

Comprehensible Counterfactual Interpretation on Kolmogorov-Smirnov Test

Zicun Cong
Simon Fraser University
Burnaby, Canada
zcong@sfu.ca

Yu Yang
City University of Hong Kong
Hong Kong, China
yuyang@cityu.edu.hk

Lingyang Chu
McMaster University
Hamilton, Canada
chul9@mcmaster.ca

Jian Pei
Simon Fraser University
Burnaby, Canada
jpei@cs.sfu.ca

ABSTRACT

The Kolmogorov-Smirnov (KS) test is popularly used in many applications, such as anomaly detection, astronomy, database security and AI systems. One challenge remained untouched is how we can obtain an interpretation on **why a test set fails the KS test**. In this paper, we tackle the problem of producing counterfactual interpretations for test data failing the KS test. Concept-wise, we propose the notion of most comprehensible counterfactual interpretations, which accommodates both the KS test data and the user domain knowledge in producing interpretations. Computation-wise, we develop an efficient algorithm MOCHI that avoids enumerating and checking an exponential number of subsets of the test set failing the KS test. MOCHI not only guarantees to produce the most comprehensible counterfactual interpretations, but also is orders of magnitudes faster than the baselines. Experiment-wise, we present a systematic empirical study on a series of benchmark real datasets to verify the effectiveness, efficiency and scalability of most comprehensible counterfactual interpretations and MOCHI.

PVLDB Reference Format:

Zicun Cong, Lingyang Chu, Yu Yang, and Jian Pei. Comprehensible Counterfactual Interpretation on Kolmogorov-Smirnov Test. PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:

The source code, data, and/or other artifacts have been made available at http://vldb.org/pvldb/format_v14.html.

1 INTRODUCTION

The well-known Kolmogorov-Smirnov (KS) test [25] is a statistical hypothesis test that checks whether a test set is sampled from the same probability distribution as a reference set. If a reference set and a test set fail the KS test, it indicates that the two sets are unlikely from the same probability distribution. The KS test has been widely used to detect abnormalities in many areas, such as astronomy [38],

database security [51] and AI systems [46]. Understanding the causes of why a test set fails the KS test not only can build trust for public users [49] but also helps locate anomalies [46]. However, the KS test itself does not come with an explanation on which data points in the test set cause the failure.

Having the highest level of interpretability [44], counterfactual interpretations [44, 53] have been widely adopted to identify the causes of decisions made in many real world applications. A counterfactual interpretation of a decision Y is the smallest set of relevant factors X such that if X had not occurred, Y would not have occurred. When a reference set R and a test set T fail the KS test, we call it a *failed KS test*. A counterfactual interpretation on a failed KS test is a minimum subset S of the test set T such that removing the subset from T reverses the failed KS test into a passed one, that is, R and $T \setminus S$ pass the KS test.

Although interpreting failed KS tests is interesting and has many potential applications, it has not been touched in literature. Many counterfactual interpretation methods interpret the decisions of machine learning models [3, 19, 30]. Unfortunately, the existing methods cannot be adopted to interpret failed KS tests. As reviewed in Section 2, to interpret a failed KS test, the existing methods have to solve an L_0 -norm optimization problem, which is NP-hard [33]. Some methods [45, 46] try to select the outliers in the test set as a hint to a failed KS test. However, because outlier detection methods and the KS test use different mechanisms to detect anomalies, there is no guarantee that the outliers are the causes of the failed KS test. Besides, due to the Roshomon effect [36], multiple counterfactuals may co-exist for a failed KS test but not all of them are comprehensible to users [36]. Simply presenting all counterfactuals not only may overwhelm users but is also computationally expensive [34]. A desirable idea is to find an interpretation that is most consistent with a user's domain knowledge so that the interpretation is best comprehensible to the user [10, 37]. However, none of the existing methods can find such most comprehensible interpretations. Moreover, finding the most comprehensible interpretation is far from trivial. A brute force method has to enumerate all subsets of a test set and, for each subset, conduct the KS test. Thus, the brute force method takes exponential time.

In this paper, we tackle the problem of producing interpretations for failed KS tests by developing the notion of comprehensible counterfactual interpretations and the finding algorithm. We make several contributions. Concept-wise, given a failed KS test, we find

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

a smallest subset of the test set such that removing the subset from the test set reverses the failed KS test into a passed one. To address user comprehensibility [10, 36], we take a user’s domain knowledge represented as a preference order on the data points in the test set, and guarantee to find the counterfactual interpretation that is most consistent with the preference. Computation-wise, we develop MOCHI, a two-step method that guarantees to find the most comprehensible counterfactual interpretation on a failed KS test fast. Specifically, MOCHI first identifies the number of data points in the interpretation and then efficiently constructs the most comprehensible interpretation. We establish an important insight that the size of removed data points is the smallest integer satisfying a group of inequalities. Leveraging this property, an efficient searching algorithm is designed to find the interpretation size. Then, MOCHI efficiently constructs the most comprehensible interpretation by one scan of the data points in the test set. Experiment-wise, we conduct a systematic empirical study on a series of benchmark real datasets to verify the effectiveness, efficiency and scalability of most comprehensible counterfactual interpretations and MOCHI.

The rest of the paper is organized as follows. We review the related work in Section 2. Then, we formulate our problem and present a naïve baseline method in Section 3. We investigate how to find the size of interpretations in Section 4. In Section 5, we present the algorithm to find the most comprehensible counterfactual interpretation. We report a systematic empirical study in Section 6 and conclude the paper in Section 7.

2 RELATED WORK

To the best of our knowledge, interpreting a failed KS test is a novel task that has not been systematically investigated in literature. Our study is broadly related to the Kolmogorov-Smirnov test [23, 24, 52], counterfactual interpretations [3, 19, 30], adversarial attacks [7, 15, 43] and outlier detection [16, 21, 47].

The Kolmogorov-Smirnov (KS) test [25] is a well-known statistical hypothesis test that checks whether two samples are originated from the same probability distribution. With the advantages of being efficient, non-parametric, and distribution-free [27], the KS test has been widely used in many applications to detect abnormalities [16], such as identifying change points in time series [16, 24], maintaining machine learning models [17, 46, 52], detecting outliers [23], ensuring quality of encrypted or anonymized data [2, 22], and protecting databases from intrusion attacks [51].

Understanding the causes of anomalies is the key to tackle many problems in real world applications [45, 46]. However, a failed KS test itself does not provide users any hints on which data points in the test set cause the failure. Therefore, finding interpretations of failed KS tests is a natural next step.

Counterfactual interpretations [44, 53] have been widely adopted to identify the causes of decisions made in many real world applications. Counterfactual interpretation methods [3, 19, 36, 55] interpret the prediction on a given instance by applying small and interpretable perturbations on the instance such that the prediction is changed [36]. For example, Fong *et al.* [19] interpret the prediction of an image by finding the smallest pixel-deletion mask that leads to the most significant drop of the prediction score. As an extension, Akula *et al.* [3] identify meaningful image patches that need to be

added to or deleted from an input image. Van Looveren *et al.* [55] use class prototypes to generate counterfactuals that lie close to the classifier’s training data distribution. Moore *et al.* [35] propose a method to generate counterfactuals from adversarial examples with gradient constraints. Le *et al.* [30] use an entropy-based feature selection approach to limit the features to be perturbed.

Unfortunately, the existing counterfactual interpretation methods cannot interpret a failed KS test by perturbing the data points in the test set. To minimize the number of perturbed data points, the existing methods need to minimize the L_0 -norm of their perturbations [33]. However, such an optimization problem is NP-hard [33, 40]. The existing methods cannot guarantee to reach a global minimum for the optimization problem.

Adversarial attack methods [7, 13, 15, 42, 43] also have a potential to be extended to find counterfactual interpretations on failed KS tests. To attack a target classifier, an adversarial attack method generates an imperceptible perturbation on an input so that the prediction on the input is changed. Brendel *et al.* [7] propose to generate adversarial perturbations by moving instances towards the estimated decision boundaries of a target model. Cheng *et al.* [13] formulate the black-box attack as an optimization problem, which can be solved by zeroth order optimization approaches. Croce *et al.* [15] propose to attack image classifiers by applying randomly selected one-pixel modifications on images.

One may think that counterfactual interpretations on a failed KS test may be generated by attacking the KS test, that is, the perturbed data points can serve as a counterfactual interpretation on the KS test. However, extending the existing adversarial attack methods to interpret failed KS tests also needs to minimize the L_0 -norm of the perturbations, which leads to the same computational challenge.

Outlier detection methods aim to detect samples that are different from the majority of the given data [1], such as distance-based approaches [4, 21, 47], density-based approaches [8, 20, 41] and ensemble-based approaches [16, 29]. In general, outliers are regarded as abnormal data points [1].

Even though both KS test and outlier detection methods can detect anomalies in data, the detected outliers in the test set cannot be used as a counterfactual interpretation on a failed KS test. This is because outlier detection methods and the KS test use different mechanisms to detect anomalies. There is no guarantee that the outliers are the causes of a failed KS test. Just removing the outliers cannot guarantee to reverse a failed KS test to a passed one.

3 PROBLEM FORMULATION AND ANALYSIS

In this section, we first review the basic concepts of the Kolmogorov-Smirnov (KS) test. Then, we investigate how to generate a counterfactual interpretation on a KS test. Third, we discuss the comprehensibility of interpretations, and formalize the problem of finding the most comprehensible interpretation on a failed KS test. Next, we investigate the existence and uniqueness of most comprehensible counterfactual interpretations. Last, we describe a brute force method.

3.1 The Kolmogorov-Smirnov Test

Denote by $R = \{r_1, \dots, r_n\}$ a multi-set of real numbers from an unknown univariate probability distribution, and by $T = \{t_1, \dots, t_m\}$

Notation	Description
R	The reference set of a KS test.
T	The test set of a KS test.
S	An h -subset of T .
L	A preference list on the test set T .
$<$	The lexicographical order based on L .
C	An h -cumulative vector.
\mathcal{I}	An interpretation.
q	The number of unique data points in $R \cup T$.
n	The size of R , that is $n = R $.
m	The size of T , that is $m = T $.
k	The size of \mathcal{I} , that is $k = \mathcal{I} $.
$C[i], c_i$	The i -th element of an h -cumulative vector C .
l_i^h	The lower bound of $C[i]$.
u_i^h	The upper bound of $C[i]$.
x_i	An element in $R \cup T$.
c_α	The critical value at significance level α .

Table 1: Frequently used notations.

another multi-set of real numbers that are sampled from a distribution may or may not be the same as R . We call R a *reference set* and T a *test set*. In this paper, by default multi-set is used. In the rest of the paper, we use the terms “set” and “multi-set” interchangeably unless specifically mentioned.

The Kolmogorov-Smirnov (KS) test checks whether T is sampled from the same probability distribution as R by comparing the empirical cumulative functions of R and T . In the KS test, the null hypothesis is that T is sampled from the same probability distribution as R . Conducting the KS test consists of 3 steps as follows.

