# Neural Network-Based Change Point Detection for Large-Scale Time-Evolving Data

Jialiang Geng, George Michailidis [*][†]

March 13, 2025

## Abstract

The paper studies the problem of detecting and locating change points in multivariate time-evolving data. The problem has a long history in statistics and signal processing and various algorithms have been developed primarily for simple parametric models. In this work, we focus on modeling the data through feed-forward neural networks and develop a detection strategy based on the following two-step procedure. In the first step, the neural network is trained over a prespecified window of the data, and its test error function is calibrated over another prespecified window. Then, the test error function is used over a moving window to identify the change point. Once a change point is detected, the procedure involving these two steps is repeated until all change points are identified. The proposed strategy yields consistent estimates for both the number and the locations of the change points under temporal dependence of the data-generating process. The effectiveness of the proposed strategy is illustrated on synthetic data sets that provide insights on how to select in practice tuning parameters of the algorithm and in real data sets. Finally, we note that although the detection strategy is general and can work with different neural network architectures, the theoretical guarantees provided are specific to feed-forward neural architectures.

**Keywords:** change point detection, neural network, nonparametric method, time-evolving data

## 1 Introduction

The problem of change point detection has a rich history across various disciplines, including statistics, econometrics, signal processing, and machine learning, owing to its wide range of applications in areas such as quality control [1], economics, finance, and risk analysis [2], medical diagnosis [3], social network

---

[*]First Author is with the Department of Statistics, University of California, Los Angeles (email: cauchygeng@g.ucla.edu).

[†]Second Author is with the Department of Statistics, University of California, Los Angeles (email: gmichail@g.ucla.edu).

analysis [4], and linguistics [5,6]. Change point detection methods are typically categorized according to the task in hand: *online* methods aim to detect changes in the data distribution/dynamics, as soon as they occur based on streaming data, while *offline* methods aim to *retrospectively* detect changes, when all data are available for analysis. The former task is also referred to in the literature as event or anomaly detection, while the latter task is also sometimes called signal segmentation.

Various detection algorithms have been developed for a number of *parametric* models in the offline setting, including exhaustive search [7], penalized cost approaches [8,9] and approaches based on screening and ranking criteria [10]. Many of these detection algorithms also come with probabilistic guarantees on the number and locations of the change points detected (Asymptotic consistency) for specific parametric models, such as mean shift time series models [11], vector autoregressive models [12,13], regression type models [14,15], random graph and stochastic block network models [16,17]. Meanwhile, the evaluation of the different detection algorithms is also important. The paper [18] offers a comprehensive overview of algorithms for offline change point detection in multivariate time series and discusses consistency for these methods including Asymptotic consistency and evaluation metrics including Hausdorff distance and F1 score, many of which will also be used in this paper.

The rapid advancement and widespread use of neural network architectures have highlighted the need for developing change point detection methods tailored specifically to these models, as most existing approaches are designed for parametric models. Current literature on this topic is relatively sparse and primarily focuses on online change point detection. For instance, [19] proposed a sequential detection procedure with a predefined detection window, utilizing a generalized likelihood ratio test where the likelihood is approximated by a simple feed-forward network. Similarly, [20] reformulated the detection problem as a classification task, employing neural network architectures to train a classifier. In [21], change point detection is performed using a novel deep neural network (DNN) structure called the Pyramid Recurrent Neural Network, which combines recurrent neural networks (RNNs) and convolutional neural networks (CNNs). The output is then passed through a binary classifier to determine the presence of a change point. Another approach, described in [22], integrates singular spectral analysis with auto-associative neural networks, using the residuals to detect change points based on a specified threshold. Lastly, [23] combines preprocessing steps such as normalization and recursive singular spectral analysis with a neural network-based autoencoder. The reconstruction error from the autoencoder serves as the criterion for identifying change points under a predefined threshold.

On the other hand, to the best of our knowledge, change point detection methods for *nonparametric offline setting* time evolving data mainly focused distribution change instead of time evolving mechanism change. Paper [24] proposed a kernel-based model selection framework that minimizes a penalized kernel least-squares criterion enabling the detection of an unknown number of change points. The optimal offline time segmentation method in [25] inno-

vatively combines dynamic programming, cost-based optimization, and error guarantees to offer a robust solution. Random Forests for Change Point Detection in [26] proposed a nonparametric method leveraging machine learning classifiers, such as random forests, to construct a classifier log-likelihood ratio which is used to detect multiple change points in time series data. However, the deep learning based methodologies are sparse. The paper [27] proposed a mSSA-based CPD algorithm for detecting change points in multivariate time series and focused on the spectral change instead of the traditional mean change or variance change. Existing deep learning methods including KLCPD [28] used kernel-based hypothesis testing and deep generative models to mitigate insufficient samples in the abnormal distribution and the paper [29] combined the CUSUM Statistics and deep learning. These paper mainly focused on the change of distribution instead of change of time evolving mechanism and many of the literature does not include the theoretical analysis of Asymptotic consistency. Meanwhile, some literature including [26] and [27] only tackle the single change point problem. Hence, the goal of this work is to develop a detection algorithm for feed-forward networks focusing in time evolving data that comes with probabilistic guarantees on identifying the correct number and locations of multiple change points. It is based on a two-step procedure and leverages a nonparametric criterion function.

Based on the few available references above, emphasize differences in the methodology, in addition to theory in the next paragraph. The main contribution of this paper are

I. Developed a nonparametric two-step framework leveraging feed-forward neural networks and a systematic approach to work on both single change point case and multiple change points case in nonparametric multidimensional time-evolving data which is illustrated by algorithm 1 and 2

II. The algorithm works in both the regression problem demonstrated in section 2 equation (1) and time evolving data like Vector Autoregressive (VAR) model.

III. Theorems in section4 provided rigorous theoretical analysis and guarantees for accuracy in detecting the number of change points and bounding the distance between true and estimated change points. The theoretical analysis varies in different cases of the dataset including temporal independent cases in theorem 4.2 and dependent dataset in theorem 4.4 and subgaussian dataset in theorem 4.3.

IV. The effectiveness of the proposed strategy is illustrated on synthetic data sets that provide insights on how to select in practice tuning parameters of the algorithm and in real datasets.

The remainder of the paper is organized as follows: Section2 sets up the problem under consideration and introduces the nonparametric criterion function. Section 3 describes the two-step algorithm, while Section 4 provides the theoretical guarantees for it. Finally, Section 5 presents numerical experiments

3

based on simulated data that showcase the role that various assumptions play in the performance of the algorithm, while Section 5 illustrates the algorithm on selected real-world data sets.

# 2    Problem Formulation

The setting under consideration is as follows: Let $Y_i, i = 1, \cdots, T_{sum}$ be a response/output vector of dimension $h$ and $X_i, i = 1, \cdots, T_{sum}$ a vector of dimension $p$ of covariates/attributes, related through the following predictive model:

$$Y_i = f_j(X_i) + \epsilon_i, \quad \tau_{j-1} < i \le \tau_j \tag{1}$$

where $\tau_0 \le i \le \tau_{N+1}, \tau_0 = 0, \tau_{N+1} = T_{sum}$ correspond to the N *change points* and $T_{sum}$ is the number of total time points available. It can be seen that the model $f_j$ is applicable for a segment of time points and changes to another one $f_{j+1}$ at change point $\tau_j + 1$. The model function $f_j, \mathbb{R}^p \to \mathbb{R}^h$ is assumed to be measurable and composed of several functions (technical details provided in the Supplement). This framework can also cover the Vector Autoregressive (VAR) model:

$$Y_i = A_{1j}Y_{i-1} + ... + A_{qj}Y_{i-q} + \epsilon_i, \tau_{j-1} \le i \le \tau_j \tag{2}$$

where $Y_i \in \mathbb{R}^h$ and $A_{ij} \in \mathbb{R}^{h \times h}, 1 \le i \le q$ is the transition (regression coefficient) matrix. If we treat the q lags of $Y_i$ as $X_i = Y_{i-q \le k < i}$, then $Y_i = \bar{f}_j(X_i) + \epsilon_i$ where $\bar{f}_j = \bar{f}_j(A_{ij})$ which can be covered by the proposed framework.(Technical details can be found in section 5).

**Criterion function:** The following criterion function is employed in the detection algorithm: $E(t) = \sum_{i=t}^{T_2+t} ||Y_i - \hat{f}(X_i)||^2$, where a fully connected feedforward network $\hat{f} : \mathbb{R}^{p_0} \to \mathbb{R}^{p_{L+1}}$ is defined as: $\hat{f} = W_L \sigma_{v_L} W_{L-1} \sigma_{v_{L-1}} ... W_1 \sigma_{v_1} W_0 x$ and $L$ is the number of layers of the network, $W_j$ is the $p_{j+1} \times p_j$ matrix, $v_j \in \mathbb{R}^{p_j}$ and $\sigma_{v_j}(x) = \max(0, x - v_j)$ is the ReLU function with bias $v_j$. Here $p_0 = p$ and $p_{L+1} = h$. $||.||$ is the 2-norm of a vector or matrix. The criterion function measures the squared error loss between the actual and fitted values and mimics analogous criteria used in change point analysis for traditional parametric statistical models. Further, the neural network is trained over a certain time period (to be specified in section 3), and then the criterion function is evaluated over a window of size $T_2$, whose starting point is a generic point in time t. The length of the time window $T_2$ is also specified in the section 3.

In this paper, we used this neural network architecture represent $\hat{f}$ and estimate the true underlying non-linear predictive model $f_j$ which can also be estimated by other regression model including kernel regression in other literature.

# 3    Detection Algorithm

**Annotations and assumptions:** The $T_{sum}$ is total number of time points available as defined in section 2. $T_1, T_2, T_3$ refer to the training window size,

4

test window size and detection window size. $\pi$ refers to the detection threshold.

**Assumption 3.1** *Define the signal of the model change $M1_i$ and $M1_i \leq \mathbb{E}[||f_{i+1}(X) - f_i(X)||^2] \leq M2_i < \infty$ where $f_i$ is model function defined in equation 1 and $M2_i$ is some constant and define $M1_* = \min_i M1_i$ as the minimum signal of the model change. We assume*

$$M1_* > 4h\sigma^2 \tag{3}$$

This assumption ensure the signal of the model change is large enough compared to noise so the algorithm will be able to detect the change point.

**Assumption 3.2** *Define $T = \min_{i=0,\ldots,N} |\tau_{i+1} - \tau_i|$ as the minimum distance between change points. We assume it has a lower bound:*

$$T \geq \frac{C_0 b(T_{sum})}{M1_*} \tag{4}$$

*for some constant $C_0$ and function $b$. Usually, we use $b(T_{sum}) = \sqrt{T_{sum}}$*

This assumption provide a lower bound for the distance between true change points and a function of the relationship between this distance and model change signal. Such balance between signal change and distance between true change points are very common in literature like the papers in review [18].

Recall that in the assumption, $\tau_0 = 0, \tau_{N+1} = T_{sum}$. This assumption also restrict the change points should be distant enough from the boundaries as well which can let the model pretrained on the data before the first change point and retrained after the change point. This requirement is also very common and mild in change point detection and the classical method always require the change point $cT_{sum} \leq \tau \leq (1-c)T_{sum}$ for some constant c. Then the search for change point starts after a time interval proportional to $T_{sum}$.

**Assumption 3.3** *Each model function $f_j = g_q \circ g_{q-1} \ldots \circ g_1 \circ g_0$ where each $g_i$ is a $\beta$-Hölder smoothness function.*

This assumption guaranteed the smoothness of the model function which give the neural network capability of approximate the true model accurately. The detailed definition of $\beta$-Hölder smoothness function will be in the supplement section 10.1.

The proposed change point detection algorithm is a two-step procedure, wherein the first step the Error Criterion function $E(t)$ is computed for each time point t from 1 to $T_{sum}$ and in the second step, its maximum point is identified. We will present this algorithm in next two subsections 3.1 and 3.1 for both single change point case and multiple change points case.

