Clustering Categorical Data via Multiple Hypothesis Testing

LIANYU HU and MUDI JIANG, School of Software, Dalian University of Technology, Dalian, China
YAN LIU, School of Software Engineering, Dalian University, Dalian, China
QUAN ZOU, Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China
ZENGYOU HE, School of Software, Dalian University of Technology, Dalian, China

Categorical data clustering is a fundamental data mining problem, which has been extensively studied during the past decades. To date, many effective clustering algorithms for categorical data are available in the literature. However, almost all existing categorical data clustering algorithms did not address the issue of the statistical significance of detected clusters. In particular, how to assess the statistical significance of a set of non-overlapping categorical clusters still remains unaddressed. In this article, we formulate the categorical data clustering problem as a multiple hypothesis testing problem, where the null hypothesis is that each attribute is independent of the given partition of clusters. Then, all individual p-values from different attributes are integrated to obtain a consensus p-value through statistical meta-analysis. Thereafter, a significance-based clustering algorithm is proposed in which the combined p-value is efficiently optimized in an indirectly and incremental manner. Experimental results on 25 real-world datasets demonstrate that our method is capable of achieving comparable performance to state-of-the-art categorical data clustering algorithms. Furthermore, our method has a good capability of determining whether there really exists a clustering structure and assessing whether a given set of clusters is statistically significant.

CCS Concepts: • **Computing methodologies** → **Cluster analysis**; • **Information systems** → **Clustering**;

Additional Key Words and Phrases: Clustering, Categorical data, Statistical significance, Multiple hypothesis testing, Meta-analysis

Associate Editor: Lingyang Chu

ACM Reference format:
Lianyu Hu, Mudi Jiang, Yan Liu, Quan Zou, and Zengyou He. 2025. Clustering Categorical Data via Multiple Hypothesis Testing. *ACM Trans. Knowl. Discov. Data.* 19, 5, Article 109 (June 2025), 31 pages.
https://doi.org/10.1145/3735977

This work was supported by the National Natural Science Foundation of China (Grant No. 62472064) and the Dalian Young Science and Technology Talent Support Program (Grant No. 2023RQ056).
Authors' Contact Information: Lianyu Hu, School of Software, Dalian University of Technology, Dalian, China; e-mail: hly4ml@gmail.com; Mudi Jiang, School of Software, Dalian University of Technology, Dalian, China; e-mail: 792145962@qq.com; Yan Liu, School of Software Engineering, Dalian University, Dalian, China; e-mail: yliu0414@qq.com; Quan Zou, Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China; e-mail: zouquan@nclab.net; Zengyou He (corresponding author), School of Software, Dalian University of Technology, Dalian, China; e-mail: zyhe@dlut.edu.cn.

ACM 1556-472X/2025/6-ART109
https://doi.org/10.1145/3735977

1 Introduction

Categorical datasets are widely accessible in many disciplines, such as bioinformatics [4] and social sciences [5]. Each attribute in this data type comprises a set of discrete values that are not quantitatively comparable. For example, eye color is a commonly used categorical variable that can take values {*Brown*, *Amber*, *Hazel*, *Green*, *Blue*, *Gray*}. As an important multivariate statistical technique, cluster analysis of categorical data aims to divide objects with categorical attributes into compact groups in an unsupervised manner [31]. To identify compact clusters from categorical datasets, many effective methods have been proposed, e.g., partitioning clustering [51], hierarchical clustering [87], and model-based clustering [26]. Among these existing categorical data clustering algorithms, partitioning clustering algorithms are probably the most widely investigated clustering methods [7, 89, 95].

Existing partitioning algorithms for categorical data produce a set of clusters primarily by optimizing an objective function [6] that is typically defined based on a specific (dis)similarity measure [15, 70, 89, 95]. More precisely, they usually employ a k-means-type objective function [8, 49, 51]. Thus, it becomes crucial for such algorithms to provide a powerful representation [76] or an appropriate distance measure [15] for categorical objects. While these types of clustering algorithms are effective in dealing with complex categorical data, it is not guaranteed to produce meaningful clustering results. That is, they do not provide statistical robustness against spurious patterns, i.e., identifying whether the clustering results are statistically significant.

To date, many methods have been proposed to assess the statistical significance of clustering results based on different types of hypothesis testing procedures (e.g., [1, 18, 34, 38, 40, 48, 67, 72, 86]). Unfortunately, almost all of them are designed for numerical data rather than categorical data. That is, it is non-trivial or not feasible to extend these approaches to handle categorical data. For example, a cluster of numerical objects is typically modeled by a single multivariate Gaussian distribution [66], whereas to model categorical objects requires a technique that can portray a discrete distribution. One notable significance-based categorical data clustering method [88] tries to evaluate each individual categorical cluster separately. As a result, it cannot be directly applied to assess the standard clustering result of a set of non-overlapping categorical clusters.

In this article, we focus on validating whether a partition of non-overlapping categorical clusters is true or not in a statistically sound manner. Our research is motivated by the following key observations. First of all, although numerous categorical data clustering algorithms are available in the literature, all these existing algorithms are unable to judge whether a partition of categorical clusters is statistically meaningful. As a consequence, we may even report a set of categorical clusters from those datasets without a true clustering structure. Second, due to the discrete nature of categorical data, the statistical assessment of categorical clusters can be more difficult than its counterpart for numeric data. Finally, most categorical data clustering algorithms produce a set of non-overlapping clusters as the output. Therefore, assessing the statistical significance of a partition of non-overlapping categorical clusters would be more appealing than evaluating an individual categorical cluster in practice.