Step 1. We compute the KS statistic [17] by

$$D(R, T) = \max_{x \in R \cup T} |F_R(x) - F_T(x)|, \quad (1)$$

where $F_R(x)$ and $F_T(x)$ are the empirical cumulative functions of R and T , respectively. Here, a larger value of $D(R, T)$ indicates that the empirical cumulative functions of R and T are more different from each other.

Step 2. For a user-specified significance level α , we compute the corresponding target p -value [17] by $p = c_\alpha \sqrt{\frac{n+m}{n*m}}$, where $c_\alpha = \sqrt{-\frac{1}{2} \ln(\frac{\alpha}{2})}$ is the critical value at significance level α , $n = |R|$ is the number of data points in R , and $m = |T|$.

Step 3. We compare p and $D(R, T)$. If $D(R, T) > p$, we reject the null hypothesis at significance level α . This means the empirical cumulative functions $F_R(x)$ and $F_T(x)$ are significantly different from each other, and thus unlikely T is sampled from the same distribution as R . If $D(R, T) \leq p$, we cannot reject the null hypothesis at significance level α . There is not enough evidence showing that T is not sampled from the same distribution as R .

If the null hypothesis is rejected by the KS test, we say R and T fail the KS test and it is a *failed KS test*. Otherwise, we say R and T pass the KS test.

To compute the KS statistic between R and T , we need to sort the elements in $R \cup T$ in ascending order. Therefore, it takes $O((n+m) \log(n+m))$ time to conduct the KS test.

Table 1 summarizes some frequently used notations in this paper.

3.2 Counterfactual Interpretations on the KS Test

Why are we interested in failed KS tests? More often than not, a failed hypothesis test indicates something unusual or unexpected [11, 27, 46]. Knowing what causes anomalies is critical in many applications [45, 46]. It is important to interpret a failed KS test so that we can effectively identify the causes of abnormalities.

Counterfactual interpretations have been well demonstrated to be more human-friendly than other types of interpretations [34], thus counterfactual interpretation methods are widely adopted to identify the causes of decisions made in many applications [9, 19, 37, 56]. The counterfactual interpretation methods describe the causal relationship between the cause X of a decision Y in the form of “If X had not occurred, Y would not have occurred”. X is called a counterfactual interpretation on Y if X is the smallest set of relevant factors, such that changing X can alter the decision Y [34, 56]. Following the above idea, we have the following definition.

DEFINITION 1. For a reference set R and a test set T that fail the KS test at a significance level α , a **counterfactual interpretation** on the failed KS test is a smallest subset \mathcal{I} of the test set T , such that R and $T \setminus \mathcal{I}$ pass the KS test at the same significance level α .

A counterfactual interpretation is also called an *interpretation* for short when the context is clear.

3.3 The Most Comprehensible Counterfactual Interpretation on a KS Test

Like many previously proposed counterfactual interpretations [19, 37, 56], the counterfactual interpretations on a failed KS test suffer from the *Roshomon effect* [36], that is, the number of unique counterfactual interpretations on a failed KS test can be as large as $\binom{|T|}{|\mathcal{I}|}$. Enumerating all those interpretations is computationally expensive. A user may be overwhelmed by a large number of possible interpretations [34].

As discovered by many studies on counterfactual interpretations [26, 37, 53], not all counterfactual interpretations are equally comprehensible to a user. Due to the effect of confirmation bias [39], an interpretation is more comprehensible if it is more consistent with the user’s domain knowledge [32]. As a result, a typical way to overcome the Roshomon effect is to rank all interpretations according to the user’s preference based on the domain knowledge, and return the most preferred interpretation to the user [5, 37].

Following the above idea, we assume a user’s preference as a total order on the data points in the test set T , that is, a *preference list* L on the test set T . A data point t that has a smaller rank in L is more preferred by the user.

A typical task of recommendation system is to recommend a group of items to a user, such that the group best satisfies the user’s preference [60]. The existing studies [6, 12, 54, 60] discover that a user’s interest in a group is dominated by the user’s top favorite items in the group. In the same vein, one can think of an interpretation as a recommended group of data points. Given two interpretations \mathcal{I}_1 and \mathcal{I}_2 on a failed KS test, if \mathcal{I}_1 includes better-ranked data points in L than \mathcal{I}_2 does, \mathcal{I}_1 is more preferred by the user than \mathcal{I}_2 , and thus is more comprehensible.

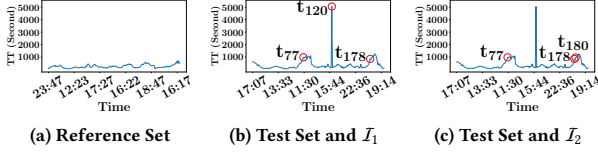


Figure 1: Two interpretations I_1 and I_2 on a failed KS test. (a) The reference set. For the ease of reading, the two interpretations I_1 and I_2 are shown in (b) and (c), respectively.

Based on the above intuition, an interpretation with a smaller lexicographical order¹ based on the preference list L is more preferred by the user. Specifically, for two interpretations I_1 and I_2 , $I_1 \subseteq T$, $I_2 \subseteq T$ and $|I_1| = |I_2|$, we sort the data points in I_1 and I_2 in the order of L . Denote by $I[i]$ the i -th data point in I in the order of L . Let $i_0 > 0$ be the smallest integer such that $I_1[i_0] \neq I_2[i_0]$. I_1 precedes I_2 in the lexicographical order, denoted by $I_1 < I_2$, if $I_1[i_0]$ precedes $I_2[i_0]$ in L . If $I_1 < I_2$, I_1 includes more top-ranked data items in L than I_2 .

DEFINITION 2. Given a failed KS test and a preference list L , the **most comprehensible counterfactual interpretation** is the interpretation that has the smallest lexicographical order based on L .

EXAMPLE 1. The Numenta Anomaly Benchmark (NAB) dataset [28] is a benchmark dataset for evaluation of time series anomaly detection algorithms, where the ground truth of abnormal segments is labeled by domain experts. NAB has different subsets of different topics, such as stock prices, AWS CPU usages and travel time. In this example, we consider a time series about the travel time between specific sensors collected by the Minnesota Department of Transportation. The time series is collected over time. At each time point, the travel time between the sensors is measured.

Figure 1a shows a normal segment right before an abnormal segment (Figure 1b) in the time series of travel time. Each segment has 200 data points. The normal segment is used as the reference set and the abnormal segment is used as the test set. They fail the KS test with significance level $\alpha = 0.05$.

Figures 1b and 1c show two interpretations of the failed KS test, $I_1 = \{t_{120}, t_{77}, t_{178}\}$ and $I_2 = \{t_{180}, t_{77}, t_{178}\}$, respectively. Figures 1b and 1c show two interpretations of the same segment of time series. A user who wants to investigate outlying travel times may provide a preference list L that sorts the data points in the Spectral Residual [48] decreasing order $L = [t_{120}, t_{180}, t_{77}, t_{178}, \dots]$.

Based on L , we have $I_1 < I_2$ in lexicographical order, that is, I_1 is more preferable than I_2 . As shown in the figures, I_1 and I_2 each includes 3 abnormally large data points. I_1 contains the most outlying data point t_{120} . Since the user is interested in outlying points, t_{120} strongly captures the user’s interest and thus I_1 is more interesting to the user than I_2 .

¹Given a total order L on items, the lexicographical order $<_{\text{lexicographical}}$ is $x_1x_2 \dots x_n <_{\text{lexicographical}} y_1y_2 \dots y_l$ if (1) $x_1 <_L y_1$; (2) there exists i_0 ($1 < i_0 \leq \min\{n, l\}$), $x_i = y_i$ for $1 \leq i < i_0$ and $x_{i_0} <_L y_{i_0}$; or (3) $m < l$ and for $1 \leq i \leq m$, $x_i = y_i$. Lexicographical order is also known as dictionary order.

The notion of comprehensible interpretation captures user preference in domain knowledge. We can employ different preference lists by applying different domain knowledge to sort the data points in the test set. For example, in some situations where distribution drifts are caused by outliers, we can rank data points based on outlying scores computed by an outlier detection method [45]. Then, the most comprehensible interpretation is the most outlying interpretation. As another example, when applying the KS test in a streaming environment, one may rank data points in their arrival time, because new coming data points are more likely to be the cause of a distribution drift [31]. Then, the more comprehensible interpretations contain more new-coming data points.

3.4 Existence and Uniqueness of Most Comprehensible Counterfactual Interpretations

For a failed KS test at significance level α , our task is to find the most comprehensible counterfactual interpretation I on the KS test. Does such an interpretation always exist?

PROPOSITION 1. When the significance level $\alpha \leq \frac{2}{e^2}$, there exists a unique most comprehensible interpretation on a failed KS test.

PROOF. (Existence) Consider a subset $S \subset T$, where $|S| = |T| - 1$. $|T \setminus S| = 1$. The p -value of the KS test between R and $T \setminus S$ is $p = c_\alpha \sqrt{\frac{n+1}{n \cdot 1}}$, where n is the size of R and $c_\alpha = \sqrt{-\ln(\frac{\alpha}{2})} \cdot \frac{1}{2}$. Since $\alpha \leq \frac{2}{e^2}$, we have $c_\alpha \geq 1$ and $p \geq \sqrt{\frac{n+1}{n}} \geq 1$. Since $D(R, T \setminus S)$ is the absolute difference between the two empirical cumulative functions, $1 \geq D(R, T \setminus S)$. Therefore, $p \geq D(R, T \setminus S)$. That is, R and $T \setminus S$ pass the KS test. Since an interpretation is a smallest subset that reverses a failed KS test to a passed one, there must exist an interpretation on a failed KS test given R and $T \setminus S$ passing the test.

(Uniqueness) Since each data point has a unique rank in L , two distinct interpretations cannot be equivalent in the lexicographical order. Thus, the most comprehensible interpretation is unique. \square

Statistical tests in practice typically use a significance level of 0.05 or lower. $\frac{2}{e^2} > 0.27$, which is far over the range of significance levels used in statistical tests. Therefore, our problem formulation is practical and guarantees a unique solution in practice.

3.5 A Brute Force Method

A naïve method to find the most comprehensible interpretation is to enumerate all subsets of the test set T and check against Definition 2. This brute-force method checks an exponential number of subsets, which is prohibitive for large test sets.

Even in a brute force method, we can significantly reduce the number of subset checking by early pruning a large number of subsets. According to Definitions 1 and 2, we can sort all subsets of T first by the size, from small to large, and then by the lexicographical order. This can be done by a breadth-first traversal of a set enumeration tree [50]. The first subset S in this order such that R and $T \setminus S$ pass the KS test is the most comprehensible interpretation.