5

## 3.1 Single change point

It would always be easier to detect the change point if there is only one. To provide intuition, consider the following simple univariate "signal plus noise" model,

$$y_i = \alpha I(i < \tau) + \beta I(i \geq \tau) + \epsilon_i \qquad (5)$$

with a single change point $\tau$. Before the change point, the output $y_i$ takes the value $\alpha = 1$ corrupted by random noise $\epsilon_i$, while after the change point it becomes $\beta = 2$ also corrupted by random noise $\epsilon_i$. The total observation interval is of length $T_{sum} = 1000$ and the change point is located in the middle of the interval at $\tau = 500$. The generated data with the signal levels $\alpha$ and $\beta$ are shown in the left panel of Figure 1. Suppose that a simple feed-forward neural network is trained on the interval (100,200) with training window size $T_1 = 100$. The error function is then evaluated over a testing window of length $T_2 = 100$ from 201 to 300 at selected time points over the observation interval, which in this case is 200, 400, and so forth until a change point is detected. The spacing between the evaluation points is a tuning parameter of the algorithm. It can be seen that at $t = 200$, the value of the error function E(t) is low as depicted by the first star in the figure 1. The results are similar for the windows starting at $t = 300$ and 400 indicated by the second and third stars in the plot. Since the signal changes at $t = 500$, while $\hat{f}$ is trained on the first model, while the testing window spans on the interval 500 to 600, the Error criterion will exhibit a large value, since the responses $Y_i$ are generated by the second model, while the fitted values are based on the first model. Later at time point 600, the neural network is trained on data from the 500-600 observations which are from the second model and the Error criterion is evaluated also on the second model and will drop to a low level depicted by the fifth star.

The first step of the algorithm generating the error function E(t) is presented in algorithm 1. Given a time point t, let $X[t - T_1 : t], Y[t - T_1 : t]$ be the input and output from time $t - T_1$ to $t$ and with the dimension $T_1 \times p, T_1 \times h$ and a feed forward network $\hat{f}_\theta$ is fitted in this input and output and $E(t)$ is calculated by the test error in time period between $t$ and $t + T_2$.
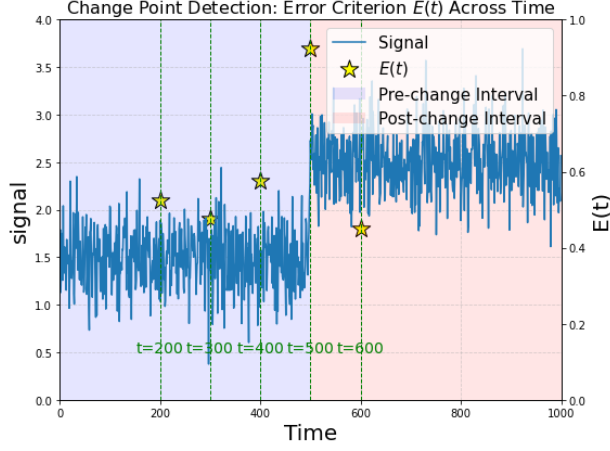
6

Figure 1: Example for generating E(t)

---

**Algorithm 1** Test Error Generator

---

1: **System Initialization**
2: Input data $X_t$, prediction $Y_t = f_i(X_t) + \epsilon_t$, $i = 1, 2, \ldots$, training window size $T_1$, test window size $T_2$, maximum training step $T$, neural network function $\hat{f}_\theta$ with trainable parameter $\theta$.
3: **for** $t = T_1, 2, \ldots, T$ **do**
4:  Compute the loss:

$$\mathcal{L}_\theta = \|Y[t - T_1 : t] - \hat{f}_\theta(X[t - T_1 : t])\|^2$$

5:  **while** $\theta$ not converged **do**
6:    Update parameters: $\theta \leftarrow \text{ADAM}(\mathcal{L}_\theta)$
7:  **end while**
8:  Record test error:

$$E[t] = \|Y[t : t + T_2] - \hat{f}_\theta(X[t : t + T_2])\|^2$$

9: **end for**
10: **Output** test error function $E(t)$

---

The detection step for single change point case is simple, we can select the time point t that maximize the function $E(t)$ as the estimated change point which is $t = 500$ in this example. The consistency of the estimation can be guaranteed by theorem 4.1.
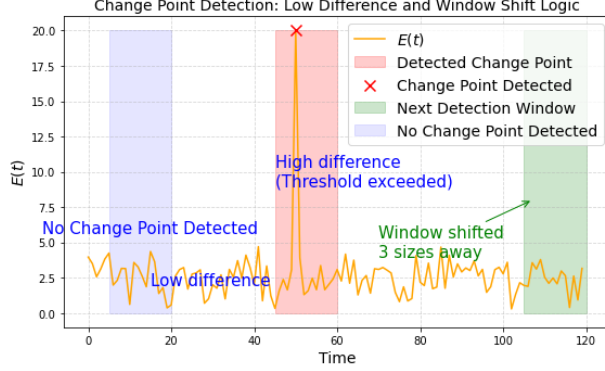
7

Figure 2: Example for detection process

## 3.2 Multiple Change points

The detection starts to get complicated when there are unknown number of multiple change points in the given data. Then the selection of window size become important because the algorithm will miss some change points if they are in the same training or testing window. And the detection algorithm need to be modified as well. We introduce a detection window with the tuning parameter detection window size $T_3$ which also need to be carefully selected for the same reason. We also introduce detection threshold $\pi$ as we need to differentiate the cases when detection window contains change point and when it does not. The detailed selection strategies of these window size and detection threshold will be discussed in the section 3.3 and 4.

In modified detection step, for each time point t, we select a detection window $(t - T_3, t + T_3)$ span at the center of this point with a designated detection window size and calculate the difference between the maximum and minimum values of $E(t)$ within this window. If the difference exceeds a fixed threshold $\pi$ (appropriately selected), we detect one change point and the location will be the time point maximizing the $E(t)$ within this detection window and the detection window can be moved to the next location without overlapping with previous detection window. The technical results established in the next section 4, provide guarantees of how well estimated the true change point is and also provide guidance of how to select the various window sizes required in steps 1 and 2 and of the detection algorithm.

For example, consider an existing test error with one peak at time point 50 in the figure 2, the peak threshold $\pi = 4$, and fix the detection window size $T_3 = 15$. At the time point before 44, within the detection window, the difference between the maximum and minimum of the error function is less than $\pi$ as shown by the first blue arrow, so we move to the next time point. At the time point between 44 and 54, which is near the peak, the difference will, however, exceed the threshold as displayed by the second blue arrow, and

then we will set the time point maximizing the test error within the window to be the detected change point. Once upon detecting one change point, we will move on to the time point where the left boundary for the detection window spanned is one window size after the right boundary for the detection window. This strategy will prevent detecting the same change point multiple times which is discussed in the next section.

The detailed strategy is described in algorithm 2

---

**Algorithm 2** Multiple Change Point Detection

---
1: **System Initialization**
2: Input error function $E(t)$, detection window size $T_3$, $n = 1$, detection threshold $\pi$, total time points $T_{\text{sum}}$.
3: **for** $t = 1, 2, \ldots, T_{\text{sum}}$ **do**
4:    Calculate

$$c_t = \max_{\max(t-T_3,0) \leq i,j \leq \min(t+T_3,T_{\text{sum}})} |E(i) - E(j)|$$

5:    **if** $c_t \geq \pi$ **then**
6:       $\hat{\tau}_n = \arg\max_{\max(t-T_3,0) \leq i,j \leq \min(t+T_3,T_{\text{sum}})} E(i)$
7:       $n = n + 1$
8:       $t = t + 3T_3$
9:    **end if**
10: **end for**
11: **Output:** $\hat{\tau}_1, \ldots, \hat{\tau}_N$.

---

## 3.3 Selection of the window size and detection threshold

Given model defined in equation 1 with total time points $T_{sum}$, the single change point case is simple as we can directly select $t = arg\max_t E(t)$. However, when it comes to multiple change points case, it is essential to select the appropriate window size for training, testing, and detection purposes. To simplify the selection process, we can set the training, testing and detection window size $T_1 = T_2 = T_0, T_3 = 2T_0$ for some window size $T_0$. Similar to the assumption that ensure distance between change point and the boundary, we also require the distribution of the change points within the total time points $T_{sum}$ not to be too dense. In other words, there should be a lower bound of the distance between change points that ensure there will not be multiple change points in the same window and the neural network can have enough time points to get optimized. Then the window size $T_0$ controls upperbound of the number of change points we can detect and the number $N \leq \frac{T_{sum}}{T_0}$. The details will be discussed further in theorems 4.2,4.3 and 4.4. Given the datasets with appropriate distribution of change points location, for independent and identically distributed data with finite variance, a good choice for window size is $T_0 = \sqrt{T_{sum}}$. For datasets with higher order moments in output space or subgaussian distributed, a narrower

window size $T_0$ can be used. On the other hand, for time-dependent data sets, a larger window size is required (see details in Section 4). The number of change points can be detected will also change according to the required window size. In general, when the selection of window size can be smaller, the distribution of change points can be denser and vice versa. In multiple change point case, We also need a large enough detection threshold $\pi$ to differentiate the detection window with and without change point. The threshold should be less than the minimum of model change signal $M1_*$, we use $\pi = \frac{1}{3}M1_*$ here. And this indicate another assumption that the $M1_*$ need to be large enough to be differentiated from the random error when there is no change point in the detection window. The detailed configuration will be discussed in 4.2 in Section 4.

## 4    Theoretical Analysis

To obtain guarantees for the estimated change point locations, results in [30, 31] for independent and dependent data are leveraged regarding how well the estimated feed-forward network approximates the true underlying $f_j$.

Further, a key assumption required is that difference in successive models $\mathbb{E}[||f_j(X) - f_{j-1}(X)||^2]$ should be large enough; otherwise, there will not be adequate signal to detect the corresponding change point.

The next result focuses on establishing properties for the proposed detection algorithm.

**Theorem 4.1** *Consider Single change point cases in equation 1 and assumption 3.1 and 3.3 hold. Further, consider the following Error criterion: $E(t) = \sum_{i=t}^{t+T_0} ||\hat{f}(X_i) - Y_i||^2$, with $T_0$ being the sample size and test size with the neural network $\hat{f}$ trained on $t - T_0 \leq i \leq t$.*
*Then, there exists a $t_0$, such that $E(t_0) - E(t) > \pi$ defined in section 3 for all t. Further, let $\hat{\tau} = argmax_t E(t)$ and $\tau$ denoting the true change point. Then, there exists a constant $C$, such that*

$$|\hat{\tau} - \tau| \leq \frac{CT_0}{M_1 - 2h\sigma^2} \tag{6}$$

*with probability*

$$1 - \frac{C_l}{T_0^{l-1}} \tag{7}$$

*for some constant $C_l$ based on l if $\mathbb{E}[||f_i(X)||^l] < \infty, l \geq 2$*
*and probability*

$$1 - C_1 e^{-C_2 T_0} \tag{8}$$

*for some constant $C_1$ and $C_2$, if the $\hat{f}, f_j$ is subgaussian for data X.*

The theorem 4.1 establishes that step 1 of the detection algorithm will identify a detectable peak near the true change point $\tau$. In addition, the estimated change point $\hat{\tau}$ detected in step 2 of the algorithm will be "close" to the true

10

change point. Hence, And so once detecting the true change point, the next detection window which will be one detection window size away from the previous one will not contain the previous true change point. As a result, one change point will only be detected for one time.

Next, we consider the case of multiple change points and provide probabilistic guarantees for the proposed detection algorithm regarding the estimated number and locations of them.

**Theorem 4.2** *Consider independent and identically distributed input data $X$ and consider the model in equation* (1), *and assumptions 3.1, 3.2 and 3.3 hold Let $\mathbb{E}[||f_i(X)||^l] < \infty$ for $l > 2$. Choose a constant $\rho \in (0,1)$ and define function in assumption 3.2 $b(T_{sum}) = \sqrt[\rho(l-1)]{T_{sum}}$*

*Then, the algorithm 1 and 2 with $T_0 = T^\rho = (\frac{C_0}{M1_*})^\rho \sqrt[l-1]{T_{sum}}$ for large $l$, detection window size $2T_0$ and detection threshold described in section 3, as $T_{sum} \to \infty$. satisfies*

$$P\left(\hat{N} = N, \max_{i=1,\dots,N} M1_*|\tau_i - \hat{\tau}_i| < C\frac{M1_*}{M1_* - 2h\sigma^2}T_{sum}^{\frac{1}{l-1}}\right) \to 1 \qquad (9)$$

This theorem establishes that, with high probability, the number of detected change points will match the number of true change points. Additionally, distances between the estimated change points and the true change points can be bounded with high probability as well. Notably, the theorem assumes a lower bound on the spacing between consecutive change points. This assumption is critical for determining the maximum number of detectable change points within the given framework, given the total number of observed time points $T_{sum}$.

For sub-Gaussian datasets, this theorem can be extended to provide a narrower error bound for the distances between the estimated and true change points. Additionally, it enables the detection of a denser and more change points given the total number of time. This extension leverages the properties of sub-Gaussian input data to refine the theoretical guarantees and enhances the performance of the detection algorithm.

