fig. 6: Fraction of times (out of 100) that a change-point is detected at a given size of mean change for the changes in various coordinates (dc).

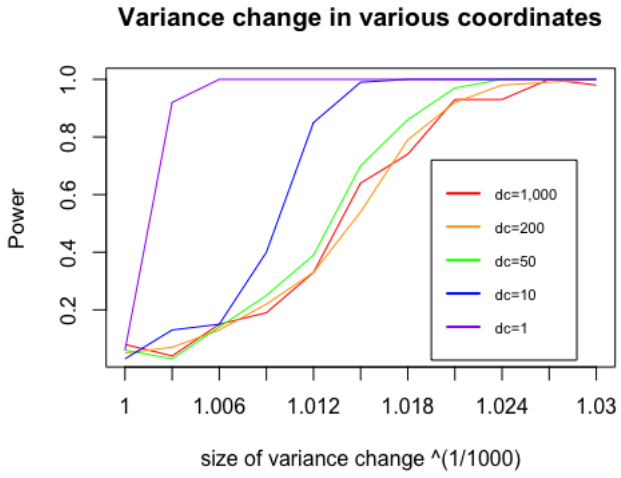


Fig. 7: Fraction of times (out of 100) that a change-point is detected at a given size of variance change for the changes in various coordinates (dc).

are $(0.02, 0.04, \ldots, 0.20)$, corresponding to the x-axis $(1, 2, \ldots, 10)$ in Fig. 8. The result shows that our new method is also very sensitive to changes in the covariance structure.

- Change in skewness: Before the change the observations in each coordinate are from independent Gaussian distributions with mean ν and standard deviation $\sqrt{2\nu}$. After the change, observations in each coordinate are from independent Chi-square distributions with degree of freedom ν. The skewness of a χ^2_ν distribution is computed as $\sqrt{8/\nu}$. The ten values of the ν's used in the experiment are chosen so that the skewness of each variable are $(0.2, 0.4, 0.6, \ldots, 2.0)$, corresponding to the x-axis $(1, 2, \ldots, 10)$ in Fig. 8. This setting does not change mean and variance while changing the skewness.
- Change in excess kurtosis: Before the change the observations in each coordinate are from independent standard Gaussian distributions. After the change, observations in each coordinate are from independent t distributions with degree of freedom ν. The excess kurtoosis of a t_ν distribution is computed as $\frac{6}{\nu-4}$. The ten values of the ν's used in the experiment are chosen so that the excess kurtosis of each variable are $(0.01, 0.02, 0.03, \ldots, 0.10)$, corresponding to the x-axis $(1, 2, \ldots, 10)$ in Fig. 8. The result shows that our method can also detect changes in excess kurtosis.

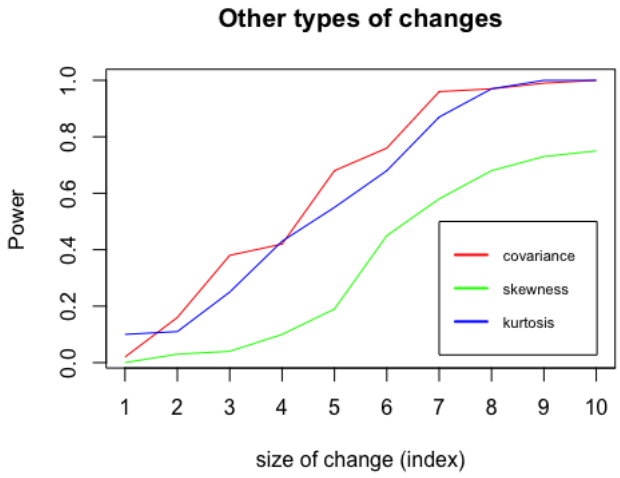


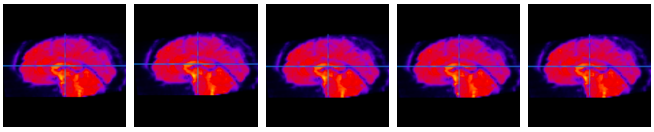
Fig. 8: Fraction of times (out of 100) that a change-point is detected at a given size of changes in covariance, skewness and excess kurtosis. The sizes of changes increase as the index in the x-axis grows, but the sizes among the three scenarios at each index are not comparable. We plot them on the same figure to save space.

V. Real Data Applications

A. fMRI data

This fMRI dataset was recorded when the subjects were watching certain pieces of the movie "The Grand Budapest Hotel" by Wes Anderson. It is publicly available at: `https://openneuro.org/datasets/ds003017/versions/1.0.2`. There are in total 25 subjects involved in this experiment, each of them watching 5 pieces of the movie [27]. Here, we randomly select two such sequences with subject ID SID-000005 and SID-000024 for illustration.