4 SEARCHING INTERPRETATION SIZE

In this section, we first describe the two-phase framework of MOCHI, a fast method to find the most comprehensible counterfactual interpretations. Then, we thoroughly explore how to compute the size of interpretations fast.

4.1 MOCHI

According to Definition 1, all interpretations have the same size. Once we find an interpretation \mathcal{I} , we can safely ignore all subsets of T whose sizes are larger than $|\mathcal{I}|$, no matter they can reverse the KS test or not. Based on this idea, the MOCHI method proceeds in two phases. In phase 1, MOCHI tries to find the size of interpretations. In phase 2, MOCHI tries to identify the most comprehensible interpretations, that is, the smallest one in lexicographical order.

A subset S of T such that $|S| = h$ is called an h -subset. An h -subset S is a *qualified h -subset* if R and $T \setminus S$ pass the KS test. The first bottleneck is to check, for a given $h > 0$, whether there exists a qualified h -subset S . A brute-force implementation has to conduct the KS test a large number of times on all h -subsets. The time complexity is $O(\binom{m}{h}(m+n-h)\log(n+m-h))$.

Our first major technical result in this section is that checking the existence of a qualified h -subset does not have to conduct the KS test on all h -subsets. With a carefully designed data structure named cumulative vector to represent an h -subset of T , we establish a fast verification method for qualified cumulative vectors. Checking the existence of a qualified h -cumulative vector and thus a qualified h -subset only takes $O(m+n)$ time.

The second bottleneck is to find the size of interpretations efficiently. A brute-force method has to search from 1 to $m-1$ one by one and, for each size h , check the h -subsets. The second major technical result in this section tackles this bottleneck by deriving a lower bound \hat{k} on k , the size of all interpretations. This lower bound reduces the search range of h from $[1, k]$ to $[\hat{k}, k]$, which further reduces the time complexity of phase 1 to $O((m+n)\log(m) + (k-\hat{k})(m+n))$.

4.2 Cumulative Vectors

Essentially, the KS test compares the cumulative distribution functions of a reference set and a test set. Since there are only finite numbers of data points in a reference set and a test set, we can **represent the cumulative distribution function** of a reference set, a test set or a subset of the test set **using a sequence of the values of the cumulative distribution function at the data points appearing at either the reference set or the test set**. This observation motivates the design of the cumulative vectors.

We make a *base vector* $\mathbf{V} = \langle x_1, \dots, x_q \rangle$ from sets R and T , such that x_1, \dots, x_q are the unique data points in $R \cup T$. That is, a data point x_i appears in either R or T . No matter how many times x_i appears in $R \cup T$, it only appears once in \mathbf{V} . Thus, $q = \|R \cup T\|$, where R and T are treated as sets instead of multi-sets and q is the cardinality of the union, that is, duplicate items are not double counted. The elements in \mathbf{V} are sorted in the value ascending order, that is $x_1 < x_2 < \dots < x_q$.

DEFINITION 3. The cumulative vector of an h -subset $S \subseteq T$ is a $(q+1)$ -dimensional vector $\mathbf{C}_S = \langle c_0, c_1, \dots, c_q \rangle$, where $c_0 = 0$, and

for $1 \leq i \leq q$, c_i is the number of data points in S that are smaller than or equal to x_i in \mathbf{V} , that is $c_i = |\{x \in S \mid x \leq x_i\}|$. We also write c_i as $\mathbf{C}_S[i]$.

EXAMPLE 2. Given a reference set $R = \{0.5, 2, 5\}$ and a test set $T = \{0.5, 3, 3, 4, 4.5\}$, the base set $\mathbf{V} = \langle 0.5, 2, 3, 4, 4.5, 5 \rangle$. For a subset $S = \{3, 3\}$ of T , the cumulative vector is $\mathbf{C}_S = \langle 0, 0, 0, 2, 2, 2 \rangle$.

According to Definition 3, a cumulative vector \mathbf{C}_S contains all information to derive the cumulative distribution function $F_{T \setminus S}$ straightforwardly. For a cumulative vector $\mathbf{C}_S = \langle c_0, c_1, \dots, c_q \rangle$ and any i ($1 \leq i \leq q$), $c_i - c_{i-1}$ is the number of times that x_i appears in S . Thus, the value of the empirical cumulative distribution function of $T \setminus S$ at x_i can be computed by $F_{T \setminus S}(x_i) = \frac{\mathbf{C}_T[i] - c_i}{m - c_q}$, where \mathbf{C}_T is the cumulative vector of T and $\mathbf{C}_T[i]$ is the i -th element of \mathbf{C}_T and thus is the number of data points in T that are smaller than or equal to x_i .

Clearly, given a reference set R and a test set T , every unique subset $S \subseteq T$ corresponds to a unique cumulative vector \mathbf{C}_S and a unique cumulative distribution function $F_{T \setminus S}$, and vice versa. Recall that, if a subset $T \setminus S$ and R pass the KS test, S is called a qualified h -subset, where $h = |S|$. Correspondingly, we call the cumulative vector \mathbf{C}_S a qualified h -cumulative vector.

4.3 Existence of Qualified h -Cumulative Vectors

For a given h ($1 \leq h \leq |T|$), can we quickly determine whether there exists a qualified h -cumulative vector and thus a qualified h -subset? Before we state the major result, we need the following.

LEMMA 1. Given a reference set R and a test set T , for $S \subset T$, $\mathbf{C}_S = \langle c_0, c_1, \dots, c_q \rangle$ is a qualified cumulative vector if and only if, for each i ($1 \leq i \leq q$), the following two inequalities hold.

$$\max(\lceil \Gamma(i, h) - \Omega(h) \rceil, h - m + \mathbf{C}_T[i], \mathbf{C}_S[i - 1]) \leq \mathbf{C}_S[i] \quad (2a)$$

$$\mathbf{C}_S[i] \leq \min(\lceil \Gamma(i, h) + \Omega(h) \rceil, \mathbf{C}_T[i] - \mathbf{C}_T[i - 1] + \mathbf{C}_S[i - 1], h) \quad (2b)$$

where $\Omega(h) = c_\alpha \sqrt{m - h + \frac{(m-h)^2}{n}}$, $\Gamma(i, h) = \mathbf{C}_T[i] - \frac{m-h}{n} \mathbf{C}_R[i]$, and \mathbf{C}_R and \mathbf{C}_T are the cumulative vectors of R and T , respectively.

PROOF. (Necessity) According to the definition of KS statistic in Equation 1, an h -subset S is qualified if and only if $\forall i (1 \leq i \leq q)$

$$|F_R(x_i) - F_{T \setminus S}(x_i)| \leq c_\alpha \sqrt{\frac{n + m - h}{n * (m - h)}}.$$

Since $F_R(x_i) = \frac{\mathbf{C}_R[i]}{n}$ and $F_{T \setminus S}(x_i) = \frac{\mathbf{C}_T[i] - \mathbf{C}_S[i]}{m - h}$, we have

$$\left| \frac{\mathbf{C}_R[i]}{n} - \frac{\mathbf{C}_T[i] - \mathbf{C}_S[i]}{m - h} \right| \leq c_\alpha \sqrt{\frac{n + m - h}{n * (m - h)}}.$$

After simplification, we have

$$\Gamma(i, h) - \Omega(h) \leq \mathbf{C}_S[i] \leq \Gamma(i, h) + \Omega(h).$$

Since $\mathbf{C}_S[i]$ is a non-negative integer, we immediately have

$$\lceil \Gamma(i, h) - \Omega(h) \rceil \leq \mathbf{C}_S[i] \leq \lfloor \Gamma(i, h) + \Omega(h) \rfloor. \quad (3)$$

Since $h - \mathbf{C}_S[i]$ and $m - \mathbf{C}_T[i]$ are the numbers of data points in S and T that are larger than x_i , respectively, and $S \subset T$, $h -$

$C_S[i] \leq m - C_T[i]$ holds, that is, $h - m + C_T[i] \leq C_S[i]$. Since $C_S[i-1] \leq C_S[i]$, Equation 2a holds.

Since $C_S[i] - C_S[i-1]$ and $C_T[i] - C_T[i-1]$ are the numbers of times x_i appears in S and T , respectively, and $S \subset T$, $C_S[i] - C_S[i-1] \leq C_T[i] - C_T[i-1]$, that is, $C_S[i] \leq C_S[i-1] + C_T[i] - C_T[i-1]$. Using the righthand side of Equation 3 and $C_S[i] \leq h$ by definition, Equation 2b holds.

(Sufficiency) For any h -cumulative vector C_S that satisfies Equations 2a and 2b, we construct a set S such that for each i ($1 \leq i \leq q$), data point x_i appears in S ($C_S[i] - C_S[i-1]$) times. Since $C_S[i]$ and $C_S[i-1]$ satisfy the inequality $C_S[i] - C_S[i-1] \leq C_T[i] - C_T[i-1]$, the number of times x_i appearing in S is smaller than or equal to the number of times x_i appearing in T . Plugging $C_T[q] = m$ into Equation 2a, we have $h \leq C_S[q]$. Since $C_S[q]$ also satisfies Equation 2b, we have $h = C_S[q]$. From the way that S is constructed, we know that S has $C_S[q]$ elements. Therefore, S is an h -subset of T .

We show that S is a qualified h -subset. Since C_S satisfies Equation 2, for each i ($1 \leq i \leq q$), $C_S[i]$ satisfies Equation 3. Plugging Equation 3 into $F_{T \setminus S}$, we have $\forall i (1 \leq i \leq q)$, $|F_R(x_i) - F_{T \setminus S}(x_i)| \leq c\alpha \sqrt{\frac{n+m-h}{n*(m-h)}}$. This means R and $T \setminus S$ can pass the KS test, thus S is a qualified h -subset of T and C_S is a qualified h -cumulative vector. The sufficiency follows. \square

Lemma 1 transforms conducting the KS test to checking Equations 2a and 2b. Given h ($1 \leq h \leq m-1$), Equations 2a and 2b recursively give a lower bound and an upper bound of each element $C[i]$ ($1 \leq i \leq q$) of an h -cumulative vector C , respectively. The lower bound and the upper bound of $C[i]$ depend on the lower bound and the upper bound of $C[i-1]$, respectively.

Denote by l_i^h and u_i^h the lower bound and the upper bound of $C[i]$ in any qualified h -cumulative vector C . We compute l_1^h and u_1^h by plugging $C[0] = 0$ into Equations 2a and 2b. Then, we plug $C[1] = l_1^h$ into Equation 2a and $C[1] = u_1^h$ into Equation 2b to compute the lower bound l_2^h and the upper bound u_2^h of $C[2]$, respectively. By iteratively plugging $C[i-1] = l_{i-1}^h$ into Equation 2a and $C[i-1] = u_{i-1}^h$ into Equation 2b, we can compute the lower bound and the upper bound of every $C[i]$ of qualified h -cumulative vectors C . The closed form formulae of l_i^h and u_i^h ($1 \leq i \leq q$) are

$$l_i^h = \max(\lceil M(i, h) - \Omega(h) \rceil, h - m + C_T[i], 0) \quad (4a)$$

$$u_i^h = \min(\lceil \Gamma(i, h) + \Omega(h) \rceil, C_T[i], h) \quad (4b)$$

where $M(i, h) = \max_{j=1}^i \{\Gamma(j, h)\}$. We define $l_0^h = u_0^h = 0$, as $C[0] = 0$ is a constant.