**Theorem 4.3** *Consider independent and identically distributed input data $X$ and the model defined in equation* (1) *with subgaussian output and assumptions 3.1, 3.2 and 3.3 hold Choose a constant $\rho \in (0,1)$ and define function in assumption 3.2 $b(T_{sum}) = \sqrt[\rho]{log(T_{sum})}$*

*then with the algorithm 1 and 2 and select $T_0 = T^\rho = (\frac{C_0}{M1_*})^\rho \log(T_{sum})$, detection window size $2T_0$ and detection threshold described in section 3, as $T_{sum} \to \infty$.*

$$P\left(\hat{N} = N, \max_{i=1,\dots,N} M1_*|\tau_i - \hat{\tau}_i| < C\frac{M1_*}{M1_* - 2h\sigma^2}log(T_{sum})\right) \to 1 \qquad (10)$$

In many applications, the datasets may not be independently distributed and may exhibit temporal dependencies. Then, the extension of the theorem to this case is given below.

**Theorem 4.4** *Consider dependent and identically distributed input data $X$ and the model defined in equation* (1) *and assumptions 3.1, 3.2 and 3.3 hold Let $\mathbb{E}[||f_i(X)||^2] < \infty$ and $|cov(f_i, f_j)| < c < 1, i \neq j$ for some constant c.*

*Choose two constant $\rho, g \in (0, 1)$ and define function in assumption 3.2 $b(T_{sum}) = \sqrt[g\rho+1]{T_{sum}}$*

*Then, with the algorithm 1 and 2 and select $T_0 = T^\rho = (\frac{C_0}{M1_*})^\rho (T_{sum})^{\frac{\rho}{g\rho+1}}$, detection window size $2T_0$ and detection threshold described in section 3, as $T_{sum} \to \infty$.*

$$P\left(\hat{N} = N, \max_{i=1,\ldots,N} M1_*|\tau_i - \hat{\tau}_i| < C\frac{M1_*}{M1_* - 2h\sigma^2}T_{sum}^{\frac{\rho}{1+g\rho}}\right) \to 1 \qquad (11)$$

## 4.1 Remarks

In the assumptions of theorems 4.2,4.3 and 4.4, different characteristics of the data sets requires different lower bounds of the minimum distances between true change points or the density of the change points. For the independent datasets with the finite $l-$momentum of the model output with sufficiently large l, the lower bound of minimum distances for consecutive change points can scale as $T_{sum}^{\frac{1}{\rho(l-1)}}$. Consequently, the number of detectable change points, given a fixed $T_{sum}$ scales as $T_{sum}^{1-\frac{1}{\rho(l-1)}}$. For sub-Gaussian data sets, the lower bound can approach the scale of $\log T_{sum}$ enabling the detection of a greater number of change points for a fixed $T_{sum}$. In contrast, for dependent input data, the lower bound is not less than $T_{sum}^{\frac{1}{2}}$ and limiting the number of detectable change points to at most $\sqrt{T_{sum}}$.

# 5 Experiments

In this section, we present several experiments to illustrate the performance of the proposed algorithm and examine the sensitivity of the results based on the posited assumptions underlying the theoretical results.

## 5.1 Synthetic Data

We focus on experiments with synthetic data and examine the following three settings: the first two involve responses and predictors $(Y_i, X_i)$ generated by different nonlinear mechanisms, while the third one time series data, where the predictors $X_i$ correspond to past lags of the response variable.

Following the regression model in Equation (1), the input data $X_i$ is generated from i.i.d normal distribution N(0,I) and the dimension of input $X_i$ is 400 and the dimension of output $Y_i$ is 200. The error term $\epsilon_i$ is generated from i.i.d normal distribution N(0,$I\sigma^2$) where I is the identity matrix and $\sigma = 0.4$.

For the feed-forward network data, Each $f_j$ model used a neural network with three layers. In each layer, the fully connected weight matrix is given by $W_i + W_{ij}$ with no bias vector and activated by the ReLU function. Here $W_i$

is a fixed matrix for each model function in layer $i$ and $W_{ij}$ is a sparse matrix generated uniquely for each model function $f_j$ in layer $i = 1, 2, 3$. The dimension of the first layer matrix $W_1$ and $W_{1j}$ is $400 \times 100$, the dimension of the second layer matrix $W_2$ and $W_{2j}$ is $100 \times 100$, and the dimension of the third layer matrix is $100 \times 200$. Each element in $W_i$ is generated from normal distribution N(0,1), while in $W_{ij}$ is generated by a sparse distribution with nonzero element following the normal distribution: Sparse(0.1)*N(0,1) where Sparse(0.1)=1 with probability 0.1 and 0 with probability 0.9.

Each $f_j$ in the Random forest data is generated by a different random forest model with input dimension 400 and output dimension 200. The generating process of each random forest model $f_j$ is: Given input $X_i$ and a nonlinear function $g : \mathbb{R}^{400} \to \mathbb{R}^{200}$ (Here in our experiments, we used the feed-forward network with weight matrix $W_i + W_{ij}$ defined in the previous paragraph as g). Let $\tilde{Y}_i = g(X_i)$ and random forest model $f_j = arg\min_f ||f(X_i) - g(X_i)||^2$. And similarly, the output $Y_i = f_j(X_i)$.

The third data set as previously mentioned, is generated from Vector Autoregression (VAR) model with $q$ lags, as follows:

$$Y_t = A_{1j}Y_{t-1} + ... + A_{qj}Y_{t-q} + \epsilon_t, \tau_{j-1} \leq t \leq \tau_j \tag{12}$$

where each $Y_t$ is a 200 dimensional vector and $A_{ij} \in \mathbb{R}^{200 \times 200}, 1 \leq i \leq q$ is the transition (regression coefficient) matrix. This model can be put into the form of the nonlinear regression model in (1) as follows: flatten the predictors (input) as $X_t = (Y_t, Y_{t-1}, ..., Y_{t-q+1}), \boldsymbol{\epsilon}_t = (\epsilon_t, ..., \epsilon_{t-q+1})$, we have $Z_t = B_j Z_{t-1} + \boldsymbol{\epsilon}_t$ where

$$B_j = \begin{bmatrix} A_{1j} & A_{2j} & \cdots & A_{qj} \\ I & \cdots & 0 & 0 \\ \vdots & \ddots & & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} \tag{13}$$

$I$ is the identity matrix with the same dimension as $A_{ij}$. We represent different model $f_j$ with different matrix $B_j$ and $B_j Z_{t-1} = f_j(Z_{t-1})$. Then this flattened VAR(q) model is one special case of the regression model with input to be $Z_{t-1}$ and output to be $Z_t$. The input is distributed dependently, then the consistency analysis is guaranteed by theorem 4.4. In this section with use the VAR(4) model and the coefficient matrices for different models are generated by the algorithm in paper [32].

Table 1 showed the performance of our algorithms on these baseline simulations. The neural network for all the experiments we run in the training step of the algorithm is a fully-connected network with 2 hidden layers with 256 units and ReLU activation function. The change points $\tau_j$ are simulated by a random sample function. The "distance between change points" means the range that random function used to generate the distance between the change points and N means the number of change points. For each setting of distance range and change point number N, we generate the data set for 10 repetitions and run the two-step algorithm. Then we demonstrate the performance of the algorithm by

13

Table 1: Several Basic experiments

| Model | Distance Between CPs | N | $\bar{d}_{\tau-\hat{\tau}}$ | $\bar{d}_{|N-\hat{N}|}$ | Proportion ($\hat{N}=N$) |
|---|---|---|---|---|---|
| Feed-Forward Network Data | 300–1500 | 30 | 17.68 | 0.0 | 1.0 |
| Random Forest | 300–1500 | 30 | 17.30 | 0.0 | 1.0 |
| VAR(4) Flatten | 900–1500 | 30 | 20.89 | 0.0 | 1.0 |

Table 2: Independent vs dependent input

| model | Distance between cps | N | $\bar{d}_{\tau-\hat{\tau}}$ | $\bar{d}_{|N-\hat{N}|}$ | Prop($\hat{N}=N$) |
|---|---|---|---|---|---|
| Independent input data | 200-300 | 30 | 18.76 | 0.3 | 0.7 |
| Dependent input data | 200-300 | 30 | 88.14 | 7.7 | 0.0 |

$\bar{d}_{\tau-\hat{\tau}} = \frac{1}{10}\sum_{i=1}^{10} d_i(\tau - \hat{\tau})$ where each $d_i(\tau - \hat{\tau}) = \frac{1}{N}\sum_{j=1}^{N}|\tau_j - \hat{\tau}_j|$ (Here $\tau_j, \hat{\tau}_j$ is the location of jth true change point and estimated jth change point and the we will choose the closest estimated change point to $\tau_j$ to be the estimated jth change point) for ith repetition, $\bar{d}_{|N-\hat{N}|} = \frac{1}{10}\sum_{i=1}^{10}|N - \hat{N}_i|$ where each $\hat{N}_i$ is the number of estimated change points for ith repetition and $\text{Prop}(\hat{N} = N)$ is the proportion of the 10 repetitions that we have $\hat{N}_i = N$.

In this section, we will also run simulations to illustrate the consistency of the locations and number of change points detected for different types of datasets. The first comparison is the independent vs dependent data input. The input for the independent setting is generated by i.i.d normal distribution $N(0,I)$ and dependent input is generated by VAR(1) model and normalize the two types of input to the same scale for each $X_i$, the model function $f_j$ is the same feed-forward network structure as displayed in the previous simulation for both model setting. To easily compare the performance of the algorithm under the two types of data input, we implement a narrow distance(200-300 time points) for change points.

We can observe from the results table 2 that the algorithm will get better performance(lower distance and higher proportion of correct detection) on the independent data input than dependent input which is illustrated by the theorem 4.2 and 4.4.

In theorems 4.2 and 4.3, we also demonstrate the difference of algorithm performance given different model output distribution. The table below is the comparison between Nonsubgaussian data vs subgaussian output. The model function for both data sets is generated by feed-forward network structure. Elements in $W_i$ for both data sets are still from $N(0,1)$ and elements $W_{ij}$ are sparse(0.1)*$N(0,1)$ for subgaussian and sparse(0.1)*Uniform(0,1) for nonsubgaussian data output. To get a justifiable result, we fix the signal for both data sets. For both two types of data models, we set the model change signal to be constant:$E_{p(X)}[||f_i(X) - f_{i-1}(X)||^2] = 50, i = 1, 2, ..., N$.

Table 3: Nonsubgaussian vs subgaussian

| model | Distance between cps | N | $\bar{d}_{\tau-\hat{\tau}}$ | $\bar{d}_{|N-\hat{N}|}$ | $\mathrm{Prop}(\hat{N}=N)$ |
|---|---|---|---|---|---|
| Non-subgaussian data | 200-300 | 60 | 29.76 | 3.5 | 0.2 |
| Subgaussian data | 200-300 | 60 | 15.95 | 0.2 | 0.8 |

Table 4: Precise train

| model | Distance between cps | N | $\bar{d}_{\tau-\hat{\tau}}$ | $T_0$ |
|---|---|---|---|---|
| First train | 300-1500 | 10 | 20.46 | 50 |
| Precise train | 300-1500 | 10 | 8.43 | 20 |

We can observe from this table 3 that the algorithm will get better performance given the subgaussian data output.

In some cases, we aim to estimate the change points with greater precision by minimizing the distance between the true change points and the estimated ones. For datasets with finite $l$-moment of the output for sufficiently large $l$, decreasing the test window size $T_0$ can improve the precision of change point localization. This behavior is demonstrated in Table 4 .The input, output, and model function remain the same as those in the feed-forward network setting described in the first simulation. From the table 4, it is significant that reducing the window size $T_0$ leads to more accurate estimation of change point locations.

Finally, in the theorem, the assumption that the signal should be significantly larger than the standard deviation of the error term $\sigma$ is indispensable, as demonstrated in the proof provided in the supplementary section. The Figure 3 illustrates the relationship between the magnitude of $\sigma$ and the algorithm's performance. This is evaluated through three metrics: the average distance between true and estimated change points $\bar{d}_{\tau-\hat{\tau}}$, the average difference between the number of detected and true change points $\bar{d}_{|N-\hat{N}|}$ and the proportion of instances where the exact number of change points is detected $\mathrm{Prop}(\hat{N}=N)$. We observe that as the magnitude of $\sigma$ increases, the performance of the algorithm deteriorates significantly. This highlights the importance of the assumptions regarding the signal's magnitude in ensuring the algorithm's effectiveness.

## 5.2 Real Data

We also present the results of applying our detection strategy to a variety of real-world datasets.