To our knowledge, there is still a lack of theory and practice in bringing statistical rigor to the evaluation of a partition of categorical clusters. In this article, we formulate the categorical data clustering issue as a **multiple hypothesis testing (MHT)** problem. More precisely, given a partition and its cluster label on a target dataset, we have the null hypothesis that each attribute is independent of the cluster label, i.e., this is a random partition and there is no clustering structure. For each attribute, we can derive a p-value to determine whether we should retain or reject the corresponding null hypothesis. Intuitively, if the given partition is not a meaningful clustering result, it can be expected that we should retain most of the null hypotheses, i.e., most attributes are

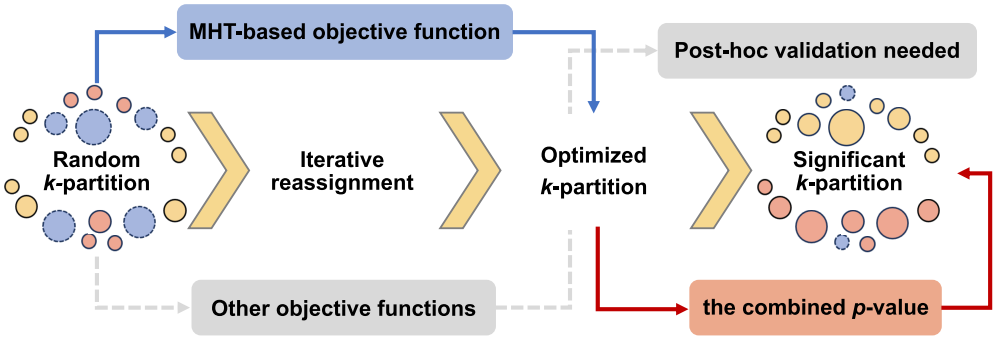


Fig. 1. The workflow of the proposed method.

independent of the target partition. Hence, we further borrow the idea from statistical meta-analysis [14] to integrate these individual p-values to obtain a combined p-value.

To demonstrate the validity of above theoretical formulation, we further derive a computationally feasible objective function as follows. We first employ the Chi-square test to obtain a p-value for determining whether the partition variable is independent of each attribute variable. Then, we use the rth order p-value method [81] to obtain a consensus p-value by combining individual p-values from different attributes. Since the combined p-value is hard to be optimized in practice, the **sum of Chi-square test statistics (SCS)** is employed as the objective function. Finally, a new categorical data clustering algorithm equipped with the new objective function is developed. Empirical studies on real datasets demonstrate the effectiveness and computational efficiency of our method.

In summary, the workflow (as illustrated in Figure 1) and the main contributions of this work can be summarized as follows:

- The categorical data clustering problem is formulated as a MHT issue. To the best of our knowledge, this is the first attempt that tackles the cluster analysis problem from a MHT aspect.
- A combined p-value based on meta-analysis is derived, which is the first method to directly calculate the statistical significance of a partition of categorical objects. The p-value has the potential to be a universal metric for evaluating categorical clustering results, regardless of the specific clustering algorithm used.
- A new clustering algorithm by using iterative updating in an incremental manner is proposed. Experimental results on real categorical datasets show that the presented method is comparable to **state-of-the-art (SOTA)** categorical data clustering methods in terms of both **accuracy (ACC)** and running time.

The remaining parts of this article are organized as follows. Section 2 reviews the works that are closely related to our method. Section 3 describes our method in detail. Experiments on real datasets are given in Section 4. Finally, we conclude this article in Section 5.

2 Related Work

To assess the statistical significance of a set of non-overlapping clusters, existing significance-based clustering methods are only available for numerical datasets rather than categorical datasets. Thus, we discuss categorical data clustering algorithms and significance-based clustering methods separately.

2.1 Categorical Data Clustering

Existing methods for extracting clusters from categorical data can be roughly divided into the following categories: partitioning method, hierarchical method [42, 87], density-based method [16, 39], model-based method [22, 26], and ensemble method [53, 93]. Since partitioning algorithms have been extensively investigated and our method falls into this category as well, we will elaborate partitioning algorithms for clustering categorical data in this section. In general, existing partitioning algorithms can be divided into two types, depending on whether the objective function requires pairwise distances or not.

2.1.1 Within-Cluster Sum of Squares (WSS)-Based Objective Function. The objective function is defined based on the WSS, i.e., the sum of (squared) distance between each object and its cluster center [51, 55]. Existing partitioning algorithms based on WSS can be further categorized into two groups: (1) The objective function is defined on the original categorical data. (2) The objective function is defined after categorical data embedding.