Now, we use the lower bounds and the upper bounds of $C[1], C[2], \dots, C[q]$ to derive a sufficient and necessary condition for the existence of a qualified h -cumulative vector C as follows.

THEOREM 1. *Given the KS test with a reference set R and a test set T , for h ($1 \leq h \leq m-1$), there exists a qualified h -cumulative vector if and only if for each i ($1 \leq i \leq q$), $l_i^h \leq u_i^h$.*

PROOF. (Necessity) Since l_i^h and u_i^h are the lower bound and the upper bound of $C[i]$, respectively, the necessity is straightforward.

(Sufficiency) Assuming for each i ($1 \leq i \leq q$), $l_i^h \leq u_i^h$, we construct a qualified h -cumulative vector C as follows. We start by

setting $C[q] = u_q$, and then for i iterating from q to 1, we choose an integer $C[i-1]$ from $[l_{i-1}^h, u_{i-1}^h]$, such that

$$0 \leq C[i] - C[i-1] \leq C_T[i] - C_T[i-1].$$

Now we show that such an integer $C[i-1]$ always exists. Since l_i^h is derived by setting $C[i-1] = l_{i-1}^h$ in Equation 2a and u_i^h is derived by setting $C[i-1] = u_{i-1}^h$ in Equation 2b, we have $l_{i-1}^h \leq l_i^h$ and $u_i^h \leq u_{i-1}^h + C_T[i] - C_T[i-1]$. Since $l_i^h \leq C[i] \leq u_i^h$, we have

$$l_{i-1}^h \leq C[i] \leq u_{i-1}^h + C_T[i] - C_T[i-1].$$

Since $l_{i-1}^h \leq u_{i-1}^h$ and they are integers, there exists an integer $C[i-1] \in [l_{i-1}^h, u_{i-1}^h]$, such that $0 \leq C[i] - C[i-1] \leq C_T[i] - C_T[i-1]$. Thus, an h -cumulative vector C can be constructed by iteratively applying the above operations to set up elements in C .

Last, we prove that C is a qualified h -cumulative vector by showing that for each i ($1 \leq i \leq q$), $C[i]$ satisfies Equations 2a and 2b. According to the definition of an h -cumulative vector, we have $C[i-1] \leq C[i]$. By Equation 4a, we have $h - m + C_T[i] \leq l_i^h$ and $\lceil \Gamma(i, h) - \Omega(h) \rceil \leq l_i^h$. Since $l_i^h \leq C[i]$, $C[i]$ satisfies Equation 2a. By Equation 4b, we have $u_i^h \leq h$ and $u_i^h \leq \lceil \Gamma(i, h) + \Omega(h) \rceil$. According to how $C[i-1]$ is selected, $C[i]$ and $C[i-1]$ satisfy $C[i] - C[i-1] \leq C_T[i] - C_T[i-1]$. Since $C[i] \leq u_i^h$, $C[i]$ satisfies Equation 2b. The sufficiency follows Lemma 1 immediately. \square

According to Theorem 1, we can efficiently check the existence of a qualified h -cumulative vector by checking the q pairs of lower bounds and upper bounds, $(l_1^h, u_1^h), \dots, (l_q^h, u_q^h)$. Each pair of bounds can be computed and checked in $O(1)$ time. Since $q \leq n+m$, the time complexity of checking the existence of a qualified h -cumulative vector is $O(n+m)$.

Since the existence of a qualified h -cumulative vector is equivalent to the existence of a qualified h -subset, we can tackle the first efficiency bottleneck by checking the q pairs of lower bounds and upper bounds. This reduces the time complexity of checking the existence of a qualified h -subset from $O(\binom{m}{h}(m+n-h) \log(m+n-h))$ to $O(n+m)$.

To find the size of interpretations, for each subset size h ($1 \leq h \leq m-1$), we need to apply Theorem 1 to check the existence of a qualified h -cumulative vector. Therefore, the overall time complexity of finding the size of interpretations is $O(m(m+n))$. Next, we further reduce the time complexity to $O((m+n) \log(m) + (m+n)(k - \hat{k}))$, where \hat{k} is a lower bound on the size of interpretations k .

4.4 Finding a Lower Bound on Interpretation Size by Binary Search

To tackle the second efficiency bottleneck, in this subsection, we develop a technique to find a lower bound on the size of interpretations in $O((m+n) \log m)$ time. Using this technique, to find the size of interpretations, we only need to check the subset sizes that are larger than or equal to the lower bound.

To reduce the number of subset sizes to be checked, we propose a necessary condition for the existence of a qualified h -cumulative vector with respect to h . The necessary condition has a nice monotonicity with respect to h . If an integer h ($1 \leq h \leq m-2$) satisfies the condition, all integers from $h+1$ to $m-1$ also satisfy the necessary condition. Thus, we can leverage this property to find a lower bound

\hat{k} of the interpretation size k by a binary search in $O((m+n) \log m)$ time. The lower bound reduces the search range of k from $[1, k]$ to $[\hat{k}, k]$. This helps us further reduce the complexity of phase 1 in MOCHI from $O(m(n+m))$ to $O((n+m) \log(m) + (k-\hat{k})(n+m))$.

THEOREM 2. *Given the KS test with a reference set R and a test set T , for h ($1 \leq h \leq m-1$), there exists a qualified h -cumulative vector only if for each i ($1 \leq i \leq q$), the following holds.*

$$0 \leq \lfloor \Gamma(i, h) + \Omega(h) \rfloor \quad (5a)$$

$$\lceil M(i, h) - \Omega(h) \rceil \leq h \quad (5b)$$

$$M(i, h) - \Omega(h) \leq \Gamma(i, h) + \Omega(h) \quad (5c)$$

Moreover, if Equation 5 holds for any h , then it also holds for $h+1$.

PROOF. We first prove the necessary condition. Since there exists a qualified h -cumulative vector, by Theorem 1, for each i ($1 \leq i \leq q$), $l_i^h \leq u_i^h$. l_i^h and u_i^h are the maximum and the minimum of the three terms in Equations 4a and 4b, respectively. Thus, every term in u_i^h is larger than or equal to every term in l_i^h . Therefore, we immediately have Equations 5a and 5b, as well as the following.

$$\lceil M(i, h) - \Omega(h) \rceil \leq \lfloor \Gamma(i, h) + \Omega(h) \rfloor \quad (6)$$

Since $M(i, h) - \Omega(h) \leq \lceil M(i, h) - \Omega(h) \rceil$ and $\lfloor \Gamma(i, h) + \Omega(h) \rfloor \leq \Gamma(i, h) + \Omega(h)$, Equation 5c follows Equation 6 immediately.

Next, we prove the monotonicity of the necessary condition with respect to h . For each inequality in Equation 5, we show that for each i ($1 \leq i \leq q$), if a size h ($1 \leq h \leq m-2$) satisfies the inequality, the size $h+1$ also satisfies the inequality.

Equation 5a: Plugging the definitions of $\Gamma(i, h)$ and $\Omega(h)$ into Equation 5a, the inequality can be simplified to $\frac{C_T[i]}{m-h} - \frac{C_R[i]}{n} \geq -c\alpha\sqrt{\frac{1}{m-h} + \frac{1}{n}}$. Since $-c\alpha\sqrt{\frac{1}{m-h} + \frac{1}{n}} > -c\alpha\sqrt{\frac{1}{m-h-1} + \frac{1}{n}}$ and $\frac{C_T[i]}{m-h-1} \geq \frac{C_T[i]}{m-h}$, we have $\frac{C_T[i]}{m-h-1} - \frac{C_R[i]}{n} > -c\alpha\sqrt{\frac{1}{m-h-1} + \frac{1}{n}}$, which can be simplified to $0 \leq \lfloor \Gamma(i, h+1) + \Omega(h+1) \rfloor$.

Equation 5b: Plugging the definition of $M(i, h)$ into Equation 5b, we have $\lceil \Gamma(j, h) - \Omega(h) \rceil \leq h$, for each integer j ($1 \leq j \leq i$). Plugging the definitions of $\Gamma(j, h)$ and $\Omega(h)$ into the inequality, the inequality can be simplified to $\frac{C_T[j]-h}{m-h} - \frac{C_R[j]}{n} \leq c\alpha\sqrt{\frac{1}{m-h} + \frac{1}{n}}$. Since $c\alpha\sqrt{\frac{1}{m-h} + \frac{1}{n}} < c\alpha\sqrt{\frac{1}{m-h-1} + \frac{1}{n}}$ and $\frac{C_T[j]-h-1}{m-h-1} < \frac{C_T[j]-h}{m-h}$, we have $\frac{C_T[j]-h-1}{m-h-1} - \frac{C_R[j]}{n} \leq c\alpha\sqrt{\frac{1}{m-h-1} + \frac{1}{n}}$, which can be simplified to $\lceil \Gamma(j, h+1) - \Omega(h+1) \rceil \leq h$. Since h is an integer, we immediately have $\lceil \Gamma(j, h+1) - \Omega(h+1) \rceil \leq h$. Applying the definition of $M(i, h)$, we have $\lceil M(i, h+1) - \Omega(h+1) \rceil \leq h$.

Equation 5c: According to the definition of $M(i, h)$, from Equation 5c, we have $\Gamma(j, h) - \Omega(h) \leq \Gamma(i, h) + \Omega(h)$, for each integer j ($1 \leq j \leq i$). Plugging the definitions of $\Omega(h)$ and $\Gamma(j, h)$ into the inequality, the inequality can be simplified to $-2c\alpha\sqrt{\frac{1}{m-h} + \frac{1}{n}} \leq \frac{C_T[i]-C_T[j]}{m-h} - \frac{1}{n}(C_R[i] - C_R[j])$. Since $\frac{C_T[i]-C_T[j]}{m-h-1} > \frac{C_T[i]-C_T[j]}{m-h}$ and $-2c\alpha\sqrt{\frac{1}{m-h} + \frac{1}{n}} > -2c\alpha\sqrt{\frac{1}{m-h-1} + \frac{1}{n}}$, we have $-2c\alpha\sqrt{\frac{1}{m-h-1} + \frac{1}{n}} < \frac{C_T[i]-C_T[j]}{m-h-1} - \frac{1}{n}(C_R[i] - C_R[j])$, which can be simplified to $\Gamma(j, h+1) - \Omega(h+1) \leq \Gamma(i, h+1) + \Omega(h+1)$. Applying the definition of $M(i, h)$, we have $M(i, h+1) - \Omega(h+1) \leq \Gamma(i, h+1) + \Omega(h+1)$. \square

The smallest integer \hat{k} that satisfies the necessary condition in Theorem 2 is a lower bound on the size k of the interpretations. We do not need to check any h -subset smaller than \hat{k} , as they are guaranteed not to contain a qualified h -cumulative vector. Based on the monotonicity of Equation 5 with respect to h , we can apply binary search to find the smallest integer \hat{k} that satisfies Theorem 2. For $h \in [1, m-1]$, it takes $O(n+m)$ time to verify the q groups of inequalities in Theorem 2, because $q \leq n+m$. Therefore, the overall time complexity of finding \hat{k} is $O((m+n) \log m)$. Once \hat{k} is found, we iteratively use Theorem 1 to find the exact size of interpretations. The overall time complexity of finding the exact size of interpretation is $O((m+n) \log m + (m+n)(k-\hat{k}))$, where k is the exact size. In the worst case, $k - \hat{k} = O(m)$, and the complexity is still $O(m(m+n))$. However, as verified by our experiments, $k - \hat{k}$ is often a very small number and our technique can significantly improve the efficiency of searching the size of interpretations.