The first dataset consists of data from 43 individuals with bladder tumors, where each value represents the copy number of specific gene mutations for each segment of each individual. The dataset contains 2216 segments [33]. Using
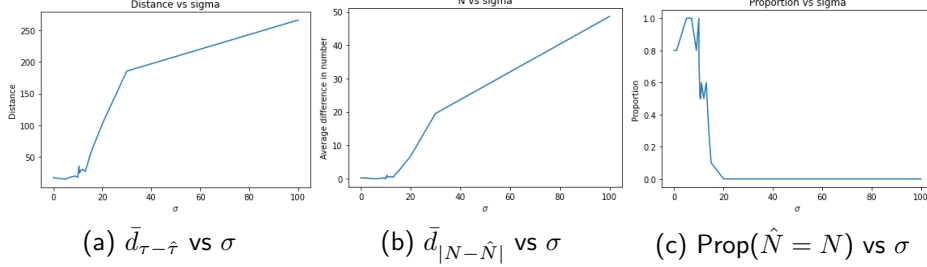
(a) $\bar{d}_{\tau - \hat{\tau}}$ vs $\sigma$      (b) $\bar{d}_{|N - \hat{N}|}$ vs $\sigma$      (c) Prop$(\hat{N} = N)$ vs $\sigma$

Figure 3: Performance metrics vs $\sigma$

the same detection strategy as in the VAR model, we apply a window size of $T_0 = 50 \simeq \sqrt{2216}$. Our two-step algorithm detected 17 change points in this dataset. Figure (a) in Figure 4 illustrates the number of mutation copies related to bladder cancer, with the detected change points marked by stars and vertical dashed lines.

The second dataset contains the log weekly returns for 29 stocks comprising the Dow Jones Industrial Average (DJIA). The dataset spans 1138 time points, from April 1990 to January 2012, where each time point represents one week [34].For this dataset, we use a window size of $T_0 = 30 \simeq \sqrt{1138}$. Our two-step algorithm successfully detected changes near the years 1998, 2000, 2002, and 2008, corresponding to significant events such as the 1997 financial crisis, the collapse of the technology bubble, the stock market downturn of 2002, and the financial crisis of 2008. Figure (c) in Figure 4 shows the log weekly returns for the 10th stock used to calculate the DJIA index, with detected change points highlighted.
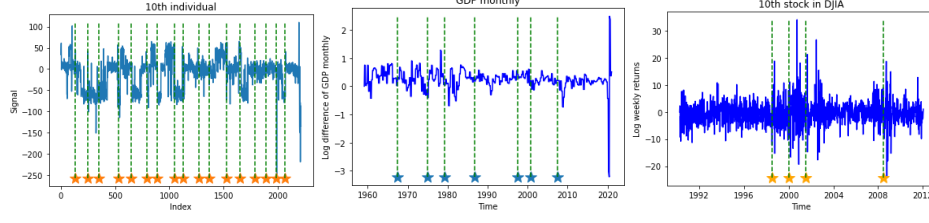
The third dataset, sourced from [35], consists of 743 time points, where each time point represents a month. The 1-dimensional output represents the log difference in monthly GDP, while the 122-dimensional input comprises various economic indicators, including stock market data, real personal income, and US treasury bond returns. This dataset spans several key recessions in U.S. economic history. Our algorithm detected change points corresponding to events such as the 1973 oil crisis, the 1980 recession, the dot-com bubble of 2000, and the great recession of 2008, among others.

## 5.3 Comparison with commonly used algorithms

We have also compared our algorithm with popular algorithms including KL-CPD in paper [28], Sparsified Binary Segmentation in paper [36] and double CUSUM in paper [37].

The data we used are generated from three different synthetical data models VAR(5), Non-linear VAR, and Lotka-Volterra model with multiple species.

The VAR(5) model is defined same as in section 5 equation (12) with dimension 200.

(a) The genes mutations of Cancer patients  (b) log difference of GDP  (c) DJIA Stock market index

Figure 4: Real data

Non-linear VAR model introduced in paper [38] is defined as:

$$x_t = Ax_{t-1} + \sum_{i=0}^{q} \Lambda_i \rho_i(f_{rt-i} + \epsilon_t) \tag{14}$$

where

$$f_t = \Gamma_1 f_{t-1} + \Gamma_2 f_{t-2} + \epsilon_t^f \tag{15}$$

Here $\Gamma_j, \epsilon_t^f$ is the coefficient matrix and error term for VAR(2) model $f_t$ and A and $\epsilon_t$ is the coefficient matrix and error term for Non-linear VAR model $x_t$. $\Lambda_i$ is the coefficient matrix for non-linear lag terms in Non-linear VAR model and $\rho_i$ is the corresponding non-linear function. In experiment, we set q=6, dimension of $f_t$ to be 10, dimension of $x_t$ to be 200. We also have the frequency parameter $r = 3$ here to skip direct connection between the consecutive components $f_t, f_{t-1}$.

Lotka-Volterra model introduced in paper [39] (displayed as LV in the table 5) with multiple species are generated by the following differential equations:

$$\frac{dx^i}{dt} = \alpha x^i - \beta x^i \sum_{j \in Pa(x^i)} y^j - \alpha(x^i)^2, 1 \le i \le p \tag{16a}$$

$$\frac{dy^j}{dt} = \delta y^j \sum_{k \in Pa(y^j)} x^k - \gamma(y^j), 1 \le j \le p \tag{16b}$$

where $x^i$ corresponds to the population sizes of prey species and $y^j$ is the population sizes of predator species, $\alpha, \beta, \delta, \gamma$ are the fixed parameters controlling strengths of interactions, and the $Pa(x^i), Pa(y^j)$ are the sets of Granger-causes of $x^i, y^j$.

Here we define 3 types of distance. The regular distance $d_{\tau-\hat{\tau}}$ we used in section 5, the sum version Hausdorff distance

$$d\_hau\_w = \max_{\tau_i \in T} \sum_{\hat{\tau}_i \in \hat{T} | \tau_i - \hat{\tau}_j | <= \min_k |\tau_k - \hat{\tau}_j|} |\tau_i - \hat{\tau}_j| \tag{17}$$

17

the product version Hausdorff distance

$$d\_hau\_prod = \max_{\tau_i \in T} \prod_{\hat{\tau}_i \in \hat{T} | \tau_i - \hat{\tau}_j | <= \min_k |\tau_k - \hat{\tau}_j|} |\tau_i - \hat{\tau}_j| \qquad (18)$$

where $T, \hat{T}$ are the set for true change points and detected change points. For the comparison, the precision and recall will also be presented. We use these two modified versions of Hausdorff distances because many algorithms can get decent results in distance and accuracy when detecting change points. But they usually suggest multiple estimation around the true change point and it is not applicable in real world when we don't know which estimated change point should be selected finally. By introducing these new metrics, we can penalize such behavior and make sure one algorithm will not propose too many estimations around the true change points when it gets good performance in these metrics and we can get a better understanding of the performance and robustness of different change point detection algorithms.

In the experiments, the distances between change points are randomly generated from the uniform distribution ranging from 1200 to 1500.

The results of the comparison experiments are displayed in the table 5. In the table, the five metrics are displayed for every combination of detection methods and datasets. Our methods beat the 3 competitive methods in two of the datasets except for Nonlinear VAR data where Double Cusum has better performance under some metrics.

# 6    Conclusion and Discussion

In this paper, we introduced a two-step algorithmic framework for offline change point detection using feed-forward neural networks. This approach effectively combines a training step for estimating model parameters and a detection step for identifying change points based on an error criterion. We established theoretical guarantees for the number and locations of change points detected by our method, ensuring its reliability under specific assumptions. These guarantees include precise bounds on the distances between estimated and true change points, as well as the conditions under which the detected change points accurately match the true ones.

Our extensive experiments validated the performance of the algorithm across various simulated and real-world datasets. For instance, in simulated datasets based on feed-forward networks, random forests, and vector autoregressive models, our algorithm demonstrated robust detection capabilities under diverse settings. Moreover, experiments with real-world data—including genetic mutation copy number datasets, stock market returns for the Dow Jones Industrial Average (DJIA), and GDP-related economic indicators—further highlighted the applicability and accuracy of the proposed approach. These results showcased the algorithm's ability to detect meaningful change points in practical scenarios, such as identifying shifts during economic recessions or significant market events.

Table 5: Comparison of Methods on Various Models

| Method | Data | $\bar{d}_{\tau-\hat{\tau}}$ | d_hau_w | d_hau_prod | precision | recall |
|---|---|---|---|---|---|---|
| ML model | Nonlinear VAR | 118.16 | **134.0** | 31.7 | 0.675 | 0.70 |
| Sparsified Binary Segmentation | Nonlinear VAR | 157.94 | 400.2 | 284.5 | 0.362 | **0.975** |
| Double Cusum | Nonlinear VAR | **90.00** | 130.20 | **8.9** | **0.704** | 0.925 |
| KLCPD | Nonlinear VAR | 499.36 | 302.7 | 1108.5 | 0.09 | 0.05 |
| ML model | LV | **87.84** | **132.90** | **48.2** | **0.58** | **0.75** |
| Sparsified Binary Segmentation | LV | 480.84 | 391.0 | 5803.7 | 0.11 | 0.65 |
| Double Cusum | LV | 561.25 | 850.0 | 273.70 | 0.0 | 1.0 |
| KLCPD | LV | 1601.09 | 209.4 | inf | 0.0 | 0.0 |
| ML model | VAR | **20.05** | **32.5** | **32.5** | **0.90** | **0.875** |
| Sparsified Binary Segmentation | VAR | 480 | 391.0 | 5803.7 | 0.11 | 0.65 |
| Double Cusum | VAR | 561.25 | 850.0 | 273.70 | 0.0 | 1.0 |
| KLCPD | VAR | 862.64 | 284.0 | inf | 0.09 | 0.1 |

In this paper, we also compare our algorithm with methods including KL-CPD and Sparsified Binary Segmentation on different multivariate scenarios.

While the detection strategy is generalizable to other neural network architectures, such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), and long short-term memory networks (LSTMs), the current theoretical framework does not extend to these architectures. Future work could address this limitation by establishing theoretical guarantees for these network types, potentially broadening the scope of applications.

Another promising direction for future research involves refining the method for dependent data cases. Specifically, by incorporating assumptions such as martingale properties or Markov chain dependencies, the algorithm could achieve more precise change point detection with reduced window sizes. This adjustment would enhance its applicability to time series data with complex dependencies, such as financial or biological systems.

Additionally, exploring the scalability of the algorithm to larger datasets or online scenarios represents another avenue for advancement. Optimizing computational efficiency and adapting the algorithm for real-time detection could make it suitable for applications requiring immediate responses, such as anomaly detection in streaming data or high-frequency financial data analysis.

In summary, the proposed two-step framework not only offers a theoretically grounded and experimentally validated solution for offline change point detection but also paves the way for future research to expand its versatility and

robustness in broader contexts.

# 7  Annotation and Assumption

In this supplement section, we will provide proof of the theorems in the consistency analysis section.

Before heading into the proof of the main theorems, as we said in theoretical analysis section4, results in [30, 31] for independent and dependent data are leveraged regarding how well the estimated feed-forward network approximates the true underlying $f_j$. The first section of the supplement will be focused on the notations, assumptions, and the lemmas we need for the main proof. The second section will be mainly on the proof of theorem 4.1 and the third section will be proof of theorem 4.2, 4.3, and 4.4.

# 8  Proof

## 8.1  Proof of Theorem4.1

In this section, the proof of subgaussian and dependent cases will be covered in the next section for multiple change point cases. Also, the proof of existence for the peak in E(t) will be covered in the next section. Moreover, the lemma above keeps the output of the neural network to be one dimension. In our algorithm, the output, however, is the h dimension. To mitigate this difference, we consider each feed-forward network we use as $\hat{f} = (\hat{f}_1, ..., \hat{f}_h)$ where each $\hat{f}_k \in \mathcal{F}(L, \boldsymbol{p}, s, F)$ represents a network defined the same as in the lemma and similarly $f_0 = (f_0^1, ..., f_0^h)$ where $f_0^k \in \mathcal{G}(q, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta}, K)$. Then

$$\mathbb{E}_{f_0}[||\hat{f}(X) - f_0(X)||^2] = \sum_{k=1}^{h} \mathbb{E}_{f_0}[(\hat{f}_k(X) - f_0^k(X))^2] \tag{19}$$

where each term in the sum is controlled by the lemma 10.1 and the sum will also be controlled with multiplication by h.