This piece of the movie is about 10 minutes long. The total length of the sequence is $n = 598$ with one time unit as 1 second. Each observation is a 3-dimensional fMRI image with size $96 \times 96 \times 48$. To get an idea of how the fMRI data look like, Fig. 9 shows the profiles of five observations at $t = 150, 250, 350, 450$, and 550, from one certain perspective. There are three different perspectives available, and the other two can be found in Supplement F. We rearrange each observation into a length-d vector ($d = 96 \times 96 \times 48 = 442,368$).



$t = 150$ $t = 250$ $t = 350$ $t = 450$ $t = 550$

Fig. 9: The snapshots of the fMRI images for subject SID-000005 at different timestamps.

In the first sequence (SID-000005), the signal is strong that all three methods find the change-point at around 435 (see Table VII). We can see from the heatmap of the pairwise distances of the observations (Fig. 10) that the existence of a change-point at around 435 is reasonable. In the second sequence (SID-000024), all three methods find the change-point at around 260, which also appears to be consistent with the heatmap (Fig. 11). Among the three methods, the new test is much more efficient to run (Table VII, last column). We can also observe that here the computation time for 5-MST is similar to that of ecp as constructing the 5-MST dominates the overall runtime when d is large.

TABLE VII: Results of the estimated change-point locations ($\hat{\tau}$), p-values, and the overall runtimes. For the two graph-based methods, the analytical p-values are reported; for the ecp method, the p-value is based on 999 permutaions.

Subject	Methods	$\hat{\tau}$	p-value	time cost (minutes)
SID-000005	New (d-a5NN)	437	< 0.001	3.8
	5-MST	437	< 0.001	120.9
	ecp	433	0.001	118.4
SID-000024	New (d-a5NN)	260	< 0.001	3.9
	5-MST	260	< 0.001	147.8
	ecp	261	0.001	133.1

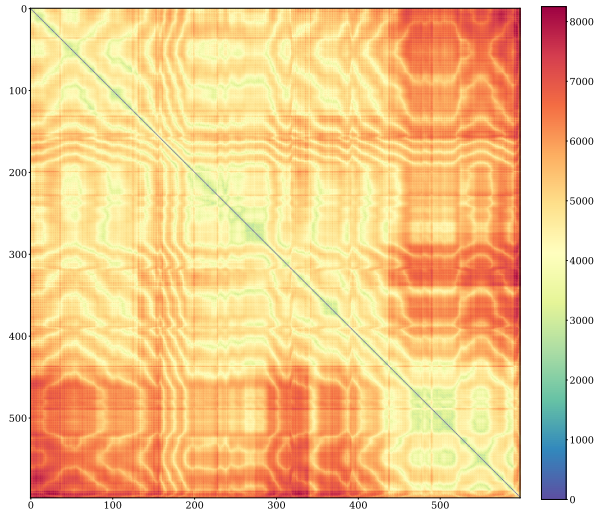


Fig. 10: Heatmap of pairwise distances of the observations in the sequence (SID-000005).

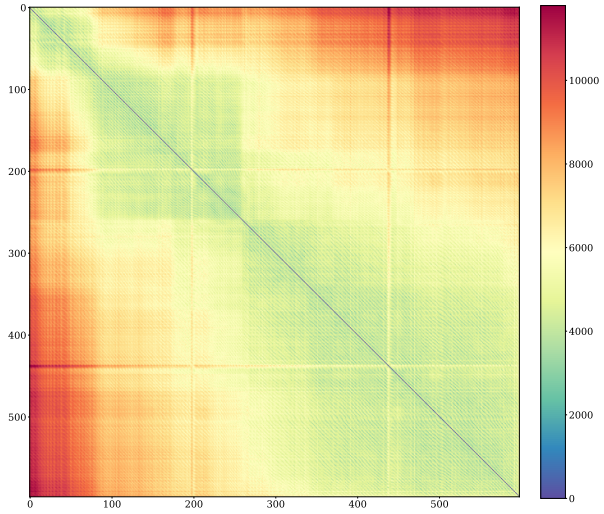


Fig. 11: Heatmap of pairwise distances of the observations in the sequence (SID-000024).

B. Neuropixels data

Neuropixels probes are new technology in neuroscience that can record hundreds of sites in the brain simultaneously [6], [28]. Here, we analyze a dataset that records the spiking activities of the neurons in the brain of a mouse while it is awake in darkness during spontaneous behavior. The dataset is publicly available at: `https://figshare.com/articles/dataset/Eight-probe_Neuropixels_recordings_during_spontaneous_behaviors/7739750`. This dataset contains simultaneous recordings from nine brain regions, with each region having hundreds of recording sequences. This dataset was analyzed in [29], and we follow the same preprocessing procedure there. The lengths of all the sequences are the same $n = 39,053$, and the dimensions are the numbers of recordings which vary from $d = 42$ to $d = 334$. We apply the two graph-based methods to all the nine sueqences. The results are presented in Table VIII. The ecp method is not applicable to this dataset because the memory space is not enough under the same environment as in Table II.