5 GENERATING MOST COMPREHENSIBLE INTERPRETATIONS

Given the size of interpretations k , the brute force method takes $O(\binom{m}{k}(m+n-k) \log(m+n-k))$ time to find the most comprehensible interpretation by enumerating the k -subsets of T . In this section, we develop a method to directly construct the most comprehensible interpretation in $O(m(n+m))$ time without enumerating the k -subsets.

An h -subset $S \subset T$ is called an h -partial interpretation if there exists an interpretation that is a superset of S . When it is clear from the context, we also call S a *partial interpretation* for short.

According to Definition 1, the most comprehensible interpretation is the interpretation that has the smallest lexicographical order. This property facilitates the design of our construction algorithm. Our algorithm scans the data points in T in the order of L and selects the first data point x_{i_1} that is in an interpretation, that is, x_{i_1} is a 1-partial interpretation. Since x_{i_1} is the first such data point in L , the most comprehensible interpretation must contain x_{i_1} , otherwise we have the contradiction that the interpretation containing x_{i_1} precedes the most comprehensible interpretation in the lexicographical order. Then, the algorithm continues to scan the points after x_{i_1} in L , still in the order of L , and finds the next data point x_{i_2} such that $\{x_{i_1}, x_{i_2}\}$ are part of an interpretation, that is, $\{x_{i_1}, x_{i_2}\}$ is a 2-partial interpretation. Clearly, $\{x_{i_1}, x_{i_2}\}$ is part of the most comprehensive interpretation. The search continues until k points are obtained, which is the most comprehensive interpretation. The construction method is summarized in Algorithm 1.

Now, the remaining question is how we can determine whether an h -subset S is a partial interpretation. We first establish that a subset S is a partial interpretation if and only if there exists a qualified k -cumulative vector, which satisfies a small group of inequalities derived from S . Then, we introduce a sufficient and necessary condition for the existence of such a k -cumulative vector, which can be efficiently checked in $O(n+m)$ time.

LEMMA 2. *Given the KS test with a reference set R and a test set T , for a subset $S \subset T$, S is a partial interpretation if and only if there exists a qualified k -cumulative vector C , such that the following*

Algorithm 1: Find the most comprehensible interpretation

Input: a reference set R , a test set T , a significant level α , a preference list L , the size of interpretations k

Output: I := the most comprehensible interpretation

```
1 Initialize  $I \leftarrow \emptyset$ 
2  $T \leftarrow$  sort the data points in  $T$  in the order of  $L$ 
3 for  $i \leftarrow 1; i \leq |T|; i++$  do
4   if  $I \cup \{T[i]\}$  is a partial interpretation then
5      $I \leftarrow I \cup \{T[i]\}$ 
6   end
7   if  $|I| = k$  then
8     return  $I$ 
9   end
10 end
```

inequality holds for $1 \leq i \leq q$,

$$C[i] - C[i-1] \geq C_S[i] - C_S[i-1] \quad (7)$$

PROOF. (Necessity) Since S is a partial interpretation, by definition, there exists an interpretation \mathcal{I} such that $S \subseteq \mathcal{I}$. Denote by C the qualified k -cumulative vector of \mathcal{I} . For each i ($1 \leq i \leq q$), since $C[i] - C[i-1]$ and $C_S[i] - C_S[i-1]$ are the numbers of times x_i appearing in \mathcal{I} and S , respectively, and $S \subseteq \mathcal{I}$, $C[i] - C[i-1] \geq C_S[i] - C_S[i-1]$ holds. The necessity follows.

(Sufficiency) Assume a qualified k -cumulative vector C that satisfies Equation 7. Let \mathcal{I} be the interpretation corresponding to C . For each data point $x_i \in S$, x_i appears $C_S[i] - C_S[i-1]$ times in S . Due to Equation 7, x_i appears in \mathcal{I} the same or more number of times. Thus, $S \subseteq \mathcal{I}$ and S is a partial interpretation. \square

Next, we derive a sufficient and necessary condition for the existence of such a k -cumulative vector C by investigating the lower bound and the upper bound of each element $C[i]$. Since C is a qualified k -cumulative vector, by Theorem 1, $l_i^k \leq C[i] \leq u_i^k$. Equation 7 can be rewritten as $C[i-1] \leq C[i] - C_S[i] + C_S[i-1]$. That is, the upper bound of $C[i-1]$ depends on the upper bound of $C[i]$. Denote by $\bar{l}_i = l_i^k$ a lower bound of $C[i]$ and by \bar{u}_i an upper bound of $C[i]$. Since $C[i] \leq \bar{u}_i$ and $C[i-1] \leq u_{i-1}^k$, about the upper bounds we have, for i ($1 \leq i \leq q$),

$$\bar{u}_{i-1} = \min(u_{i-1}^k, \bar{u}_i - C_S[i] + C_S[i-1]). \quad (8)$$

Given the size of interpretations k , we first compute u_i^k for each i ($1 \leq i \leq q$) by Equation 4b. Then, we iteratively compute \bar{u}_i for each i ($0 \leq i \leq q$). We define $\bar{u}_q = u_q^k$ and plug \bar{u}_q into Equation 8 to compute \bar{u}_{q-1} , and iteratively compute the upper bound of each $C[i]$ of a qualified k -cumulative vector C that satisfies Equation 7.

Since \bar{u}_i depends on u_i^k , once the size of interpretations k is determined using the techniques developed in Section 4, we can use the lower bound \bar{l}_i and the upper bound \bar{u}_i to derive a sufficient and necessary condition for the existence of a qualified k -cumulative vector C that satisfies Equation 7 as stated in the following result, and thus decide whether an h -subset S is a partial interpretation.

THEOREM 3. Given the KS test with a reference set R and a test set T , for a subset $S \subset T$, there exists a qualified k -cumulative vector C that satisfies Equation 7 if and only if for each i ($0 \leq i \leq q$), $\bar{l}_i \leq \bar{u}_i$.

PROOF. (Sufficiency) Given S , assume for each i ($0 \leq i \leq q$), $\bar{l}_i \leq \bar{u}_i$. We construct a k -cumulative vector C such that for each i ($0 \leq i \leq q$), $C[i] = \bar{u}_i$. We show that C is a qualified k -cumulative vector and also satisfies Equation 7.

We first prove that C is a qualified k -cumulative vector by showing that $C[0] = 0$, and each $C[i]$ ($1 \leq i \leq q$) satisfies Equations 2a and 2b. Since $l_0^k = \bar{l}_0 \leq \bar{u}_0 = u_0^k = 0$, we have $\bar{l}_0 = \bar{u}_0 = 0$ and thus $C[0] = 0$. Plugging $C[i-1] = \bar{u}_{i-1}$ and $C[i] = \bar{u}_i$ into Equation 8, we have $C[i-1] \leq C[i]$. Since $\bar{l}_i = l_i^k \leq C[i]$, from Equation 4a, we have $[\Gamma(i, h) - \Omega(h)] \leq C[i]$ and $h - m + C_T[i] \leq C[i]$. Therefore, $C[i]$ satisfies Equation 2a.

Plugging $C[i-1] = \bar{u}_{i-1}$ into Equation 8, the value of $C[i-1]$ falls into one of the following two cases.

Case 1: $C[i-1] = u_{i-1}^k$. As u_i^k is derived by plugging $C[i-1] = u_{i-1}^k$ into Equation 2b, we have $u_i^k \leq C_T[i] - C_T[i-1] + u_{i-1}^k$. Since $C[i] \leq u_i^k$ and $C[i-1] = u_{i-1}^k$, we have $C[i] \leq C_T[i] - C_T[i-1] + C[i-1]$.

Case 2: $C[i-1] = \bar{u}_i - C_S[i] + C_S[i-1]$. Since $S \subset T$, $C_S[i] - C_S[i-1] \leq C_T[i] - C_T[i-1]$. Since $C[i] = \bar{u}_i$, we have $C[i] \leq C_T[i] - C_T[i-1] + C[i-1]$.

Since $C[i] = \bar{u}_i \leq u_i^k$, from Equation 4b, we have $C[i] \leq k$ and $C[i] \leq [\Gamma(i, k) + \Omega(k)]$. Therefore, $C[i]$ satisfies Equation 2b. By Lemma 1, C is a qualified k -cumulative vector.

Since for each i ($0 \leq i \leq q$), $C[i] = \bar{u}_i$, from Equation 8, we can derive that every $C[i]$ satisfies Equation 7.

(Necessity) Given S , assume a qualified k -cumulative vector C that satisfies Equation 7. Since \bar{l}_i and \bar{u}_i are the lower and the upper bound of $C[i]$, respectively, the necessity follows immediately. \square

Given an interpretation size k and a subset S , it takes $O(m+n)$ time to verify the $q+1$ groups of inequalities in Theorem 3, because $q \leq n+m$. Since for each data point t_i in T , we need to check whether $I \cup \{t_i\}$ is a partial interpretation, where I is the partial interpretation found so far, the overall time complexity of constructing the most comprehensible interpretation is $O(m(n+m))$.

As shown in Section 4, it takes $O((m+n) \log(m)) + O((n+m)(k - \hat{k})) = O((m+n)(\log m + k - \hat{k}))$ time to identify the interpretation size. In total, our method takes $O(m(n+m))$ time to find the most comprehensible interpretation for a failed KS test.

6 EXPERIMENTS

In this section, we evaluate the effectiveness of most comprehensible counterfactual interpretations and the efficiency and scalability of MOCHI. We describe the datasets and the settings of the experiments in Section 6.1. Counterfactual interpretations have two fundamental requirements, which are being small and being contrastive [30]. In Section 6.2, we evaluate the size of our interpretations. In Section 6.3, we evaluate whether our interpretations are contrastive, that is, they can reverse failed KS tests into passed ones. In Section 6.4, we use case studies to investigate whether our interpretations can effectively explain the causes of failed KS tests. Last, in Section 6.5, we verify the efficiency and scalability of our proposed method.