(1) The K-modes algorithm [51] and its extensions [23, 85] define the WSS objective function based on the Hamming distance between two categorical objects and the mode is employed as the cluster center. Therefore, subsequent research efforts toward this direction mainly focus on deploying a more appropriate distance function [11, 15] or developing a more expressive center representation. For instance, the distance measure is further enhanced by imposing different weights on different attributes or different attribute values [2, 49, 56, 57, 70, 89, 91]. Meanwhile, the development of new center representations instead of modes for categorical clusters is widely studied as well [8, 24].
(2) The embedding method typically converts categorical vectors into numerical ones, then k-means or other clustering methods designed for numerical data can be utilized. That is, the WSS objective function in the transformed data space is implicitly adopted when k-means-type algorithms are used to obtain the clustering result. In the process of categorical data embedding, those methods learn complex characteristics such as hierarchical couplings [58, 59, 95], reconstructed features based on the co-occurrence probability of objects [73, 94] and integrated vectors derived from a graph representation [7, 90].

2.1.2 Non-WSS-Based Objective Function. The objective function in this category does not rely on an explicit distance measure and cluster representative. Two classic objective functions are the *Entropy-based function* [10, 25, 62] and the *Category Utility (CU) function* [35, 65, 69], which measure the compactness of each cluster based on the distribution of categorical values within each cluster. However, these objective functions are not derived based on a rigorous significance testing procedure, which cannot be directly employed for assessing the statistical significance of a partition.

2.2 Significance-Based Clustering

To quantify the level of clusterability [1, 44] in terms of p-value, significance-based clustering methods incorporate statistical rigor and hypothesis testing procedure into cluster analysis [27, 40]. Significance-based clustering methods can be classified into two basic types, depending on the target clustering structure [13]: partition or hierarchy. In significance-based hierarchical clustering methods, some testing procedures are developed to guide the cutting [18, 61] or merging process [40, 83] during the process for constructing the dendrogram, i.e., to produce a statistically significant nested partition. The significance-based partitioning methods directly determine the statistical significance of each single cluster or a set of k clusters. Since our method is also a partitioning method, we will focus on related methods toward this direction. More precisely, existing significance-based

partitioning algorithms can be further divided into two categories according to the null hypothesis [12] and null model [41]:

(1) *Unimodality Hypothesis.* Null models in this category assume that the data samples are drawn from a unimodal distribution [77]. If the given data exhibit a clustering structure, the alternative hypothesis states that the data follows a bimodal distribution (e.g., Dip statistic [28]) or a multimodal distribution (e.g., Silverman's test [78]). In this case, the null hypothesis is rejected. To obtain a statistically significant partition of k clusters, we can either sequentially test the unimodality of single clusters (subsets of the data) or derive a p-value on the k-partition. When testing single clusters, i.e., whether the cluster has significant binary splits, some methods based on the extension of Dip statistics are proposed [21, 60, 68], and the unimodal distribution is typically specified as a single multivariate Gaussian distribution [48, 66]. As for testing the k-partition, the Silverman's test is usually employed [3], where k can be estimated by using an orthounimodal reference distribution [43]. In addition, the Gaussian mixture distribution [29, 38, 74] is used to test the k-partition in terms of separation on mean vectors.

(2) *Uniformity Hypothesis.* In contrast to the Unimodality hypothesis that data samples are gathered around a mode in a single cluster, this hypothesis states that data samples are purely randomly distributed in a given region. That is, all data samples have the same probability to appear at each location within the region, which can be described by the so-called homogeneity [33, 34]. Specifically, the randomness can be characterized through homogeneous Poisson process [9, 50, 79, 86], **minimum spanning tree (MST)** [54, 72, 80], random graph [63, 64], and random partition [37]. If a k-partition is statistically significant, then any homogeneous clusters are spatially separated from each other, or the k-partition deviates from its reference k homogeneous groups [67]. When testing the k-partition, the determination of whether there is "no gap" between neighboring homogeneous clusters is widely employed as the key procedure. More precisely, existing methods utilize the ratio between two masses [34] and the two-sample test based on MST [36, 54] to test if there is "no gap."

All the above-mentioned null models and corresponding clustering methods are developed for numerical data, which are not appropriate for categorical data since none of them are based on discrete probabilistic models. To the best of our knowledge, only a few significance-based clustering methods for categorical data have been proposed, including those by Zhang et al. [88] and, more recently, by Hu et al. [46, 47]. The hypothesis testing procedure in [88] focuses on assessing the statistical significance of an individual categorical cluster, with the DV algorithm sequentially extracting statistically significant clusters, if they exist. The interpretable clustering method in [47] aims to assess the statistical significance of each candidate split point during decision-tree construction, with the SigDT algorithm forming categorical clusters that correspond to leaf nodes in a top-down manner. However, neither of these methods can be directly applied to assess the statistical significance of a given candidate k-partition. Notably, the clusterability test designed for categorical data in [45] assesses the statistical significance of clustering tendency rather than clustering result.

3 Methods

3.1 Problem Formulation

Suppose we have M attributes in a categorical dataset $X = [X_1, \cdots, X_M]$. For each attribute X_m, we use x_m to denote the corresponding categorical variable with Q_m distinct values $\{a_1, \cdots, a_{Q_m}\}$. In cluster analysis, N objects in X are divided into K non-overlapping clusters, and π is used to denote the cluster label of an object. Obviously, π is a categorical variable with K distinct values

$\{c_1, \cdots, c_K\}$. For each object o_i $(1 \le i \le N)$, we have an observed pair (x_m^i, π^i) for the mth attribute $(1 \le m \le M)$ where x_m^i and π^i are sampled from categorical variables x_m and π, respectively.