For the single change point case in theorem4.1, $\tau$ is the estimated change point where $\tau = arg\max_t(Error(t))$. Here we first assume $T_0 \gg C|\tau - \tau^*|$ for some constant C.

If $\tau^* > \tau$:

$$E(\tau) - E(\tau^*) = \sum_{i=\tau}^{\tau+T_0} ||\hat{f}_\tau^1(X_i) - Y_i||^2 - \sum_{i=\tau^*}^{\tau^*+T_0} ||\hat{f}_{\tau^*}^2(X_i) - Y_i||^2 \geq 0 \tag{20}$$

Where $\hat{f}_\tau^1 \in arg\min_{f \in \mathcal{F}(L,p,s,F)} \sum_{i \leq \tau}(Y_i - f(X_i))^2$ and $\hat{f}_{\tau^*}^2 \in arg\min_{f \in \mathcal{F}(L,p,s,F)} \sum_{i \leq \tau^*}(Y_i - f(X_i))^2$

And then

$$\sum_{i=\tau}^{\tau^*}||\hat{f}_\tau^1(X_i)-Y_i||^2+\sum_{i=\tau^*}^{\tau+T_0}||\hat{f}_\tau^1(X_i)-Y_i||^2-\sum_{i=\tau^*}^{\tau+T_0}||\hat{f}_{\tau^*}^2(X_i)-Y_i||^2-\sum_{i=\tau+T_0}^{\tau^*+T_0}||\hat{f}_{\tau^*}^2(X_i)-Y_i||^2\geq 0$$

$$\sum_{i=\tau}^{\tau^*}||\hat{f}_\tau^1(X_i)-Y_i||^2+\sum_{i=\tau^*}^{\tau+T_0}(||\hat{f}_\tau^1(X_i)-Y_i||^2-||\hat{f}_{\tau^*}^2(X_i)-Y_i||^2)-\sum_{i=\tau+T_0}^{\tau^*+T_0}||\hat{f}_{\tau^*}^2(X_i)-Y_i||^2\geq 0$$

$$(21)$$

Here let $I_1 = \sum_{i=\tau}^{\tau^*}||\hat{f}_\tau^1(X_i) - Y_i||^2$, $I_2 = \sum_{i=\tau^*}^{\tau+T_0}(||\hat{f}_\tau^1(X_i) - Y_i||^2 - ||\hat{f}_{\tau^*}^2(X_i) - Y_i||^2)$ and $I_3 = \sum_{i=\tau+T_0}^{\tau^*+T_0}||\hat{f}_{\tau^*}^2(X_i) - Y_i||^2$
For $I_1$:

$$I_1 \leq \sum_{i=\tau}^{\tau^*}(||\hat{f}_\tau^1(X_i) - f_0(X_i)||^2 + ||Y_i - f_0(X_i)||^2)$$

$$\leq (\tau^* - \tau)\mathbb{E}_{f_0}[(\hat{f}_\tau^1(X) - f_0(X)^2] + o(\frac{1}{\tau^* - \tau}) + \sum_\tau^{\tau^*}||Y_i - f_0(X_i)||^2$$

$$\stackrel{Theorem 10.1}{\leq} C_1\phi_T L\log^2 T(\tau^* - \tau) + h\sigma^2(\tau^* - \tau) \quad (22)$$

For $I_2$:

$$I_2 = \sum_{i=\tau^*}^{\tau+T_0}(||\hat{f}_\tau^1(X_i) - Y_i||^2 - ||\hat{f}_{\tau^*}^2(X_i) - Y_i||^2) \overset{\text{Cauchy Inequity}}{\leq}$$

$$\sum_{i=\tau^*}^{\tau+T_0}(||\hat{f}_\tau^1(X_i) - \hat{f}_{\tau^*}^2(X_i)||)(||\hat{f}_\tau^1(X_i) + \hat{f}_{\tau^*}^2(X_i) - 2Y_i||)$$

$$\overset{\text{Cauchy Inequity}}{\leq} \sqrt{T_0 E_{f_1}[||\hat{f}_\tau^1(X) - \hat{f}_{\tau^*}^2(X)||^2] + o(\frac{1}{T_0})}$$

$$\sqrt{\left(\sum_{i=\tau^*}^{\tau+T_0} ||\hat{f}_\tau^1(X_i) - f_0(X_i)||^2 + ||\hat{f}_{\tau^*}^2(X_i) - f_0(X_i)||^2 + 2||Y_i - f_0(X_i)||^2\right)}$$

$$\overset{Theorem 10.1}{\leq} \sqrt{C_2 T_0 \phi_T L \log^2 T}$$

$$\sqrt{\left(\sum_{i=\tau^*}^{\tau+T_0} ||\hat{f}_\tau^1(X_i) - f_0(X_i)||^2 + ||\hat{f}_{\tau^*}^2(X_i) - f_0(X_i)||^2 + 2||Y_i - f_0(X_i)||^2\right)}$$

$$\overset{Minkovsky Inequity}{\leq} \sqrt{C_2 T_0 \phi_T L \log^2 T}$$

$$(\sqrt{\sum_{i=\tau^*}^{\tau+T_0} ||\hat{f}_\tau^1(X_i) - f_0(X_i)||^2 + ||\hat{f}_{\tau^*}^2(X_i) - f_0(X_i)||^2}$$

$$+ \sqrt{\sum_{i=\tau^*}^{\tau+T_0} 2||Y_i - f_1(X_i)||^2 + 2||f_0(X_i) - f_1(X_i)||^2})$$

$$\leq \sqrt{C_2 T_0 \phi_T L \log^2 T}(\sqrt{2T_0 \mathbb{E}_{f_1}[||\hat{f}_{\tau^*}^2(X) - f_0(X)||^2] + o(\frac{1}{T_0})}$$

$$+ \sqrt{\sum_{i=\tau^*}^{\tau+T_0} 2||Y_i - f_1(X_i)||^2 + 2||f_0(X_i) - f_1(X_i)||^2})$$

$$\overset{Thoerem 10.1}{\leq} \sqrt{C_2 T_0 \phi_T L \log^2 T}\left(\sqrt{2C_2 T_0 \phi_T L \log^2 T} + \sqrt{\sum_{i=\tau^*}^{\tau+T_0} 2T_0 h\sigma^2 + 2M_2 T_0 + o(\frac{1}{T_0})}\right)$$

$$\leq 2C_2 T_0 \phi_T L \log^2 T + \sqrt{2C_2 \phi_T L \log^2 Th T_0}\sigma$$

$$+ \sqrt{2C_2 \phi_T L \log^2 T M_2 T_0} \leq C_4(M_2, C_2, L) T_0 \phi_T \log^2 T \quad (23)$$

Where $C_4$ is a constant given $M_2, C_2$ and L.

For $I_3$:

$$I_3 = \sum_{i=\tau+T_0}^{\tau^*+T_0} ||\hat{f}^2_{\tau^*}(X_i) - Y_i||^2 \geq \sum_{i=\tau+T_0}^{\tau^*+T_0} (||f_0(X_i) - f_1(X_i)||^2$$
$$- ||\hat{f}^2_{\tau^*}(X_i) - f_0(X_i)||^2 - ||f_1(X_i) - Y_i||^2)$$
$$\geq (\tau^* - \tau)\mathbb{E}_{f_0}[(f_0(X) - f_1(X))^2] - (\tau^* - \tau)\mathbb{E}_{f_1}[(\hat{f}^2_{\tau^*}(X) - f_1(X))^2]$$
$$- (\tau^* - \tau)\mathbb{E}[\epsilon^2] + o(\frac{1}{\tau^* - \tau})$$
$$\overset{Theorem 10.1}{\geq} (\tau^* - \tau)(M_1 - h\sigma^2 - C_3\phi_T L \log^2 T) \quad (24)$$

By the previous equation $I_1 + I_2 \geq I_3$

$$C_1\phi_T L \log^2 T(\tau^* - \tau) + h\sigma^2(\tau^* - \tau) + C_4(M_2, C_2, L)T_0^2\phi_T \log^2 T$$
$$\geq (\tau^* - \tau)(M_1 - h\sigma^2 - C_3\phi_T L \log^2 T)$$
$$|\tau^* - \tau| \leq \frac{C_4(M_2, C_2, L)T_0\phi_T \log^2 T}{M_1 - 2h\sigma^2 - (C_1 + C_3)\phi_T L \log^2 T} \quad (25)$$

Similarly, if $\tau > \tau^*$:

$$Error(\tau) - Error(\tau^*) = \sum_{i=\tau}^{\tau+T_0} ||\hat{f}^1_\tau(X_i) - Y_i||^2 - \sum_{i=\tau^*}^{\tau^*+T_0} ||\hat{f}^2_{\tau^*}(X_i) - Y_i||^2 \geq 0 \quad (26)$$

Where $\hat{f}^1_\tau \in arg\min_{f\in\mathcal{F}(L,p,s,F)} \sum_{\tau^*\leq i\leq \tau}(Y_i - f(X_i))^2$ and $\hat{f}^2_{\tau^*} \in arg\min_{f\in\mathcal{F}(L,p,s,F)} \sum_{i\leq\tau^*}(Y_i - f(X_i))^2$
And then

$$\sum_{i=\tau^*+T_0}^{\tau+T_0} ||\hat{f}^1_\tau(X_i) - Y_i||^2 + \sum_{i=\tau}^{\tau^*+T_0} ||\hat{f}^1_\tau(X_i) - Y_i||^2$$
$$- \sum_{i=\tau}^{\tau^*+T_0} ||\hat{f}^2_{\tau^*}(X_i) - Y_i||^2 - \sum_{i=\tau^*}^{\tau} ||\hat{f}^2_{\tau^*}(X_i) - Y_i||^2 \geq 0$$
$$- \sum_{i=\tau^*}^{\tau} ||\hat{f}^2_{\tau^*}(X_i) - Y_i||^2 + \sum_{i=\tau}^{\tau^*+T_0} (||\hat{f}^1_\tau(X_i) - Y_i||^2 - ||\hat{f}^2_{\tau^*}(X_i) - Y_i||^2)$$
$$+ \sum_{i=\tau^*+T_0}^{\tau+T_0} ||\hat{f}^1_\tau(X_i) - Y_i||^2 \geq 0 \quad (27)$$

Here let $I_1 = \sum_{i=\tau^*+T_0}^{\tau+T_0} ||\hat{f}^1_\tau(X_i) - Y_i||^2$, $I_2 = \sum_{i=\tau}^{\tau^*+T_0} (||\hat{f}^1_\tau(X_i) - Y_i||^2 - ||\hat{f}^2_{\tau^*}(X_i) - Y_i||^2)$ and $I_3 = \sum_{i=\tau^*}^{\tau} ||\hat{f}^2_{\tau^*}(X_i) - Y_i||^2$

For $I_1$:

$$I_1 \leq \sum_{i=\tau^*+T_0}^{\tau+T_0} (||\hat{f}_\tau^1(X_i) - f_1(X_i)||^2 + ||Y_i - f_1(X_i)||^2)$$

$$\leq (\tau^* - \tau)\mathbb{E}_{f_1}[(\hat{f}_\tau^1(X) - f_1(X)^2] + o(\frac{1}{\tau - \tau^*}) + \sum_{\tau^*+T_0}^{\tau+T_0} ||Y_i - f_1(X_i)||^2$$

$$\overset{Theorem 10.1}{\leq} C_1\phi_{\tau-\tau^*}L\log^2(\tau-\tau^*)(\tau-\tau^*)+h\sigma^2(\tau-\tau^*) \leq C_1\min_{i=1,...,L}p_i\log^2(\tau-\tau^*)+h\sigma^2(\tau-\tau^*)$$

$$\leq h\sigma^2(\tau - \tau^*) + (\tau - \tau^*)o(1) \quad (28)$$

For $I_2$:

$$I_2 = \sum_{i=\tau}^{\tau^*+T_0} (||\hat{f}_\tau^1(X_i) - Y_i||^2 - ||\hat{f}_{\tau^*}^2(X_i) - Y_i||^2)$$

$$\leq \sum_{i=\tau}^{\tau^*+T_0} (||\hat{f}_\tau^1(X_i) - f_1(X_i)||^2 + ||f_1(X_i) - Y_i||^2) - \sum_{i=\tau}^{\tau^*+T_0} ||\hat{f}_{\tau^*}^2(X_i) - Y_i||^2$$

$$\leq T_0\mathbb{E}_{f_1}[(\hat{f}_\tau^1(X) - f_1(X))^2] + T_0h\sigma^2 + o(\frac{1}{T_0})$$

$$- \sum_{i=\tau}^{\tau^*+T_0} ||(f_0(X_i) - f_1(X_i)) + (\hat{f}_{\tau^*}^2(X_i) - f_0(X_i)) + (f_1(X_i) - Y_i)||^2$$

$$\leq T_0\mathbb{E}_{f_1}[(\hat{f}_\tau^1(X) - f_1(X))^2] + T_0h\sigma^2$$

$$- \sum_{i=\tau}^{\tau^*+T_0} (||f_0(X_i) - f_1(X_i)||^2 - ||\hat{f}_{\tau^*}^2(X_i) - f_0(X_i)||^2 - ||f_1(X_i) - Y_i||^2)$$

$$\overset{Theorem 10.1}{\leq} T_0C_1\phi_{\tau-\tau^*}L\log^2(\tau - \tau^*) + 2T_0h\sigma^2 +$$

$$C_1\phi_T L\log^2(T) - T_0M_2$$

$$\leq T_0\min_{i=1,...,L}p_i\frac{log^2(\tau - \tau^*)}{\tau - \tau^*} + 2T_0h\sigma^2 - T_0M_2 + C_1\phi_T L\log^2(T) \quad (29)$$