Dataset	# Time series	Length	# Abnormal observations
AWS	17	1,243 ~ 4,700	124 ~ 472
AD	6	1,538 ~ 1,624	150 ~ 162
TRF	7	1,127 ~ 2,500	112 ~ 248
TWT	10	15,831 ~ 15,902	1,429 ~ 1,584
KC	7	1,882 ~ 22,695	188 ~ 2,268
ART	6	4032	402

Table 2: Some statistics of the datasets.

6.1 Datasets and Experiment Settings

We conduct experiments using 6 univariate time series datasets in the Numenta Anomaly Benchmark (NAB) repository [28]. Each dataset contains 6 to 10 time series and each time series contains 1,000 to 20,000 observations. For each time series, the ground truth labels of abnormal observations are available. The AWS server metrics (AWS) dataset contains the time series of the CPU Utilization, Network Bytes In, and Disk Read Bytes of an AWS server. The online advertisement clicks (AD) dataset contains the time series of online advertisement clicking rates and cost per thousand impressions. The freeway traffic (TRF) dataset contains the time series of occupancy, speed, and travel time of freeway traffics collected by specific sensors. The Tweets (TWT) dataset contains the time series of numbers of Twitter mentions of publicly-traded companies such as Google, IBM, and Apple. The miscellaneous known causes (KC) dataset contains the time series from multiple domains, including machine temperature, number of NYC taxi passengers, and CPU usage of an AWS server. The artificial (ARF) dataset contains the artificially-generated time series with varying types of anomalies. Table 2 shows some statistics of the datasets.

We run a sliding window of W size w to obtain the reference set, and use the window of the same size following W immediately but without any overlap as the test set. The KS test is conducted multiple times as the sliding windows run through a time series. A failed KS test indicates that the time series has a distribution drift [24]. We interpret the failed KS test to find the data points that cause the drift. The significant level of the KS test is always set to 0.05 following the convention in statistical testing. We use a variety of window sizes, including 100, 200, 300, 1,000, 1,500, and 2,000.

In our experiments, we apply two widely used time series outlier detection methods, Spectral Residual [48] and Bitmap [58] to automatically generate the preference lists L of data points in the test sets. A preference list L reflects a user’s domain knowledge about data abnormality. Data points with larger outlying scores are ranked higher in L . We use the published Python codes of Spectral Residual² and Bitmap³ with the default parameters.

Baselines. To the best of our knowledge, interpreting failed KS tests has not been studied in literature. Therefore, to evaluate the performance of MOCHI, we design three baseline methods, namely Greedy, Extended-CornerSearch, and Extended-GRACE.

Greedy generates a counterfactual interpretation \mathcal{I} by greedily selecting the first l data points in L such that R and $T \setminus \mathcal{I}$ can pass the KS test. When the preference list is generated by an outlier detection method, Greedy can be regarded as an extension of the outlier detection method to interpret failed KS tests.

Extended-CornerSearch is extended from CornerSearch [15], a state-of-the-art L_0 -norm adversarial attack method on image classifiers. CornerSearch generates adversarial images by randomly searching a small portion of the top- K important pixels of input images and masking them to 0 or 1. Although CornerSearch is not proposed to interpret failed KS tests, it may be extended to serve the purpose. The Extended-CornerSearch treats data points as pixels and perturbs the selected data points \mathcal{I} by removing them from T . Ideally, CornerSearch keeps the set of pixels perturbed as small as possible. Thus, after applying each perturbations, it conducts the KS test on R and $T \setminus \mathcal{I}$ to check if \mathcal{I} is an interpretation, meaning passing the KS test so that the unchanged part is regarded as normal by classifiers. When the context is clear, we call this baseline method **CornerSearch** for short.

Extended-GRACE is a direct extension from GRACE [30], the state-of-the-art counterfactual interpretation method on neural networks. To interpret a prediction on an input vector \mathbf{x} , GRACE perturbs the most important K features of \mathbf{x} , which are ranked by an external method, to change the prediction. GRACE only accepts vectors as inputs and generates interpretations by minimizing a target classifier’s prediction scores. Correspondingly, we extend GRACE to interpret failed KS tests by first accommodating the inconsistency between the inputs of GRACE and our problem through a mapping from an m -dimensional vector \mathbf{x} to a subset $S \subseteq T$, where $m = |T|$. We project \mathbf{x} to its nearest 0-1 vector and put the i -th data point t_i into S if the i -th element of the vector is 1. Next, we extend the objective function of GRACE to find interpretations on failed KS tests by perturbing \mathbf{x} to minimize $g(\mathbf{x}) = \sqrt{\frac{n \cdot (m - |S|)}{n + (m - |S|)}} D(R, T \setminus S)$, where S is the set of data points picked by vector \mathbf{x} . Based on the definition of the KS test, S is an interpretation on the failed KS test if $g(\mathbf{x})$ is smaller than the critical value c_α . Since $g(\mathbf{x})$ is not differentiable, we adopt the zeroth order optimization algorithm in [13] to solve the problem. We skip the entropy-based feature selection step used in GRACE, as it requires access to training data of classifiers, which is not available in our problem setting. When the context is clear, we call this baseline method **GRACE** for short.

Parameter Settings. We adopt the same parameter setting used in [15] for CornerSearch. For GRACE, we set $K = 100$ to be consistent with CornerSearch and set the remaining parameters to the same as [30]. We use the same parameters as [13] for the zeroth optimization algorithm used in GRACE.

We implement all algorithms using Python. All experiments are conducted on a server with two Xeon(R) Silver 4114 CPUs (2.20GHz), four Tesla P40 GPUs, 400GB main memory, and a 1.6TB SSD running Centos 7 OS. Our source code is published on GitHub <https://github.com/research0610/MOCHI>.

Since CornerSearch and GRACE are too slow to compute interpretations for all failed KS tests in 24 hours, for each combination of time series and window size, we uniformly sample 10 failed KS tests, where the test sets contain the corresponding ground truth abnormal observations. We conduct all experiments on the sampled 2, 690 failed KS tests.

²<https://github.com/SeldonIO/alibi-detect>

³<https://github.com/linkedin/luminol>

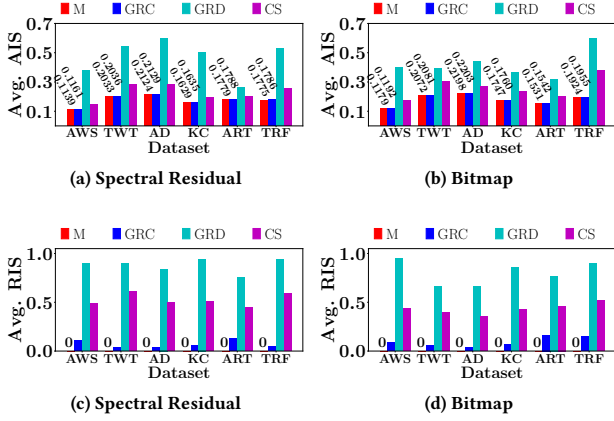


Figure 2: The average AIS and RIS of all methods. Smaller AIS and RIS values indicate better performance. “M”, “GRC”, “GRD” and “CS” are short for MOCHI, GRACE, Greedy and CornerSearch, respectively. We give the average AIS and RIS values of some methods on top of the corresponding bars for the ease of reading.

6.2 Are the Most Comprehensible Interpretations Small?

Small interpretations help users focus on predominant factors in a decision [57]. Therefore, being small is a key measurement of counterfactual interpretations [30, 36].

To comprehensively evaluate the performance of the compared methods in producing small counterfactual interpretations, we measure the size of the interpretations by two metrics. The *Absolute Interpretation Size (AIS)* $AIS = \frac{|I|}{|T|}$ is the proportion of the removed data points in the test set. The *Relative Interpretation Size (RIS)* measures whether an interpretation is smaller than its counterparts. For the interpretations produced by all methods on the same failed KS test, the RIS of the smallest interpretation is 0, which is good, and the RIS of the other interpretations is 1.

We evaluate MOCHI, Greedy, GRACE, and CornerSearch in AIS and RIS on the failed KS tests of all datasets. To observe the influence of the preference lists on AIS and RIS, we test each method with two different preference lists generated by Spectral Residual [48] and Bitmap [58], respectively. GRACE and CornerSearch cannot find counterfactual interpretations for some failed KS tests. To fairly compare the methods, among the 2,690 failed KS tests in those datasets, in this experiment we only consider the 847 ones (31.4%) where all methods can generate counterfactual interpretations. Figure 2 shows the average AIS and RIS of all interpretations.

Greedy and CornerSearch do not perform well. Both methods generate interpretations by taking the first several data points in the preference lists until the picked data points reverse the KS tests. However, since the preference lists are generated by the methods independent from the KS test, some data points that are not the causes of failed KS tests may still be ranked high in the preference lists. As a result, those two methods select many data points irrelevant to the failure of the KS test and lead to unnecessarily large interpretations.

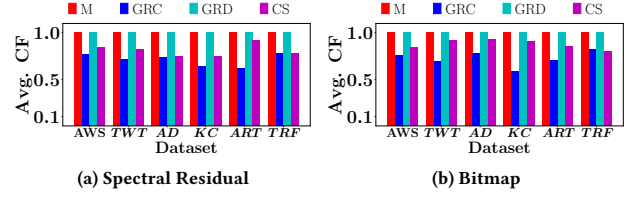


Figure 3: The average CF of all methods. A large value indicates good performance. “M”, “GRC”, “GRD” and “CS” are short for MOCHI, GRACE, Greedy and CornerSearch, respectively.

As a counterfactual interpretation method, GRACE generates interpretations by solving an optimization problem, which allows it to re-rank the data points based on their effects on the KS tests. Therefore, as shown in Figure 2, GRACE finds smaller interpretations than the other two baselines. However, as shown in Figures 2c and 2d, GRACE still cannot guarantee to find the smallest interpretations, because its objective function is non-differentiable and hard to minimize. MOCHI has the smallest average AIS and RIS and guarantees to find the smallest interpretation.

6.3 Are the Most Comprehensible Interpretations Contrastive?

Contrastivity is a fundamental requirement for counterfactual interpretations [30, 36]. A counterfactual interpretation on a failed KS test should reverse the failed KS test into a passed one. In this subsection, we quantitatively evaluate the performance of the methods in providing contrastive interpretations.

We define the *change of a failed KS test (CF)* to measure whether an interpretation I can change a failed KS test to a passed one. If R and $T \setminus I$ can pass a KS test, $CF=1$ and the interpretation I is contrastive. Otherwise, $CF=0$ and I is not contrastive.