In this article, we solve the categorical data clustering problem from a significance testing perspective. Under the null hypothesis that the clustering structure is not present, i.e., no clusters exist in the categorical dataset X, we know that the partition variable π is independent of each attribute variable x_m. More precisely, we use $\theta_m = 0$ and $\theta_m \neq 0$ to denote whether π is independent of the mth attribute x_m, respectively. Then, the problem of assessing whether a given set of clusters exist can be formulated as MHT issue:

$$H_0 : \bigcap_{1 \le m \le M} \{\theta_m = 0\} \text{ versus } H_a : \sum_{m=1}^{M} I\{\theta_m \neq 0\} \ge r, \tag{1}$$

where the null hypothesis states that the partition is statistically independent of each attribute and the alternative hypothesis requires that π should have a statistical correlation with at least r attributes.

In fact, the same and other related significance testing issues in the form of Equation (1) have been widely investigated in the field of statistical meta-analysis [14, 20, 71, 84]. The basic idea works as follows: we first obtain a p-value for each individual statistical test and then combine these M p-values via meta-analysis. After employing a well-calibrated p-value combination procedure, we obtain a final single p-value to determine the acceptance or rejection of the null hypothesis. In our context, we can use the combined p-value as an objective function to guide the search of meaningful categorical clusters. That is, the categorical data clustering problem is to find a partition π such that the combined p-value is minimized.

3.2 MHT-Based Objective Function

To derive an objective function via MHT, we first use the Chi-square test to determine whether a partition variable is independent of each attribute variable in terms of p-values. Then, we obtain a single p-value using a meta-analysis approach that combines M p-values. Since directly optimizing the combined p-value is challenging, and extremely small p-values may exceed the limits of computational precision, the SCS is employed as the MHT-based objective function and is iteratively optimized in practice.

3.2.1 Chi-Square Test. The Chi-square test is a natural choice for quantifying the independence relationship between two categorical variables. In our setting, these two random variables correspond to the partition variable π and the attribute variable x_m. The test statistic χ^2 is obtained by comparing observed frequencies with expected frequencies under the null hypothesis that π and x_m are independent.

For an observed pair (x_m, π), we have N samples $o_i(x_m^i, \pi^i)$ with $i = 1, 2, \cdots, N$ in X. We use $\mathrm{N}_{qk}^{(m)}$ to denote the observed frequency when the qth $(1 \le q \le Q_m)$ attribute value of x_m falls into the kth $(1 \le k \le K)$ cluster. That is, we count the number of observed o_i with $(x_m^i = a_q, \pi^i = c_k)$ in a cell (q, k) of $Q_m \times K$ contingency table shown in Table 1. Then, we can calculate the corresponding expected frequency for each cell as follows:

$$\mathrm{E}_{qk}^{(m)} = \left(\sum_{q=1}^{Q_m} \mathrm{N}_{qk}^{(m)} \cdot \sum_{k=1}^{K} \mathrm{N}_{qk}^{(m)} \right) / N = (\mathrm{N}_{\cdot k} \cdot \mathrm{N}_{q\cdot}^{(m)}) / N, \tag{2}$$

Table 1. The Contingency Table for the Chi-Square Test on the Independence between x_m and π

	c_1	c_2	$\cdots$	c_K	Total
a_1	$\mathrm{N}_{11}^{(m)}$	$\mathrm{N}_{12}^{(m)}$	$\mathrm{N}_{1k}^{(m)}$	$\mathrm{N}_{1K}^{(m)}$	$\mathrm{N}_{1\cdot}^{(m)}$
a_2	$\mathrm{N}_{21}^{(m)}$	$\mathrm{N}_{22}^{(m)}$	$\mathrm{N}_{2k}^{(m)}$	$\mathrm{N}_{2K}^{(m)}$	$\mathrm{N}_{2\cdot}^{(m)}$
$\vdots$	$\mathrm{N}_{q1}^{(m)}$	$\mathrm{N}_{q2}^{(m)}$	$\mathrm{N}_{qk}^{(m)}$	$\mathrm{N}_{qK}^{(m)}$	$\mathrm{N}_{q\cdot}^{(m)}$
a_{Q_m}	$\mathrm{N}_{Q_m1}^{(m)}$	$\mathrm{N}_{Q_m2}^{(m)}$	$\mathrm{N}_{Q_mk}^{(m)}$	$\mathrm{N}_{Q_mK}^{(m)}$	$\mathrm{N}_{Q_m\cdot}^{(m)}$
Total	$\mathrm{N}_{\cdot 1}$	$\mathrm{N}_{\cdot 2}$	$\mathrm{N}_{\cdot k}$	$\mathrm{N}_{\cdot K}$	

where $\mathrm{N}_{\cdot k}$ and $\mathrm{N}_{q\cdot}^{(m)}$ are total frequency counts in c_k and a_q, respectively. The Chi-square test statistic has the following form:

$$\chi_m^2(X_m, \pi) = \sum_{k=1}^{K} \sum_{q=1}^{Q_m} \frac{\left(\mathrm{N}_{qk}^{(m)} - \mathrm{E}_{qk}^{(m)}\right)^2}{\mathrm{E}_{qk}^{(m)}}. \tag{3}$$

As N increases, the probability distribution of χ^2 follows the Chi-square distribution asymptotically with $(Q_m - 1) \cdot (K - 1)$ degrees of freedom. Given X and a partition π, we can derive M p-values by using the Chi-square test for all M attributes. Each p-value can be used to determine whether there is a correlation between the partition and the corresponding attribute.