For $I_3$:

$$I_3 = \sum_{i=\tau^*}^{\tau} ||\hat{f}_{\tau^*}^2(X_i) - Y_i||^2$$

$$= \sum_{i=\tau^*}^{\tau} ||(\hat{f}_{\tau^*}^2(X_i) - f_0(X_i)) + (f_0(X_i) - f_1(X_i)) + (f_1(X_i) - Y_i)||^2$$

$$\geq \sum_{i=\tau^*}^{\tau} (||f_0(X_i) - f_1(X_i)||^2 - ||\hat{f}_{\tau^*}^2(X_i) - f_0(X_i)||^2 - ||f_1(X_i) - Y_i||^2)$$

$$\geq (\tau - \tau^*)\mathbb{E}_{f_0}[(f_0(X) - f_1(X))^2]$$

$$- (\tau - \tau^*)\mathbb{E}_{f_1}[(\hat{f}_{\tau^*}^2(X) - f_1(X))^2] - (\tau - \tau^*)\mathbb{E}[\epsilon^2] + o(1)$$

$$\overset{Theorem 10.1}{\geq} (\tau - \tau^*)(M_1 - h\sigma^2 - C_3\phi_T L \log^2 T) \quad (30)$$

By the previous equation $-I_3 + I_2 + I_1 \geq 0$

$$(\tau - \tau^*)o(1) + h\sigma^2(\tau - \tau^*) + T_0 \min_{i=1,...,L} p_i \frac{\log^2(\tau - \tau^*)}{\tau - \tau^*}$$

$$+ 2T_0 h\sigma^2 + C_1\phi_T L \log^2(T) - T_0 M_2$$

$$\geq (\tau - \tau^*)(M_1 - h\sigma^2 - C_3\phi_T L \log^2 T)$$

$$|\tau - \tau^*| \leq$$

$$\frac{2T_0 h\sigma^2 + C_1\phi_T L \log^2(T) - T_0 M_2 + T_0 \min_{i=1,...,L} p_i \frac{\log^2(\tau-\tau^*)}{\tau-\tau^*}}{M_1 - 2h\sigma^2 - C_3\phi_T L \log^2 T - o(1)} \quad (31)$$

Here, if $|\tau - \tau^*| \geq \sqrt{T_0}$, there $\exists C_5$ constant, $T_0 \min_{i=1,...,L} p_i \frac{\log^2(\tau-\tau^*)}{\tau-\tau^*} \leq C_5 T_0$

$$|\tau - \tau^*| \leq \frac{2T_0 h\sigma^2 - T_0 M_2 + C_5 T_0 + C_1\phi_T L \log^2(T)}{M_1 - 2h\sigma^2 - C_3\phi_T L \log^2 T}$$

$$\leq \frac{C_5 T_0 + C_1\phi_T L \log^2(T)}{M_1 - 2h\sigma^2 - C_3\phi_T L \log^2 T} \quad (32)$$

Here if $|\tau - \tau^*| > CT_0$, if $\tau^* > \tau$.

$$E(\tau) - E(\tau^*) = \sum_{i=\tau}^{\tau+T_0} ||\hat{f}_{\tau}^1(X_i) - Y_i||^2 - \sum_{i=\tau^*}^{\tau^*+T_0} ||\hat{f}_{\tau^*}^2(X_i) - Y_i||^2$$

$$\overset{Theorem 10.1}{\leq} 2T_0 C\phi_T L \log^2 T + 2T_0 h\sigma^2 - M_1 T_0 < 0 \quad (33)$$

If $\tau > \tau^*$

$$E(\tau) - E(\tau^*) = \sum_{i=\tau}^{\tau+T_0} ||\hat{f}_{\tau}^1(X_i) - Y_i||^2 - \sum_{i=\tau^*}^{\tau^*+T_0} ||\hat{f}_{\tau^*}^2(X_i) - Y_i||^2$$

$$\overset{Theorem 10.1}{\leq} 2T_0 C\phi_{T_0} L \log^2(T_0) + 2T_0 h\sigma^2 - M_1 T_0 < 0 \quad (34)$$

A contradiction of $Error(\tau) \geq Error(\tau^*)$.

In all, there exists constant $C_5, C$, with probability $1 - \frac{C}{T_0}$

$$|\tau^* - \tau| \leq \frac{C_5 T_0}{M_1 - 2h\sigma^2} \tag{35}$$

According to the proof in section 8.1, the probability

$$P\left(|\sum_{i=\tau}^{\tau+T_0} ||\hat{f}_\tau^1(X_i) - f_1(X_i)||^2 - T_0 \mathbb{E}[||\hat{f}_\tau^1(X_i) - f_1(X_i)||^2]| > \delta\right) \leq \frac{C(\delta)}{T_0} \tag{36}$$

given $\forall \delta$, C is a constant depending on $\delta$. For other approximations in the proof, the technique is similar. This probability will be specified clearly in the proof of multiple change point cases in the section 8.2 given the different assumptions of the model and data.

## 8.2 Proof of theorems 4.2, 4.3 and 4.4

Let $T_0$ be the test window size. $\hat{\tau}_i$ denotes the $i^{th}$ estimate change point. By the proof of a single change point,

$$|\hat{\tau}_i - \tau_i| \leq \max\left(\frac{C_{f_i}(M2_i)T_0\phi_{|\tau_{i+1}-\tau_i|}\log^2|\tau_{i+1}-\tau_i|}{M1_i - 2h\sigma^2 - C\phi_{|\tau_{i+1}-\tau_i|}L\log^2|\tau_{i+1}-\tau_i|},\right.$$
$$\left.\frac{C_5 T_0 + C_1\phi_{|\tau_{i+1}-\tau_i|}L\log^2(|\tau_{i+1}-\tau_i|)}{M1_i - 2h\sigma^2 - C_3\phi_{|\tau_{i+1}-\tau_i|}L\log^2|\tau_{i+1}-\tau_i|}, \sqrt{T_0}\right)$$
$$\leq \max\left(\frac{C_{f_i}(M2_i)T_0\phi_T\log^2 T}{M1_i - 2h\sigma^2}, \frac{C_5 T_0 + C_1\phi_T L\log^2(T)}{M1_i - 2h\sigma^2}, \sqrt{T_0}\right) \leq C\frac{C_5}{M1_i - 2h\sigma^2}T_0 \tag{37}$$

By the lemma 10.5, with probability $1 - \frac{C}{T_0^{p-1}}$ if X is independent of each other and $1 - \frac{C}{T_0}$ if X is dependent of each other, the above inequity will hold.

If X variable is subgaussian and independent of each other, by the lemma 10.3, the probability will be $1 - Ce^{-C_1 T_0}$.

Inside the algorithm 2, define $T_0 = \sqrt[p]{T} = o(T)$, for each peak found, define the peak window as $C_6 T_0 = C\frac{C_5}{M1_* - 2h\sigma^2}T_0$ and $\pi = \frac{M1_*}{2} - 2h\sigma^2$ then inside the interval $[\min(t - C_6 T_0, 0), (t + C_6 T_0)]$ when detecting peak at time t, there is one and only one change point with high probability depending on the type of input and model because windows of the nearby two peaks detected $[\min(t' - C_6 T_0, 0), (t' + C_6 T_0)]$ which containing the relative true change points will not intersect with it. Also, the next detection window will be at least one detection window size $C_6 T_0$ away from the previous window, and will not detect the previous change point for the second time.

Then we need also to prove near each change point, the algorithm will detect a peak in the error function. Given any change point $\tau_i, 1 \leq i \leq N$. Let $\hat{f}_{\tau_i - 2T_0} \in$

$arg\min_{f\in\mathcal{F}(L,p,s,F)}\sum_{k=\tau_{i-1}}^{\tau_i-2T_0}||Y_k-f(X_k)||^2$, $\hat{f}_{\tau_i-T_0}\in arg\min_{f\in\mathcal{F}(L,p,s,F)}\sum_{k=\tau_{i-1}}^{\tau_i-T_0}||Y_k-f(X_k)||^2$, $\hat{f}_{\tau_i}\in arg\min_{f\in\mathcal{F}(L,p,s,F)}\sum_{k=\tau_{i-1}}^{\tau_i}||Y_k-f(X_k)||^2$, $\hat{f}_{\tau_i+T_0}\in arg\min_{f\in\mathcal{F}(L,p,s,F)}\sum_{k=\tau_i}^{\tau_i+T_0}||Y_k-f(X_k)||^2$ and $\hat{f}_{\tau_i+2T_0}\in arg\min_{f\in\mathcal{F}(L,p,s,F)}\sum_{k=\tau_i}^{\tau_i+2T_0}||Y_k-f(X_k)||^2$. To detect a peak near the true change point, the equation below holds.

$$\min\left(Error(\tau_i)-Error(\tau_i-T_0),Error(\tau_i)-Error(\tau_i+T_0)\right)$$
$$\gg\max\left(|Error(\tau_i-T_0)-Error(\tau_i-2T_0)|\right.$$
$$\left.,|Error(\tau_i+T_0)-Error(\tau_i+2T_0)|\right)\quad(38)$$

Which means

$$\min(\sum_{k=\tau_i}^{\tau_i+T_0}||Y_k-\hat{f}_{\tau_i}(X_k)||^2-\sum_{k=\tau_i-T_0}^{\tau_i}||Y_k-\hat{f}_{\tau_i-T_0}(X_k)||^2,$$
$$\sum_{k=\tau_i}^{\tau_i+T_0}||Y_k-\hat{f}_{\tau_i}(X_k)||^2-\sum_{k=\tau_i+T_0}^{\tau_i+2T_0}||Y_k-\hat{f}_{\tau_i+T_0}(X_k)||^2)$$
$$\gg\max(|\sum_{k=\tau_i-T_0}^{\tau_i}||Y_k-\hat{f}_{\tau_i-T_0}(X_k)||^2-\sum_{k=\tau_i-2T_0}^{\tau_i-T_0}||Y_k-\hat{f}_{\tau_i-2T_0}(X_k)||^2|$$
$$,|\sum_{k=\tau_i+T_0}^{\tau_i+2T_0}||Y_k-\hat{f}_{\tau_i+T_0}(X_k)||^2-\sum_{k=\tau_i+2T_0}^{\tau_i+3T_0}||Y_k-\hat{f}_{\tau_i+2T_0}(X_k)||^2|)\quad(39)$$

Let $J_1=\sum_{k=\tau_i}^{\tau_i+T_0}||Y_k-\hat{f}_{\tau_i}(X_k)||^2-\sum_{k=\tau_i-T_0}^{\tau_i}||Y_k-\hat{f}_{\tau_i-T_0}(X_k)||^2$, $J_2=\sum_{k=\tau_i}^{\tau_i+T_0}||Y_k-\hat{f}_{\tau_i}(X_k)||^2-\sum_{k=\tau_i+T_0}^{\tau_i+2T_0}||Y_k-\hat{f}_{\tau_i+T_0}(X_k)||^2$, $J_3=|\sum_{k=\tau_i-T_0}^{\tau_i}||Y_k-\hat{f}_{\tau_i-T_0}(X_k)||^2-\sum_{k=\tau_i-2T_0}^{\tau_i-T_0}||Y_k-\hat{f}_{\tau_i-2T_0}(X_k)||^2|$ and $J_4=|\sum_{k=\tau_i+T_0}^{\tau_i+2T_0}||Y_k-\hat{f}_{\tau_i+T_0}(X_k)||^2-\sum_{k=\tau_i+2T_0}^{\tau_i+3T_0}||Y_k-\hat{f}_{\tau_i+2T_0}(X_k)||^2|$.