Same as Section 6.2, we test using two different preference lists. To limit the runtime of GRACE and CornerSearch to reasonable, we constrain the two methods to only generate interpretations using the top-100 ranked data points in the preference lists L . In other words, GRACE and CornerSearch abort if a failed KS test does not have a counterfactual interpretation that only contains the top-100 data points. To fairly compare the methods, in this experiment, we only count the 1,293 (48.1%) among the 2,690 failed KS tests where Grace and CornerSearch do not abort. Figure 3 shows the average CF of all methods.

CornerSearch and GRACE perform poorly on all datasets. The smaller average CF values of those two methods indicate that they cannot find counterfactual interpretations for a large number of failed KS tests. GRACE does not perform well, as its non-differential objective function is hard to minimize, and thus cannot efficiently find a good solution to its optimization problem. One may improve the average CF of GRACE by more optimization steps. However, as to be shown in Section 6.5, GRACE is very slow, and more optimization steps make its even slower. The performance of CornerSearch varies heavily on different preference lists. The top-ranked data points in a preference list are much more likely to be sampled by

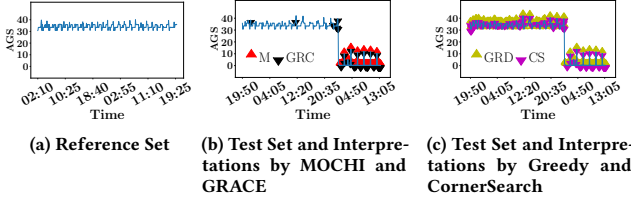


Figure 4: The interpretations on a failed KS test conducted on the AGS time series. Different markers indicate the data points chosen as interpretations by different methods. “M”, “GRC”, “GRD” and “CS” are short for MOCHI, GRACE, Greedy and CornerSearch, respectively.

CornerSearch [15]. Therefore, if the top-ranked data points are not the causes of a failed KS test, CornerSearch may repeatedly sample a set of data points irrelevant to the failure of the KS test. Even though Greedy has a good average CF, as shown in Figure 2, the method tends to find very large interpretations, which are hard for people to understand [30, 36].

The average CF of MOCHI is always 1 on all datasets and on both preference lists, because MOCHI guarantees to produce the most comprehensible counterfactual interpretations.

6.4 Are the Most Comprehensible Interpretations Effective?

Good interpretations should be easy to understand by human beings [14]. In this subsection, we first conduct two case studies to illustrate the effectiveness of the most comprehensible interpretations. Then, we quantitatively evaluate the effectiveness of the interpretations generated by MOCHI and the three baseline methods.

In the first case study, we examine the interpretations on a failed KS test conducted on two segments of a time series from the AWS dataset in NAB [28] about auto group scaling (AGS) of a computer cluster, measured by the number of running EC2 servers. At each time point, the AGS of the computer cluster is measured. Figure 4a shows the reference window, which is right before the test window (Figure 4b). Each window has 500 data points. The interpretations on the failed KS test generated by different methods are shown in Figures 4b and 4c with different markers. The sliding windows and the time series segments in Figures 4b and 4c are identical. We present the results in two figures using different markers for different methods for the ease of viewing. The preference list L of the data points is generated by Spectral Residual [48].

Figure 4b shows that, at around 20:35, the AGS value abnormally drops to a very low level. A significant decrease in the customer application workload causes many EC2s being terminated. The distribution change of AGS causes the failure of the KS test. MOCHI correctly identifies the time period that the cluster has abnormally low workload. The interpretation provided by GRACE also highlights a similar time period as MOCHI. However, it also selects several data points before the drop when the cluster works normally. Those extra points in the interpretation may confuse users about the cause of the failed KS test. As shown in Figure 4c, the interpretations generated by Greedy and CornerSearch contain

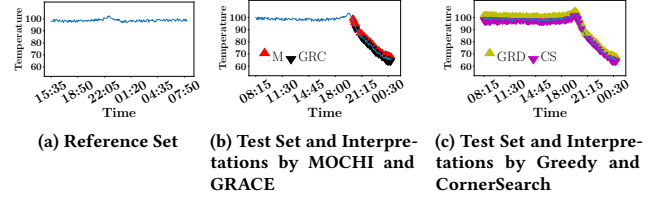


Figure 5: The interpretations on a failed KS test conducted on the machine temperature time series. Different markers indicate the data points chosen as interpretations by different methods. “M”, “GRC”, “GRD” and “CS” are short for MOCHI, GRACE, Greedy and CornerSearch, respectively.

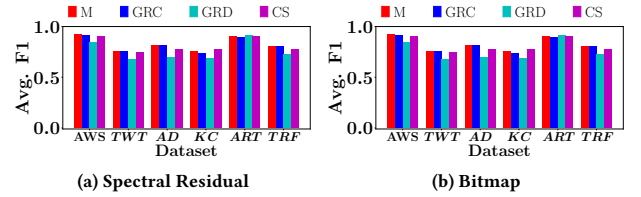


Figure 6: The average F1 of all methods. “M”, “GRC”, “GRD” and “CS” are short for MOCHI, GRACE, Greedy and CornerSearch, respectively.

almost all data points in the test set and thus are not informative about the real cause of the abnormality.

Our second case study examines the interpretations on a failed KS test conducted on two segments of a time series about the temperature of a large industrial machine, which is from the KC dataset in NAB [28]. At each time point, the temperature of the machine is measured. Figure 5a shows the reference window right before the test window (Figure 5b). Each sliding window has 200 data points. The interpretations on the KS test generated by different methods are highlighted in Figures 5b and 5c with their corresponding markers. The preference list L of the data points in T is generated by Spectral Residual [48].

Figure 5a shows that normally the machine runs at a high temperature. Figure 5b shows that, at around 21:15, the machine has a catastrophic system failure that causes a shutdown of the machine [28], and the machine temperature decreases slowly after 21:15. The change of temperature causes the failure of the KS test. MOCHI and GRACE correctly identify the period of abnormal temperatures. However, although GRACE picks fewer points than MOCHI, the set of points picked by GRACE cannot reverse the failed KS test. As shown in Figure 5c, Greedy and CornerSearch cannot produce meaningful interpretations on the failed KS test.

To quantitatively evaluate the effectiveness of the generated interpretations, we compare all methods in identifying the ground truth abnormal observations in the datasets. A good interpretation should include abnormal points in the test sets. We treat the interpretation generated by each method as the abnormal observations detected and compute the F1 score of each method as [61]. The average F1 of all methods are reported in Figure 6.

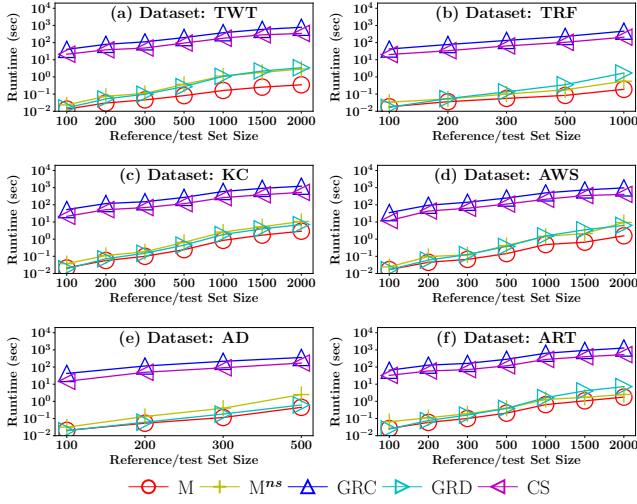


Figure 7: The average runtime of all methods. The runtime is plotted in logarithmic scale. “M”, “M^{ns}”, “GRC”, “GRD” and “CS” are short for MOCHI, MOCHI^{ns}, GRACE, Greedy and CornerSearch, respectively.

Greedy performs worst among all methods, since the interpretations generated by Greedy include many data points that are irrelevant to the failure of KS tests. CornerSearch performs better than Greedy, because CornerSearch generates interpretations that are much shorter than Greedy. The performances of MOCHI and GRACE are comparable. They outperform both Greedy and CornerSearch on almost all datasets, because they are specifically designed to find counterfactual interpretations.

6.5 Efficiency and Scalability

In this subsection, we report the runtime of all methods. In addition, to evaluate the effectiveness of our pruning techniques, we implement a lower-bound ablation MOCHI^{ns} by disabling the pruning using the lower bound of the interpretation size (Section 4.4).

We vary the sliding window size, which is also the size of reference sets and test sets. As explained in Section 6.1, for a given sliding window size, there are multiple failed KS tests. Figure 7 shows the average runtime of all methods.

MOCHI constantly outperforms all baselines on all datasets. The runtime of all methods increases when the test sets become larger. MOCHI is 3 orders of magnitudes faster than GRACE and CornerSearch.

The poor performance of GRACE, CornerSearch, and Greedy is due to the cost of conducting huge numbers of KS tests. In each optimization step of GRACE, the zeroth order optimization algorithm needs to conduct m KS tests to estimate the gradient of the objective function with respect to the m -dimensional input vector. Therefore, GRACE needs to conduct $l \cdot m$ KS tests to find an interpretation, where l is the number of optimization steps. According to the parameter settings in [30], in the worst case, GRACE has to perform $l = 10,000$ steps. CornerSearch needs to conduct a KS test to verify whether a sampled set of data points is an interpretation. Comparing to the huge sample space of size $O(2^m)$, the

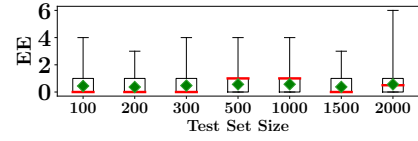


Figure 8: The estimation errors (EE) of the interpretation size of all failed KS tests. Each bar in the figure shows the estimation errors on the KS tests with a specific size of test sets. The upper and lower edges of a box show the first and third quartiles of the estimation errors, respectively. The upper and lower ends of an error bar shows the maximum and minimum estimation errors, respectively. The red line segment in a box and the green diamond marker show the median and the mean of the estimation errors, respectively.

number of interpretations is very small. Therefore, CornerSearch has to generate a large number of samples to find an interpretation, which takes a long time to verify. In the worst case, according to the parameter settings in [15], CornerSearch has to generate 150,000 random samples. Since Greedy needs to conduct the KS test after removing each data point, the KS test needs to be conducted $O(m)$ times to find an interpretation. Since each time the KS test takes $O((m+n) \log(m+n))$ time, the time complexity of Greedy is $O(m * (m+n) \log(m+n))$. Since our estimated lower bound on the size k of interpretations effectively reduces the search range of k , MOCHI interprets failed KS tests faster than MOCHI^{ns}.