We illustrate how to calculate the p-value by taking the Loan Data in Table 2 as an example. The Loan Data is a categorical dataset $X = [Sex, Age, Credit]$ with $N = 7, M = 3$. Suppose we divided the data into non-overlapped clusters with $K = 2$ and the *Status* is a partition variable π with $c_1 = Approved$ and $c_2 = Unapproved$, the contingency table for the correlation test between π and attribute variable *Age* $\{a_1 = Young, a_2 = Middle, a_3 = Older\}$ is shown in Table 3. According to the derived p-value, if the significance level is specified to be 0.05, then we can determine that there is a correlation between the attribute variable *Age* (x_2) and the specified partition variable *Status* (π). For other attributes, we have $\chi^2 = 0.1944$, p-value $= 0.6592$ and $\chi^2 = 7$, p-value $= 0.0302$ for (*Sex*, *Status*) and (*Credit*, *Status*), respectively. We can see that the partition π in Table 3 is statistically associated with two attributes when the significance level is 0.05. Hence, it is likely to be a partition that is composed of **ground-truth (GT)** clusters.

In contrast, for the partition π' with two clusters: $\{o_2, o_4, o_5\}$ and $\{o_1, o_3, o_6, o_7\}$, we have $\chi^2 = 1.2153$, p-value $= 0.2703$ for (*Sex*, *Status*), $\chi^2 = 4.2778$, p-value $= 0.1178$ for (*Age*, *Status*) and $\chi^2 = 0.8750$, p-value $= 0.6456$ for (*Credit*, *Status*). Obviously, this partition is statistically independent of each attribute under the significance level of 0.05 and we may claim that it is not a meaningful clustering result.

3.2.2 Meta-Analysis. To combine multiple p-values, we apply the rth ordered p-value (rOP) method [81] to solve our issue. We derive a single p-value by using rOP as follows: we first take the rth smallest p-value among M sorted p-values as the test statistic. According to [81], such a test statistic follows a Beta distribution in which two parameters are specified as: $\alpha = r, \beta = M - r + 1$.

Table 2. An Example Dataset: The Loan Data Consists of Seven Objects, Three Attributes, and Two Clusters

Loan ID (o_i)	Sex (x_1)	Age (x_2)	Credit (x_3)	Status (π)
Applicant 1	Female	Young	Good	Approved
Applicant 2	Male	Young	Fair	Approved
Applicant 3	Female	Young	Fair	Approved
Applicant 4	Male	Middle	Poor	Unapproved
Applicant 5	Female	Middle	Poor	Unapproved
Applicant 6	Male	Older	Poor	Unapproved
Applicant 7	Female	Older	Poor	Unapproved

Table 3. The Contingency Table for the Chi-Square Test on the Independence between Age (x_2) and Status (π) of the Loan Data

Observed Frequencies

	Approved (c_1)	Unapproved (c_2)	Total
Young (a_1)	3	0	3
Middle (a_2)	0	2	2
Older (a_3)	0	2	2
Total	3	4	

Expected Frequencies

	Approved (c_1)	Unapproved (c_2)	Total
Young (a_1)	9/7	12/7	3
Middle (a_2)	6/7	8/7	2
Older (a_3)	6/7	8/7	2
Total	3	4	

We obtain $\chi^2 = (3 - \frac{9}{7})^2/(\frac{9}{7}) + \cdots + (2 - \frac{8}{7})^2/(\frac{8}{7}) = 7$, and p-value = 0.0302 with $(3-1)\cdot(2-1) = 2$ df.

Accordingly, we can derive the final p-value for assessing the statistical significance of the target partition.

In the Loan Data, if we set $r = 2$ and then the second smallest p-value in the sorted list $[0.0302, 0.0302, 0.6592]$ for π in Table 2 will be the test statistic. The parameters for the corresponding Beta distribution are $\alpha = 2, \beta = 3 - 2 + 1 = 2$. Consequently, we can derive the final p-value as 0.0027. Similarly, for the partition π', the combined p-value will be 0.1797. That is, if we employ the combined p-value as the objective function, then π is better than π' and the former has a statistically significant clustering structure according to the hypothesis testing result.

However, it is not a good choice to directly optimize the combined p-value due to the following reasons: (1) Since we do not have a priori knowledge about how many attributes are statistically associated with a target partition, it is a dilemma for choosing the parameter r to combine p-values

with rOP. (2) Since the rth smallest p-value is an ordered statistic, this test statistic may be associated with different attributes for different partitions. As a result, it will be a non-trivial task to develop an efficient algorithm for searching a desirable partition when such a test statistic or its corresponding p-value is deployed as the objective function. To overcome the above limitations, we employ the SCS as the objective function.

3.2.3 The Sum of Test Statistics. Given any partition π with K clusters on a categorical dataset X, we denote the Chi-square test statistic for the mth attribute as $\chi^2_m(X_m, \pi)$. The MHT-based objective function named the SCS can be written as:

$$\mathrm{SCS}(X, \pi) = \sum_{m=1}^{M} \chi^2_m(X_m, \pi) = \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{q=1}^{Q_m} \frac{\left(\mathrm{N}_{qk}^{(m)} - \mathrm{E}_{qk}^{(m)}\right)^2}{\mathrm{E}_{qk}^{(m)}}. \tag{4}$$

To detect clusters from a given categorical dataset X, we can try to maximize the objective function in Equation (4). This is because: (1) A maximal test statistic corresponds to the minimal p-value for the Chi-square test for each attribute. (2) A GT partition is expected to be statistically correlated with most attributes. If the sum of all individual test statistics is maximized, then the objective of minimizing the rth smallest p-value in rOP can be partially achieved in an indirect manner.