We see that the new method on the directed approximate 5-NN graph is on average ten times faster than the method in [22] on 5-MST. Such improvement can be very imperative especially when analyzing large datasets.

VI. Conclusion

As we enter the era of big data, the importance of the scalibility of statistical methods or data analysis teachniques cannot be overemphasized. Nowadays we are collecting data with exploding sizes (either the dimensionality gets higher or the sequence of observations gets longer). To address this problem, we propose a new nonparametric framework for change-point analysis using the information of approximate k-NN graphs. The time complexity of performing our proposed test is $O\left(dn(\log n + k \log d) + nk^2\right)$, and our method is so

TABLE VIII: Results of the estimated change-point locations ($\hat{\tau}$), p-values, and the overall runtimes (in minutes). For the two graph-based methods, the analytical p-values are reported.

Region	Methods	$\hat{\tau}$	p-value	time
Caudate putamen ($d = 176$)	New (d-a5NN)	35,148	< 0.001	7.7
	5-MST	35,056	< 0.001	96.1
Frontal motor ($d = 78$)	New (d-a5NN)	31,081	< 0.001	6.0
	5-MST	32,242	< 0.001	77.8
Hippocampus ($d = 265$)	New (d-a5NN)	4,109	< 0.001	20.7
	5-MST	4,382	< 0.001	159.1
Lateral septum ($d = 122$)	New (d-a5NN)	29,616	< 0.001	11.4
	5-MST	29,636	< 0.001	89.3
Midbrain ($d = 127$)	New (d-a5NN)	20,580	< 0.001	13.9
	5-MST	20,590	< 0.001	105.6
Superior colliculus ($d = 42$)	New (d-a5NN)	23,539	< 0.001	4.0
	5-MST	31,328	< 0.001	65.4
Somatomotor ($d = 91$)	New (d-a5NN)	30,316	< 0.001	7.6
	5-MST	30,312	< 0.001	81.9
Thalamus ($d = 227$)	New (d-a5NN)	28,613	< 0.001	21.7
	5-MST	28,608	< 0.001	146.1
V1 ($d = 334$)	New (d-a5NN)	30,226	< 0.001	17.5
	5-MST	30,338	< 0.001	173.8

far the fastest change-point detection method available with a proper control on the false discovery rate.

In constructing the test statistic, we take into account a pattern caused by the curse of dimensionality. As a result, the new test can detect various types of changes (such as change in mean and/or variance, and change in the covariance structure) in long sequences of moderate- to high- dimensional data. Moreover, our method does not impose any distributional assumption on the data, making it desirable in many real applications where the distribution could be heavy-tailed and/or skewed, the dimension of the data could be much higher than the number of observations, and the change could be global or in a sparse/dense subset of coordinates.

We apply our method to two large real datasets, the fMRI images and the Neuropixels recordings, with the former having a very large dimension and the latter a very large sample size. Both examples show that our new method improves the computational efficiency upon the existing methods by a significant amount, while its performance remains as reliable as other state-of-the-art methods.

REFERENCES

[1] M. Basseville, I. V. Nikiforov *et al.*, *Detection of abrupt changes: theory and application*. Prentice hall Englewood Cliffs, 1993, vol. 104.

[2] B. Brodsky and B. Darkhovsky, "Applications of nonparametric change-point detection methods," in *Nonparametric Methods in Change-Point Problems*. Springer, 1993, pp. 169–182.

[3] E. G. Carlstein, H.-G. Müller, and D. Siegmund, "Change-point problems." IMS, 1994.

[4] M. Csörgö, M. Csörgö, and L. Horváth, "Limit theorems in change-point analysis," 1997.

[5] J. Chen and A. K. Gupta, *Parametric statistical change point analysis: with applications to genetics, medicine, and finance*. Springer Science & Business Media, 2011.

[6] J. J. Jun, N. A. Steinmetz, J. H. Siegle, D. J. Denman, M. Bauza, B. Barbarits, A. K. Lee, C. A. Anastassiou, A. Andrei, C. Aydın *et al.*, "Fully integrated silicon probes for high-density recording of neural activity," *Nature*, vol. 551, no. 7679, pp. 232–236, 2017.

[7] D. Q. Zeebaree, H. Haron, and A. M. Abdulazeez, "Gene selection and classification of microarray data using convolutional neural network," in *2018 International Conference on Advanced Science and Engineering (ICOASE)*. IEEE, 2018, pp. 145–150.