To investigate the tightness of the lower bound on the size of interpretations, we also report the *estimation error* (EE), which is the difference between the estimated lower bound \hat{k} and the true interpretation size k . A small value of EE indicates that our estimated lower bound is tight. Figure 8 shows the results with respect to different sizes of test sets by box plot [59]. For more than 25% of the failed KS tests, our estimated lower bound \hat{k} is the same as the true value of k . For more than 75% of the failed KS tests, the estimation errors are 1 or 0. In the worst case, our estimation error is only 6, which happens on a KS test with 2,000 data points in the test set. Considering the large size of the test set, 6 is a very small value. Besides, we observe that when the test sets become larger, the average value of estimation errors is always smaller than 1. The results seem to suggest that estimation errors may be treated as a constant in practice. This result is consistent with our observation in Figure 7 that MOCHI is more efficient than MOCHI^{ns} in interpreting failed KS tests.

7 CONCLUSIONS

In this paper, we tackle the novel problem of producing counterfactual interpretations on failed KS tests. We propose the notion of most comprehensible counterfactual interpretation, and develop a two-phase algorithm, MOCHI, which guarantees to find the most comprehensible interpretation fast. We report extensive experiments demonstrating the superior capability of MOCHI in efficiently interpreting failed KS tests. As future work, we plan to extend MOCHI to interpret failed KS tests conducted on multidimensional data points [18, 46].

REFERENCES

- [1] Charu C Aggarwal. 2015. Outlier analysis. In *Data mining*. Springer, 237–263.
- [2] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. 2004. Order preserving encryption for numeric data. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. 563–574.
- [3] Arjun R Akula, Shuai Wang, and Song-Chun Zhu. 2020. CoCoX: Generating Conceptual and Counterfactual Explanations via Fault-Lines. In *AAAI*. 2594–2601.
- [4] Fabrizio Angiulli and Clara Pizzuti. 2002. Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery*. Springer, 15–27.
- [5] André Artelt and Barbara Hammer. 2019. Efficient computation of counterfactual explanations of LVQ models. *arXiv preprint arXiv:1908.00735* (2019).
- [6] Idir Benouaret, Sihem Amer-Yahia, and Senjuti Basu Roy. 2019. An Efficient Greedy Algorithm for Sequence Recommendation. In *International Conference on Database and Expert Systems Applications*. Springer, 314–326.
- [7] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2017. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248* (2017).
- [8] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 93–104.
- [9] Miles Brundage, Shahar Avin, Jasmine Wang, Haydn Belfield, Gretchen Krueger, Gillian Hadfield, Heidy Khlaaf, Jingying Yang, Helen Toner, Ruth Fong, et al. 2020. Toward trustworthy AI developments: mechanisms for supporting verifiable claims. *arXiv preprint arXiv:2004.07213* (2020).
- [10] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics* 8, 8 (2019), 832.
- [11] Hongsong Chen, Caixia Meng, Zhiguang Shan, Zhongchuan Fu, and Bharat K Bhargava. 2019. A novel Low-rate Denial of Service attack detection approach in ZigBee wireless sensor network by combining Hilbert-Huang Transformation and Trust Evaluation. *IEEE Access* 7 (2019), 32853–32866.
- [12] Xuefeng Chen, Yifeng Zeng, Gao Cong, Shengchao Qin, Yanping Xiang, and Yuanshun Dai. 2015. On information coverage for location category based point-of-interest recommendation. In *29th AAAI Conference on Artificial Intelligence*.
- [13] Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. 2018. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457* (2018).
- [14] Zicun Cong, Lingyang Chu, Lanjun Wang, Xia Hu, and Jian Pei. 2020. Exact and Consistent Interpretation of Piecewise Linear Models Hidden behind APIs: A Closed Form Solution. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 613–624.
- [15] Francesco Croce and Matthias Hein. 2019. Sparse and imperceivable adversarial attacks. In *Proceedings of the IEEE International Conference on Computer Vision*. 4724–4732.
- [16] Zhiguo Ding and Minrui Fei. 2013. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proceedings Volumes* 46, 20 (2013), 12–17.
- [17] Denis Moreira dos Reis, Peter Flach, Stan Matwin, and Gustavo Batista. 2016. Fast unsupervised online drift detection using incremental kolmogorov-smirnov test. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1545–1554.
- [18] G. Fasano and A. Franceschini. 1987. A multidimensional version of the Kolmogorov-Smirnov test. *Monthly Notices of the Royal Astronomical Society* 225, 1 (03 1987), 155–170. <https://doi.org/10.1093/mnras/225.1.155> [arXiv:https://academic.oup.com/mnras/article-pdf/225/1/155/18522274/mnras225-0155.pdf](https://academic.oup.com/mnras/article-pdf/225/1/155/18522274/mnras225-0155.pdf)
- [19] Ruth C Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*. 3429–3437.
- [20] Markus Goldstein and Andreas Dengel. 2012. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track* (2012), 59–63.
- [21] Xiaoyi Gu, Leman Akoglu, and Alessandro Rinaldo. 2019. Statistical analysis of nearest neighbor methods for anomaly detection. In *Advances in Neural Information Processing Systems*. 10923–10933.
- [22] Michael Hay, Jerome Miklau, David Jensen, Don Towsley, and Philipp Weis. 2008. Resisting structural re-identification in anonymized social networks. *Proceedings of the VLDB Endowment* 1, 1 (2008), 102–114.
- [23] Fabian Keller, Emmanuel Muller, and Klemens Böhm. 2012. HiCS: High contrast subspaces for density-based outlier ranking. In *2012 IEEE 28th international conference on data engineering*. IEEE, 1037–1048.
- [24] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. 2004. Detecting change in data streams. In *VLDB*, Vol. 4. Toronto, Canada, 180–191.
- [25] Jerome Klotz. 1967. Asymptotic efficiency of the two sample Kolmogorov-Smirnov test. *J. Amer. Statist. Assoc.* 62, 319 (1967), 932–938.
- [26] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. In *Advances in neural information processing systems*. 4066–4076.
- [27] Ashwin Lall. [n.d.]. Data streaming algorithms for the Kolmogorov-Smirnov test. In *2015 IEEE International Conference on Big Data (Big Data)*. IEEE, 95–104.
- [28] Alexander Lavin and Subutai Ahmad. 2015. Evaluating Real-Time Anomaly Detection Algorithms—The Numenta Anomaly Benchmark. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 38–44.
- [29] Aleksandar Lazarevic and Vipin Kumar. 2005. Feature bagging for outlier detection. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 157–166.
- [30] Thai Le, Suhang Wang, and Dongwon Lee. 2020. GRACE: Generating Concise and Informative Contrastive Sample to Explain Neural Network Model’s Prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Virtual Event, CA, USA) (KDD ’20)*. Association for Computing Machinery, New York, NY, USA, 238–248. <https://doi.org/10.1145/3394486.3403066>
- [31] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. 2018. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering* 31, 12 (2018), 2346–2363.
- [32] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (2019), 1–38.
- [33] Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. 2019. Sparsefool: a few pixels make a big difference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9087–9096.
- [34] Christoph Molnar. 2019. *Interpretable Machine Learning*. <https://christophm.github.io/interpretable-ml-book/>.
- [35] Jonathan Moore, Nils Hammerla, and Chris Watkins. 2019. Explaining deep learning models with constrained adversarial examples. In *Pacific Rim International Conference on Artificial Intelligence*. Springer, 43–56.
- [36] Raha Moraffah, Mansoor Karami, Ruocheng Guo, Adrienne Raglin, and Huan Liu. 2020. Causal Interpretability for Machine Learning-Problems, Methods and Evaluation. *ACM SIGKDD Explorations Newsletter* 22, 1 (2020), 18–33.
- [37] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 607–617.
- [38] Sigurd Kirkevold Næss. 2012. Application of the Kolmogorov-Smirnov test to CMB data: Is the universe really weakly random? *Astronomy & Astrophysics* 538 (2012), A17.
- [39] Raymond S Nickerson. 1998. Confirmation bias: A ubiquitous phenomenon in many guises. *Review of general psychology* 2, 2 (1998), 175–220.
- [40] Mila Nikolova. 2013. Description of the Minimizers of Least Squares Regularized with ℓ_0 -norm. Uniqueness of the Global Minimizer. *SIAM Journal on Imaging Sciences* 6, 2 (2013), 904–937.
- [41] Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B Gibbons, and Christos Faloutsos. 2003. Loci: Fast outlier detection using the local correlation integral. In *Proceedings 19th international conference on data engineering (Cat. No. 03CH37405)*. IEEE, 315–326.
- [42] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277* (2016).
- [43] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. 506–519.
- [44] Judea Pearl. 2018. Theoretical impediments to machine learning with seven sparks from the causal revolution. *arXiv preprint arXiv:1801.04016* (2018).
- [45] Fábio Pinto, Marco OP Sampaio, and Pedro Bizarro. 2019. Automatic Model Monitoring for Data Streams. *arXiv preprint arXiv:1908.04240* (2019).
- [46] Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. 2019. Failing loudly: An empirical study of methods for detecting dataset shift. In *Advances in Neural Information Processing Systems*. 1396–1408.
- [47] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 427–438.
- [48] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time-Series Anomaly Detection Service at Microsoft. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3009–3017.
- [49] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “Why should I trust you?” Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.
- [50] Ron Rymon. 1992. Search through Systematic Set Enumeration. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (Cambridge, MA) (KR’92)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 539–550.

- [51] Ricardo Jorge Santos, Jorge Bernardino, and Marco Vieira. 2014. Approaches and challenges in database intrusion detection. *ACM Sigmod Record* 43, 3 (2014), 36–47.
- [52] Sebastian Schelter, Tammo Rukat, and Felix Biessmann. 2020. Learning to Validate the Predictions of Black Box Classifiers on Unseen Data. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1289–1299.
- [53] Kacper Sokol and Peter A Flach. 2019. Counterfactual explanations of machine learning predictions: opportunities and challenges for AI safety. In *SafeAI@AAAI*.
- [54] Sebastian Tschiatschek, Adish Singla, and Andreas Krause. 2017. Selecting Sequences of Items via Submodular Maximization.. In *AAAI*. 2667–2673.
- [55] Arnaud Van Looveren and Janis Klaise. 2019. Interpretable counterfactual explanations guided by prototypes. *arXiv preprint arXiv:1907.02584* (2019).
- [56] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.* 31 (2017), 841.
- [57] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y Lim. 2019. Designing theory-driven user-centric explainable AI. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–15.
- [58] Li Wei, Nitin Kumar, Venkata Lolla, Eamonn J Keogh, Stefano Lonardi, and Chotirat Ratanamahatana. 2005. Assumption-free anomaly detection in time series. In *Proceedings of the 17th international conference on Scientific and statistical database management*. 237–240.
- [59] David F Williamson, Robert A Parker, and Juliette S Kendrick. 1989. The box plot: a simple visual method to interpret data. *Annals of internal medicine* 110, 11 (1989), 916–921.
- [60] Min Xie, Laks VS Lakshmanan, and Peter T Wood. 2010. Breaking out of the box of recommendations: from items to packages. In *Proceedings of the fourth ACM conference on Recommender systems*. 151–158.
- [61] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. 2018. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*. 187–196.