To further show the rationale of the objective function and reveal its nature, we will provide a simplified version of Equation (4). According to Equation (2) and $\sum_{k=1}^{K} \sum_{q=1}^{Q_m} \mathrm{N}_{qk}^{(m)} = \sum_{k=1}^{K} \mathrm{N}_{\cdot k} = \sum_{q=1}^{Q_m} \mathrm{N}_{q\cdot}^{(m)} = \sum_{k=1}^{K} \sum_{q=1}^{Q_m} \mathrm{E}_{qk}^{(m)} = N$, we simplify $\mathrm{SCS}(X, \pi)$ in Equation (4) as follows:

$$\begin{aligned}
&\sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{q=1}^{Q_m} \left(\frac{\left(\mathrm{N}_{qk}^{(m)}\right)^2}{\mathrm{E}_{qk}^{(m)}} - 2 \cdot \mathrm{N}_{qk}^{(m)} + \mathrm{E}_{qk}^{(m)} \right) \\
&= \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{q=1}^{Q_m} \frac{\left(\mathrm{N}_{qk}^{(m)}\right)^2}{(\mathrm{N}_{\cdot k} \cdot \mathrm{N}_{q\cdot}^{(m)})/N} + \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{q=1}^{Q_m} \left(-2 \cdot \mathrm{N}_{qk}^{(m)} + \mathrm{E}_{qk}^{(m)} \right) \\
&= N \cdot \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{q=1}^{Q_m} \frac{\left(\mathrm{N}_{qk}^{(m)}\right)^2}{\mathrm{N}_{\cdot k} \cdot \mathrm{N}_{q\cdot}^{(m)}} - N \cdot M \\
&= N \cdot \sum_{k=1}^{K} \left(\frac{1}{\mathrm{N}_{\cdot k}} \cdot \sum_{m=1}^{M} \sum_{q=1}^{Q_m} \frac{\left(\mathrm{N}_{qk}^{(m)}\right)^2}{\mathrm{N}_{q\cdot}^{(m)}} \right) - N \cdot M \\
&= N \cdot \sum_{k=1}^{K} T_k - N \cdot M,
\end{aligned} \tag{5}$$

where $T_k = \frac{1}{\mathrm{N}_{\cdot k}} \cdot \sum_{m=1}^{M} \sum_{q=1}^{Q_m} \frac{\left(\mathrm{N}_{qk}^{(m)}\right)^2}{\mathrm{N}_{q\cdot}^{(m)}}$, $1 \le k \le K$.

The larger the SCS value, the more likely it is to obtain a set of compact categorical clusters. In the kth cluster of the target partition on X, we have a fixed cluster size $\mathrm{N}_{\cdot k}$ and an attribute value

distribution with $\mathrm{N}_{q\cdot}^{(m)}$ on each attribute. If there are more identical attribute values in each single cluster, T_k tends to be larger, i.e., each term of SCS function will be larger.

To formally establish the connection between our SCS objective function and the widely accepted criterion of within-cluster (intra-cluster) compactness, we first rewrite $\sum_{q=1}^{Q_m}(\mathrm{N}_{qk}^{(m)})^2/\mathrm{N}_{q\cdot}^{(m)}$ in T_k and define it as a compactness measure at the attribute level. This measure exclusively considers the variables affected by different cluster assignments and incorporates self-tuning weights related to frequency. We refer to this measure as **frequency-scaled compactness (FsC)**, which is defined as the dot product of the frequency vector $\mathbf{s}$ and the corresponding object number distribution vector $\mathbf{C}$:

$$\begin{aligned} FsC = \mathbf{s}\cdot\mathbf{C} &= \sum_{q=1}^{Q_m} s_q C_q = [s_1, s_2, \ldots, s_{Q_m}]\cdot[C_1, C_2, \ldots, C_{Q_m}] \\ &= \left[\frac{\mathrm{N}_{1k}^{(m)}}{\mathrm{N}_{1\cdot}^{(m)}}, \frac{\mathrm{N}_{2k}^{(m)}}{\mathrm{N}_{2\cdot}^{(m)}}, \ldots, \frac{\mathrm{N}_{Q_m k}^{(m)}}{\mathrm{N}_{Q_m\cdot}^{(m)}}\right]\cdot[\mathrm{N}_{1k}^{(m)}, \mathrm{N}_{2k}^{(m)}, \ldots, \mathrm{N}_{Q_m k}^{(m)}]. \end{aligned} \tag{6}$$