[8] C. Lee, Z. Luo, K. Y. Ngiam, M. Zhang, K. Zheng, G. Chen, B. C. Ooi, and W. L. J. Yip, "Big healthcare data analytics: Challenges and applications," in *Handbook of large-scale distributed computing in smart healthcare*. Springer, 2017, pp. 11–41.

[9] P. Fryzlewicz, "Detecting possibly frequent change-points: Wild binary segmentation 2 and steepest-drop model selection," *Journal of the Korean Statistical Society*, pp. 1–44, 2020.

[10] P. Fryzlewicz *et al.*, "Wild binary segmentation for multiple change-point detection," *The Annals of Statistics*, vol. 42, no. 6, pp. 2243–2281, 2014.

[11] S. Kovács, H. Li, P. Bühlmann, and A. Munk, "Seeded binary segmentation: A general methodology for fast and optimal change point detection," *arXiv preprint arXiv:2002.06633*, 2020.

[12] M. Barigozzi, H. Cho, and P. Fryzlewicz, "Simultaneous multiple change-point and factor analysis for high-dimensional time series," *Journal of Econometrics*, vol. 206, no. 1, pp. 187–225, 2018.

[13] D. Wang, Y. Yu, and A. Rinaldo, "Optimal change point detection and localization in sparse dynamic networks," *arXiv preprint arXiv:1809.09602*, 2018.

[14] M. Bhattacharjee, M. Banerjee, and G. Michailidis, "Change point estimation in a dynamic stochastic block model," *arXiv preprint arXiv:1812.03090*, 2018.

[15] M. Londschien, S. Kovács, and P. Bühlmann, "Change-point detection for graphical models in the presence of missing values," *Journal of Computational and Graphical Statistics*, pp. 1–12, 2021.

[16] Z. Harchaoui and O. Cappé, "Retrospective mutiple change-point estimation with kernels," in *2007 IEEE/SP 14th Workshop on Statistical Signal Processing*. IEEE, 2007, pp. 768–772.

[17] Z. Harchaoui, E. Moulines, and F. R. Bach, "Kernel change-point analysis," in *Advances in neural information processing systems*, 2009, pp. 609–616.

[18] S. Arlot, A. Celisse, and Z. Harchaoui, "A kernel multiple change-point algorithm via model selection," *Journal of Machine Learning Research*, vol. 20, no. 162, pp. 1–56, 2019. [Online]. Available: http://jmlr.org/papers/v20/16-155.html

[19] S. Li, Y. Xie, H. Dai, and L. Song, "Scan b-statistic for kernel change-point detection," *Sequential Analysis*, vol. 38, no. 4, pp. 503–544, 2019.

[20] D. S. Matteson and N. A. James, "A nonparametric approach for multiple change point analysis of multivariate data," *Journal of the American Statistical Association*, vol. 109, no. 505, pp. 334–345, 2014.

[21] H. Chen and N. Zhang, "Graph-based change-point detection," *The Annals of Statistics*, vol. 43, no. 1, pp. 139–176, 2015.

[22] L. Chu and H. Chen, "Asymptotic distribution-free change-point detection for multivariate and non-euclidean data," *The Annals of Statistics*, vol. 47, no. 1, pp. 382–414, 2019.

[23] A. Beygelzimer, S. Kakadet, J. Langford, S. Arya, D. Mount, and S. Li, *Fast Nearest Neighbor Search Algorithms and Applications*, 2019, r package FNN: kd-tree fast k-nearest neighbor search algorithms.

[24] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 891–923, 1998.

[25] P. Ram and K. Sinha, "Revisiting kd-tree for nearest neighbor search," in *Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining*, 2019, pp. 1378–1388.

[26] H. Chen and J. H. Friedman, "A new graph-based two-sample test for multivariate and object data," *Journal of the American statistical association*, vol. 112, no. 517, pp. 397–409, 2017.

[27] M. Visconti di Oleggio Castello, V. Chauhan, G. Jiahui, and M. I. Gobbini, "The grand budapest hotel: an fmri dataset in response to a socially-rich, naturalistic movie," *bioRxiv*, 2020. [Online]. Available: https://www.biorxiv.org/content/early/2020/07/14/2020.07.14.203257

[28] C. Stringer, M. Pachitariu, N. Steinmetz, C. B. Reddy, M. Carandini, and K. D. Harris, "Spontaneous behaviors drive multidimensional, brainwide activity," *Science*, vol. 364, no. 6437, 2019.

[29] H. Chen, S. Chen, and X. Deng, "A universal nonparametric event detection framework for neuropixels data," *bioRxiv*, 2019. [Online]. Available: https://www.biorxiv.org/content/early/2019/05/27/650671