For $J_1$:

$$J_1=\sum_{k=\tau_i}^{\tau_i+T_0}||Y_k-\hat{f}_{\tau_i}(X_k)||^2-\sum_{k=\tau_i-T_0}^{\tau_i}||Y_k-\hat{f}_{\tau_i-T_0}(X_k)||^2$$
$$\geq\sum_{k=\tau_i}^{\tau_i+T_0}(||f_{i+1}(X_k)-f_i(X_k)||^2-||Y_k-f_{i+1}(X_k)||^2)-\sum_{k=\tau_i}^{\tau_i+T_0}||f_i(X_k)-\hat{f}_{\tau_i}(X_k)||^2$$
$$-\sum_{k=\tau_i-T_0}^{\tau_i}(||Y_k-f_i(X_k)||^2+||f_i(X_k)-\hat{f}_{\tau_i-T_0}||^2)$$
$$\overset{Theorem10.1}{\geq}o(1)+M1_iT_0-C_1T_0\phi_{\tau_i-\tau_{i-1}}L\log^2(\tau_i-\tau_{i-1})$$
$$-C_2T_0\phi_{\tau_i-\tau_{i-1}-T_0}L\log^2(\tau_i-\tau_{i-1}-T_0)-2T_0h\sigma^2\quad(40)$$

27

For $J_2$:

$$J_2 = \sum_{k=\tau_i}^{\tau_i+T_0} ||Y_k - \hat{f}_{\tau_i}(X_k)||^2 - \sum_{k=\tau_i+T_0}^{\tau_i+2T_0} ||Y_k - \hat{f}_{\tau_i+T_0}(X_k)||^2$$

$$\geq \sum_{k=\tau_i}^{\tau_i+T_0} (||f_{i+1}(X_k) - f_i(X_k)||^2 - ||Y_k - f_{i+1}(X_k)||^2) - \sum_{k=\tau_i}^{\tau_i+T_0} ||f_i(X_k) - \hat{f}_{\tau_i}(X_k)||^2$$

$$- \sum_{k=\tau_i+T_0}^{\tau_i+2T_0} (||Y_k - f_{i+1}(X_k)||^2 + ||f_{i+1}(X_k) - \hat{f}_{\tau_i+T_0}(X_k)||^2)$$

$$\overset{Theorem 10.1}{\geq} o(1) + M1_i T_0 - C_1 T_0 \phi_{\tau_i-\tau_{i-1}} L \log^2(\tau_i - \tau_{i-1}) - C_2 T_0 \phi_{T_0} L \log^2(T_0) - 2T_0 h \sigma^2 \tag{41}$$

For $J_3$:

$$J_3 = |\sum_{k=\tau_i-T_0}^{\tau_i} ||Y_k - \hat{f}_{\tau_i-T_0}(X_k)||^2 - \sum_{k=\tau_i-2T_0}^{\tau_i-T_0} ||Y_k - \hat{f}_{\tau_i-2T_0}(X_k)||^2|$$

$$\leq \sum_{k=\tau_i-T_0}^{\tau_i} (||Y_k - f_i(X_k)||^2 + ||f_i(X_k) - \hat{f}_{\tau_i-T_0}(X_k)||^2)$$

$$+ \sum_{k=\tau_i-2T_0}^{\tau_i-T_0} (||Y_k - f_i(X_k)||^2 + ||f_i(X_k) - \hat{f}_{\tau_i-2T_0}(X_k)||^2)$$

$$\overset{Theorem 10.1}{\leq} o(1) + C_1 T_0 \phi_{\tau_i-\tau_{i-1}-2T_0} L \log^2(\tau_i - \tau_{i-1} - 2T_0)$$

$$+ C_2 T_0 \phi_{\tau_i-\tau_{i-1}-T_0} L \log^2(\tau_i - \tau_{i-1} - T_0) + 2T_0 h \sigma^2 \tag{42}$$

For $J_4$:

$$J_4 = |\sum_{k=\tau_i+T_0}^{\tau_i+2T_0} ||Y_k - \hat{f}_{\tau_i+T_0}(X_k)||^2 - \sum_{k=\tau_i+2T_0}^{\tau_i+3T_0} ||Y_k - \hat{f}_{\tau_i+2T_0}(X_k)||^2|$$

$$\leq \sum_{k=\tau_i+T_0}^{\tau_i+2T_0} (||Y_k - f_{i+1}(X_k)||^2 + ||f_{i+1}(X_k) - \hat{f}_{\tau_i+T_0}(X_k)||^2)$$

$$+ \sum_{k=\tau_i+2T_0}^{\tau_i+3T_0} (||Y_k - f_{i+1}(X_k)||^2 + ||f_{i+1}(X_k) - \hat{f}_{\tau_i+2T_0}(X_k)||^2)$$

$$\overset{Theorem 10.1}{\leq} o(1) + 2T_0 h \sigma^2 + C_1 T_0 \phi_{T_0} L \log^2 T_0 + C_2 T_0 \phi_{2T_0} L \log^2(2T_0) \tag{43}$$

By the previous equation, $T_0 \ll T \leq \tau_i - \tau_{i-1}$, $\min(J_1, J_2) = J_2, \max(J_3, J_4) = J_4$. To let $\min(J_1, J_2) \gg \max(J_3, J_4)$ which is $J_2 \gg J_4$.

$$o(1) + M1_i T_0 - 2T_0 h \sigma^2 - C_1 T_0 \phi_{\tau_i-\tau_{i-1}} L \log^2(\tau_i - \tau_{i-1}) - C_2 T_0 \phi_{T_0} L \log^2(T_0)$$

$$\gg o(1) + 2T_0 h \sigma^2 + C_1 T_0 \phi_{T_0} L \log^2 T_0 + C_2 T_0 \phi_{2T_0} L \log^2(2T_0) \tag{44}$$

To satisfy the condition above,

$$T_0 \gg \frac{o(1)}{M1_i - 4h\sigma^2 - C\phi_{T_0}L\log^2(T_0)} \tag{45}$$

This will be satisfied as $T_0 \to \infty$ when $T_{sum} \to \infty$ and then the algorithm will detect a peak near the true change point with probability $1 - \frac{C}{T_0}$ (or $1 - \frac{C}{T_0^{p-1}}$ and $1 - e^{-CT_0}$) for some constant C depending on the type of input and model. And when $\pi = \frac{M1_*}{2} - 2h\sigma^2$, $J_2 - J_4 > \pi$ thus finishing this part of proof.

Then, we start to compute the probability of the approximation. By assumption and then we can finish the proof for these separate cases.

For independent variables, for each changepoint, with probability at least $1 - \frac{C}{T_0^{p-1}}$, the algorithm will detect the changepoint. Then, with probability at least $(1 - \frac{C}{T_0^{p-1}})^N$, the algorithm will detect all N changepoints and exactly these N changepoints and the distance between the estimated change point and the true one will be controlled.

$$(1 - \frac{C}{T_0^{p-1}})^N \geq 1 - \frac{CN}{T_0^{p-1}} \geq 1 - \frac{CT_{sum}}{TT_0^{p-1}} \geq 1 - \frac{C_0 T_{sum}}{T^{1+\rho(p-1)}}$$

$$\geq 1 - C_0 T_{sum}^{-\frac{1}{\rho(p-1)}} \xrightarrow[T_{sum}\to\infty]{} 1 \tag{46}$$

For independent and subgaussian variables, for each changepoint, with probability at least $1 - Ce^{-C_1 T_0}$, the algorithm will detect the changepoint. Then, with probability at least $(1 - Ce^{-C_1 T_0})^N$, the algorithm will detect all N changepoints and exactly these N changepoints and the distance between the estimated change point and the true one will be controlled.

$$(1 - Ce^{-C_1 T_0})^N \geq 1 - CNe^{-C_1 T_0} \geq 1 - C_0 \frac{T_{sum}}{Te^{C_1 T_0}}$$

$$\geq 1 - \frac{CT_{sum}}{C_2 \log(T_{sum})T_{sum}} = 1 - \frac{C_4}{\sqrt{\log(T_{sum})}} \to 1 \tag{47}$$

For dependent variables, for each changepoint, with probability at least $1 - \frac{C}{T_0}$, the algorithm will detect the changepoint. Then, with probability at least $(1 - \frac{C}{T_0})^N$, the algorithm will detect all N changepoints and exactly these N changepoints and the distance between the estimated change point and the true one will be controlled.

$$(1 - \frac{C}{T_0})^N \geq 1 - \frac{CN}{T_0} \geq 1 - \frac{CT_{sum}}{TT_0} \geq 1 - \frac{C_0 T_{sum}}{T^{1+\rho}}$$

$$\geq 1 - \frac{C_0 T_{sum}}{T_{sum}^{\frac{\rho+1}{1+g\rho}}} \geq 1 - C_0 T_{sum}^{-\frac{(1-g)\rho}{g\rho++1}} \xrightarrow[T_{sum}\to\infty]{} 1 \tag{48}$$

Then we finish the proof.

# 9 reference

# References

[1] T. L. Lai, "Sequential changepoint detection in quality control and dynamical systems," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 57, no. 4, pp. 613–644, 1995. [Online]. Available: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1995.tb02052.x

[2] X. Zhu, Y. Xie, J. Li, and D. Wu, "Change point detection for subprime crisis in american banking: From the perspective of risk dependence," *International Review of Economics & Finance*, vol. 38, pp. 18–28, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1059056015000040

[3] S. Liu, A. Wright, and M. Hauskrecht, "Change-point detection method for clinical decision support system rule monitoring," *Artificial Intelligence in Medicine*, vol. 91, pp. 49–56, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S093336571730595X

[4] L. Kendrick, K. Musial, and B. Gabrys, "Change point detection in social networks - critical review with experiments," *Comput. Sci. Rev.*, vol. 29, pp. 1–13, 2018.

[5] V. Kulkarni, R. Al-Rfou, B. Perozzi, and S. Skiena, "Statistically significant detection of linguistic change," in *Proceedings of the 24th International World Wide Web Conference*, ser. WWW '15, 2015.

[6] Y. Wang and C. Goutte, "Real-time change point detection using on-line topic models," in *International Conference on Computational Linguistics*, 2018.

[7] J. V. Braun, R. K. Braun, and H.-G. Müller, "Multiple changepoint fitting via quasilikelihood, with application to dna sequence segmentation," *Biometrika*, vol. 87, pp. 301–314, 2000.

[8] Z. Harchaoui and C. Lévy-Leduc, "Multiple change-point estimation with a total variation penalty," *Journal of the American Statistical Association*, vol. 105, pp. 1480 – 1493, 2010.

[9] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1590–1598, oct 2012. [Online]. Available: https://doi.org/10.1080%2F01621459.2012.737745

[10] Y. S. Niu and H. Zhang, "The screening and ranking algorithm to detect DNA copy number variations," *The Annals of Applied Statistics*, vol. 6, no. 3, sep 2012. [Online]. Available: https://doi.org/10.1214%2F12-aoas539

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

[12] P. Bai, A. Safikhani, and G. Michailidis, "Multiple change points detection in low rank and sparse high dimensional vector autoregressive models," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3074–3089, 2020.

[13] A. Safikhani, Y. Bai, and G. Michailidis, "Fast and scalable algorithm for detection of structural breaks in big VAR models," *J. Comput. Graph. Stat.*, vol. 31, no. 1, pp. 176–189, 2022. [Online]. Available: https://doi.org/10.1080/10618600.2021.1950005

[14] J. Bai and P. Perron, "Estimating and testing linear models with multiple structural changes," *Econometrica*, pp. 47–78, 1998.

[15] Garreau and Arlot, "Consistent change-point detection with kernels," *Electronic Journal of Statistics*, vol. 12, 12 2016.

[16] M. Bhattacharjee, M. Banerjee, and G. Michailidis, "Change point estimation in a dynamic stochastic block model," *Journal of Machine Learning Research*, vol. 21, no. 107, pp. 1–59, 2020. [Online]. Available: http://jmlr.org/papers/v21/18-814.html

[17] H. Chen and N. Zhang, "Graph-based change-point detection," *The Annals of Statistics*, vol. 43, no. 1, pp. 139 – 176, 2015. [Online]. Available: https://doi.org/10.1214/14-AOS1269

[18] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, p. 107299, Feb. 2020. [Online]. Available: http://dx.doi.org/10.1016/j.sigpro.2019.107299

[19] M. K. Titsias, J. Sygnowski, and Y. Chen, "Sequential changepoint detection in neural networks with checkpoints," 2020.