Since the Hessian matrix $\nabla^2\mathrm{FsC}$ is positive semidefinite, FsC is a convex function (see [17]). Given that the constraints $\sum_{q=1}^{Q_m} C_q = \mathrm{N}_{\cdot k}$ and $C_q \geq 0$ together define a standard Q_m-dimensional simplex, the maximum of FsC is attained at a vertex of the simplex, where exactly one C_q equals $\mathrm{N}_{\cdot k}$ and all others are zero. If there is only one category that exists, e.g., taking the qth category, we have

$$\mathrm{FsC} = \left[0, 0, \ldots, \frac{\mathrm{N}_{qk}^{(m)}}{\mathrm{N}_{q\cdot}^{(m)}}, \ldots, 0\right]\cdot[0, 0, \ldots, \mathrm{N}_{qk}^{(m)}, \ldots, 0] = s_q C_q. \tag{7}$$

In this cluster assignment, the convex function FsC reaches its maximum compactness. Consequently, T_k attains its maximum value. If each cluster contains exactly one unique category per attribute, the total summed T_k (SCS) also reaches its maximum. The same holds for the ***K*-modes objective function (KMF)**, where the qth category serves as the mode (i.e., the category with the highest frequency).

Next, for an arbitrary cluster assignment, we denote the mode as the q^*th category without loss of generality. The KMF-based compactness, which is inversely proportional to KMF-based distances (hereafter, KMF represents compactness), can be expressed as the number of objects in the attribute that match the mode, i.e., $\mathrm{KMF} = C_{q^*}$. Thus, FsC can be rewritten as:

$$\begin{aligned} \mathrm{FsC} &= [s_1, s_2, \ldots, s_{q^*}, \ldots, s_{Q_m}]\cdot[C_1, C_2, \ldots, C_{q^*}, \ldots, C_{Q_m}] \\ &= s_{q^*}C_{q^*} + \sum_{q\neq q^*}^{Q_m} s_q C_q = s_{q^*}\mathrm{KMF} + \sum_{q\neq q^*}^{Q_m} s_q C_q. \end{aligned} \tag{8}$$

By utilizing Equation (8), SCS can be explained as quantifying the compactness of attribute values gathering around the high-frequency categories (the mode and subdominant categories with relatively high frequencies) by using $\mathbf{s}$ to penalize the contribution of lower-frequency categories to the SCS values. Interestingly, when the mode has a large frequency and the term $s_{q^*}\mathrm{KMF}$ dominates in Equation (8), the influence of the second term diminishes, making SCS closely approximate the KMF. Unlike the KMF, which considers only mode-related compactness or distances, SCS accounts for all attribute values. By applying well-scaled frequency weights, SCS also mitigates the impact of noise from categories with extremely low frequencies.

In order to establish the relationship with other validity functions [6] such as the CU [35], we rewrite $\mathrm{SCS}(X,\pi)$ in Equation (5) as follows:

$$\begin{aligned}
&N\cdot\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{q=1}^{Q_m}\frac{\left(\mathrm{N}_{qk}^{(m)}\right)^2}{\mathrm{N}_{\cdot k}\cdot\mathrm{N}_{q\cdot}^{(m)}}-N\cdot M\\
&=N\cdot\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{q=1}^{Q_m}\left(\frac{\mathrm{N}_{qk}^{(m)}}{\mathrm{N}_{\cdot k}}\cdot\frac{\mathrm{N}_{qk}^{(m)}}{\mathrm{N}_{q\cdot}^{(m)}}\right)-N\cdot M\\
&=N\cdot\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{q=1}^{Q_m}\left(p(a_q^{(m)}|c_k)\cdot\frac{\mathrm{N}_{qk}^{(m)}}{\mathrm{N}_{q\cdot}^{(m)}}\right)-N\cdot M\\
&=N\cdot\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{q=1}^{Q_m}w_{mq}\cdot\left(\mathrm{N}_{\cdot k}\cdot p^2(a_q^{(m)}|c_k)\right)-N\cdot M,
\end{aligned}\tag{9}$$

where $p(a_q^{(m)}|c_k)=\frac{\mathrm{N}_{qk}^{(m)}}{\mathrm{N}_{\cdot k}}$ denotes the category-conditional probability that the mth attribute takes on the qth categorical value when it belongs to the kth cluster, and $w_{mq}=\frac{1}{\mathrm{N}_{q\cdot}^{(m)}}$ is the weight function depending on the categorical value distribution for each attribute.

According to the reference [6], CU can be written as:

$$\mathrm{CU}(X,\pi)=\frac{1}{N}\sum_{k=1}^{K}\mathrm{N}_{\cdot k}\sum_{m=1}^{M}\sum_{q=1}^{Q_m}p^2(a_q^{(m)}|c_k)-P=\frac{1}{N}\cdot\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{q=1}^{Q_m}\left(\mathrm{N}_{\cdot k}\cdot p^2(a_q^{(m)}|c_k)\right)-P,\tag{10}$$

where $P=\frac{1}{N}\cdot\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{q=1}^{Q_m}p^2(a_q^{(m)})$ and $p(a_q^{(m)})=\frac{\mathrm{N}_{q\cdot}^{(m)}}{N}$.

Given the dataset X, N, M, and P are constants. Therefore, to obtain the optimal partition π^* for a given X, maximizing CU is equivalent to maximizing the first term of Equation (10) and maximizing our SCS is equivalent to maximizing the first term of Equation (9). More precisely, CU is a special case of SCS when all w_{mq} $(1\le q\le Q_m)$ are equal for any mth attribute. As a generalized CU, for each cluster, our SCS objective function imposes a larger weight on higher-frequency attribute value by using $w_{mq}=\frac{\mathrm{N}_{qk}^{(m)}}{\mathrm{N}_{q\cdot}^{(m)}}$, which can facilitate the generation of more homogeneous clusters.