[20] J. Lee, Y. Xie, and X. Cheng, "Training neural networks for sequential change-point detection," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[21] Z. Ebrahimzadeh, M. Zheng, S. Karakas, and S. Kleinberg, "Pyramid recurrent neural networks for multi-scale change-point detection," 2018.

[22] M. L. Bulunga, "Change-point detection in dynamical systems using auto-associative neural networks," 2012.

[23] M. Gupta, R. Wadhvani, and A. Rasool, "Real-time change-point detection: A deep neural network-based adaptive approach for detecting changes in multivariate time series data," *Expert Systems with Applications*, vol. 209, p. 118260, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422014026

[24] 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

[25] Ángel Carmona-Poyato, N. L. Fernández-Garcia, F. J. Madrid-Cuevas, and A. M. Durán-Rosal, "A new approach for optimal offline time-series segmentation with error bound guarantee," *Pattern Recognition*, vol. 115, p. 107917, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320321001047

[26] M. Londschien, P. Bühlmann, and S. Kovács, "Random forests for change point detection," 2023. [Online]. Available: https://arxiv.org/abs/2205.04997

[27] A. Alanqary, A. O. Alomar, and D. Shah, "Change point detection via multivariate singular spectrum analysis," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=i0DmV60aeK

[28] W.-C. Chang, C.-L. Li, Y. Yang, and B. Póczos, "Kernel change-point detection with auxiliary deep generative models," 2019. [Online]. Available: https://arxiv.org/abs/1901.06077

[29] J. Li, P. Fearnhead, P. Fryzlewicz, and T. Wang, "Automatic change-point detection in time series via deep learning," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 86, no. 2, p. 273–285, Jan. 2024. [Online]. Available: http://dx.doi.org/10.1093/jrsssb/qkae004

[30] J. Schmidt-Hieber, "Nonparametric regression using deep neural networks with ReLU activation function," *The Annals of Statistics*, vol. 48, no. 4, aug 2020. [Online]. Available: https://doi.org/10.1214%2F19-aos1875

[31] M. Ma and A. Safikhani, "Theoretical analysis of deep neural networks for temporally dependent observations," 2022.

[32] C. F. Ansley and R. Kohn, "A note on reparameterizing a vector autoregressive moving average model to enforce stationarity," *J. Stat. Comput. Simul.*, vol. 24, no. 2, p. 99–106, Jun. 1986.

[33] K. Bleakley and J.-P. Vert, "The group fused lasso for multiple change-point detection," *arXiv: Quantitative Methods*, 2011.

[34] N. A. James and D. S. Matteson, "ecp: An r package for nonparametric multiple change point analysis of multivariate data," *Journal of Statistical Software*, vol. 62, no. 7, p. 1–25, 2015. [Online]. Available: https://www.jstatsoft.org/index.php/jss/article/view/v062i07

[35] "National accounts," Federal Reserve, Tech. Rep., 2020, url https://fred.stlouisfed.org/series/A939RX0Q048SBEA .

[36] H. Cho and P. Fryzlewicz, "Multiple-change-point detection for high dimensional time series via sparsified binary segmentation," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 77, no. 2, p. 475–507, Jul. 2014. [Online]. Available: http://dx.doi.org/10.1111/rssb.12079

[37] H. Cho, "Change-point detection in panel data via double cusum statistic," 2016.

[38] J. Lin and G. Michailidis, "A multi-task encoder-dual-decoder framework for mixed frequency data prediction," *International Journal of Forecasting*, vol. 40, no. 3, pp. 942–957, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016920702300078X

[39] R. Marcinkevičs and J. E. Vogt, "Interpretable models for granger causality using self-explaining neural networks," 2021. [Online]. Available: https://arxiv.org/abs/2101.07600

## 10 Supplement

### 10.1 $\beta$-Hölder smoothness

First, we need some assumptions and also demonstrate the features of the model functions $f_j$ which are composed of several functions: $f_j = g_q \circ g_{q-1} \circ ... \circ g_1 \circ g_0$. We need to define a type of function called $\beta$-Hölder smoothness whos partial derivatives exist up to $\lfloor \beta \rfloor$ order and are bounded and the partial derivatives of order $\lfloor \beta \rfloor$ are $\beta - \lfloor \beta \rfloor$ Hölder. Then define the ball of $\beta$-Hölder smoothness functions with radius K as:

$$\mathcal{C}_r^\beta(D, K) = \{ f : D \subset R^r \to R; \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|<\beta} ||\partial^{\boldsymbol{\alpha}} f||_\infty +$$
$$\sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|=\lfloor\beta\rfloor} \sup_{x,y\in D; x\neq y} \frac{|\partial^{\boldsymbol{\alpha}} f(x) - \partial^{\boldsymbol{\alpha}} f(y)|}{|x-y|_\infty^{\beta-\lfloor\beta\rfloor}} \leq K \} \quad (49)$$

where $\partial^{\boldsymbol{\alpha}} = \partial^{\alpha_1}...\partial^{\alpha_r}$ with $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_r) \in \mathbb{N}^r$ where $\partial^{\alpha_i}$ denotes the $\alpha_i$ partial derivative on the $i^{th}$ dimension and $|\boldsymbol{\alpha}| := |\boldsymbol{\alpha}|_1$. We assume $g_{ij}$ is $\beta$-Hölder smoothness then we define a function space to be:

$$\mathcal{G}(q, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta}, K) := \{ f = g_q \circ g_{q-1} \circ ... \circ g_1 \circ g_0 :$$
$$g_i = (g_{ij})_j : [a_i, b_i]^{d_i} \to [a_{i+1}, b_{i+1}]^{d_{i+1}},$$
$$g_{ij} \in \mathcal{C}_{t_i}^{\beta_i}([a_i, b_i]^{t_i}, K) \text{ where } |a_i| \leq K, |b_i| \leq K \} \quad (50)$$

where $\boldsymbol{d} := (d_0, ..., d_{q+1}), \boldsymbol{t} := (t_0, ..., t_q), \boldsymbol{\beta} := (\beta_0, ..., \beta_q)$. This function space is the domain of our model function.

## 10.2 Neural network Function space

Then we start to define the function space of network functions. First, we define a network function space with bounded parameters as:

$$
\begin{aligned}
\mathcal{F}(L, \boldsymbol{p}) := \{f : \mathbb{R}^{p_0} \to \mathbb{R}^{p_{L+1}}, \\
x \to f(x) = W_L \sigma_{v_L} W_{L-1} \sigma_{v_{L-1}} ... W_1 \sigma_{v_1} W_0 x : \\
\max_{j=0,...,L} ||W_j||_\infty \vee |v_j|_\infty \leq 1\} \quad (51)
\end{aligned}
$$

Where $\boldsymbol{p} = (p_0, ..., p_{L+1}) \in \mathbb{N}^{L+2}$, L is the number of layers of the network, $W_j$ is the $p_{j+1} \times p_j$ matrix, $v_j \in R^{p_j}$ and $\sigma_{v_j}(x) = \max(0, x - v_j)$ is the ReLU function. Then $f : \mathbb{R}^{p_0} \to \mathbb{R}^{p_{L+1}}$ defined above is a neural network with ReLU activation. Here $p_0 = l$ and $p_{L+1} = h$.

Let $|||f|_\infty||_\infty$ be sup-norm of the function $x \to |f(x)|_\infty$. Add the sparse condition to the parameters and define the network function space as:

$$
\mathcal{F}(L, \boldsymbol{p}, s) := \mathcal{F}(L, \boldsymbol{p}, s, F) :=
$$

$$
\{f \in \mathcal{F}(L, \boldsymbol{p}) : \sum_{j=0}^{L} ||W_j||_0 + |v_j|_0 \leq s, |||f|_\infty||_\infty \leq F\} \quad (52)
$$

Then start to approximate the convergence rate. First, let

$$
\beta_i^* := \prod_{i+1}^{q} (\beta_l \wedge 1) \quad (53)
$$

And define convergence rate as:

$$
\phi_n := \max_{i=0,...,q} n^{-\frac{2\beta_i^*}{\beta_i^* + t_i}} \quad (54)
$$

where n is the size of the training set where the network is trained.

## 10.3 Lemma

The next two lemmas in paper [30, 31] will be the key to our proof for the main theorems. The model function space and the network function space will be defined as above.

**Lemma 10.1** *Let $f_0 \in \mathcal{G}(q, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta}, K)$ and define a network function space $\mathcal{F}(L, \boldsymbol{p}, s, F)$. Let $Y_i = f_0(X_i) + \epsilon_i$ as defined by model?? where X is independent and $\hat{f} \in \arg\min_{f \in \mathcal{F}(L, \boldsymbol{p}, s, F)} \sum_{i=1}^{n} (Y_i - f(X_i))^2$ be an empirical risk minimizer and satisfied conditions below:*

(i)$F \geq \max(K, 1)$
(ii)$\sum_{i=0}^{q} \log_2(4t_i \vee 4\beta_i) \log_2 n \leq L \lesssim n\phi_n$
(iii)$n\phi_n \lesssim \min_{i=1,\ldots,L} p_i$
(iv)$s \simeq n\phi_n \log n$
There exists a constant C' only depending on $q, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta}$, F such that

$$R(\hat{f} - f_0) = \mathbb{E}_{f_0}[(\hat{f}(X) - f_0(X))^2] \leq C'\phi_n L \log^2 n$$

**Lemma 10.2** Let $f_0 \in \mathcal{G}(q, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta}, K)$ and define a network function space $\mathcal{F}(L, \boldsymbol{p}, s, F)$. Let $Y_i = f_0(X_i) + \epsilon_i$ as defined by equation1 in section 2 where X is dependent and $\hat{f} \in arg\min_{f \in \mathcal{F}(L, \boldsymbol{p}, s, F)} \sum_{i=1}^{n}(Y_i - f(X_i))^2$ be an empirical risk minimizer and satisfied conditions below:
(i)$F \geq \max(K, 1)$
(ii)$\sum_{i=0}^{q} \log_2(4t_i \vee 4\beta_i) \log_2 n \leq L \lesssim n\phi_n$
(iii)$n\phi_n \lesssim \min_{i=1,\ldots,L} p_i$
(iv)$s \simeq n\phi_n \log n$
There exists a constant C' only depending on $q, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta}$, F such that

$$R(\hat{f} - f_0) = \mathbb{E}_{f_0}[(\hat{f}(X) - f_0(X))^2] \leq C'\phi_n L \log^6 n$$

For the proof of our theorems, we also need 3 common concentration inequalities:

**Lemma 10.3** Given $X_1, \ldots, X_n$ i.i.d subgaussian, then for $\forall \delta > 0$, there exists constant $C_1, C_2$ greater than 0.

$$P\left(|\bar{X} - \mu| > \delta\right) \leq C_1 e^{-C_2 n}$$

where $\mu = \mathbb{E}[X_i]$ and $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$

**Lemma 10.4** Given $X_1, \ldots, X_n$ independent and from the same distribution, then if $\mathbb{E}(|X_i|^p) < \infty$, there exists a constant $C > 0$,

$$P\left(|\bar{X} - \mu| > \delta\right) \leq \frac{C}{n^{p-1}}$$

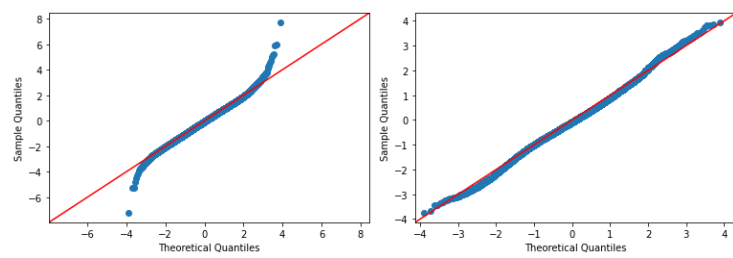where $\mu = \mathbb{E}[X_i]$ and $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$

**Lemma 10.5** Given $X_1, \ldots, X_n$ dependent and from the same distribution, then if $|cov(X_i, X_j)| < c < 1$ and $\mathbb{E}(|X_i|^2) < \infty$, there exists a constant $C > 0$,

$$P\left(|\bar{X} - \mu| > \delta\right) \leq \frac{C}{n}$$

where $\mu = \mathbb{E}[X_i]$ and $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$

## 10.4   Additional figures

The figure 5 demonstrates the normality of the data output. The left qqplot is for nonsubgaussian data and right is for subgaussian data.

(a) qqplot for nonsubgaussian data

(b) qqplot for subgaussian data

Figure 5: QQplot