To calculate the objective function according to Equation (5), we need to know the number of objects, the number of attributes and the number of the qth attribute value in the kth cluster for each attribute. Obviously, these values are easy to obtain and the objective function can be calculated quickly in an incremental manner.

3.3 The Clustering Algorithm

3.3.1 An Overview. We develop a new clustering algorithm in which SCS is utilized as the objective function. As shown in Algorithm 1, the new clustering algorithm is named K-MHTC (Clustering Categorical Data via *MHT* with K clusters), whose input is a categorical dataset and a user-specified number of clusters K. The algorithm returns a locally optimal partition with respect to the SCS objective function.

Initially, we generate a random partition that assigns each object to 1 of K clusters independently and uniformly. Then, we try to update the initial partition in an iterative manner (Lines 3~15). In each iteration (Lines 5~14), we visit every object sequentially to check if its re-assignment to other

Algorithm 1: K-MHTC

Input: A categorical data set X, and the number of clusters K.
Output: A set of K locally optimal clusters $\hat{\pi}$.

Initialization: Generate a random partition of K clusters $\hat{\pi}$.
set $ind = 1$, $\mathcal{I} = 0$
while $ind == 1$ **do**
  $\mathcal{I}++$, $ind = 0$
  for $i = 1$ to N **do**
    for $b = 1$ to K **do**
      $\pi_{(b)} \leftarrow \hat{\pi}$, $\pi^i_{(b)} \leftarrow c_b$
    end for
    $x = \arg\max_{1 \le b \le K} \mathrm{SCS}(X, \pi_{(b)})$ according to Equation 4
    if $\hat{\pi}^i \neq c_x$ **then**
      $\hat{\pi} \leftarrow \pi_{(x)}$
      $ind = 1$
    end if
  end for
end while
return $\hat{\pi}$

$K-1$ clusters can improve the objective function. If the movement of the target object to a different cluster can increase the SCS value, then target object will be assigned to the cluster that will lead to the largest SCS value. This procedure will be terminated until there are no updates in one iteration, which means that moving any object to another cluster will not increase the SCS value.

3.3.2 Updating the SCS Value Incrementally. In Algorithm 1, the most time-consuming operation is to calculate the SCS value for each new partition after re-assigning the cluster membership of the ith object. Given the old SCS value of $\hat{\pi}$, we can quickly calculate the new SCS value of π as follows.

Suppose that o_i is moved from cluster c_A to c_B, then the difference $\Delta(o_i, A, B)$ between the two SCS values is only associated with two terms in $[T_1, \cdots, T_K]$, i.e., T_A and T_B. More precisely, we only need to update those terms in T_A and T_B that are involved with the attribute values appeared in o_i (i.e., x^i_m for $1 \le m \le M$). Suppose x^i_m is the $r^{(m)}$th attribute value for the mth attribute. According to Equation (5), $\hat{T}_A$, $\hat{T}_B$ in $\mathrm{SCS}(X, \hat{\pi})$ and T_A, T_B in $\mathrm{SCS}(X, \pi)$ can be expressed as follows:

$$\hat{T}_A = \frac{1}{\mathrm{N}_{\cdot A}} \cdot \sum_{m=1}^{M} \left(\sum_{q \neq r^{(m)}} \frac{\left(\mathrm{N}^{(m)}_{qA}\right)^2}{\mathrm{N}^{(m)}_{q\cdot}} + \frac{\left(\mathrm{N}^{(m)}_{r^{(m)}A}\right)^2}{\mathrm{N}^{(m)}_{r^{(m)}\cdot}} \right),$$

$$\hat{T}_B = \frac{1}{\mathrm{N}_{\cdot B}} \cdot \sum_{m=1}^{M} \left(\sum_{q \neq r^{(m)}} \frac{\left(\mathrm{N}^{(m)}_{qB}\right)^2}{\mathrm{N}^{(m)}_{q\cdot}} + \frac{\left(\mathrm{N}^{(m)}_{r^{(m)}B}\right)^2}{\mathrm{N}^{(m)}_{r^{(m)}\cdot}} \right),$$

$$T_A = \frac{1}{\mathrm{N}_{\cdot A} - 1} \cdot \sum_{m=1}^{M} \left(\sum_{q \neq r^{(m)}} \frac{\left(\mathrm{N}^{(m)}_{qA}\right)^2}{\mathrm{N}^{(m)}_{q\cdot}} + \frac{\left(\mathrm{N}^{(m)}_{r^{(m)}A} - 1\right)^2}{\mathrm{N}^{(m)}_{r^{(m)}\cdot}} \right),$$

$$T_B = \frac{1}{\mathrm{N}_{\cdot B} + 1} \cdot \sum_{m=1}^{M} \left(\sum_{q \neq r^{(m)}} \frac{\left(\mathrm{N}^{(m)}_{qB}\right)^2}{\mathrm{N}^{(m)}_{q\cdot}} + \frac{\left(\mathrm{N}^{(m)}_{r^{(m)}B} + 1\right)^2}{\mathrm{N}^{(m)}_{r^{(m)}\cdot}} \right).$$

Then, we can obtain the following two equations:

$$\begin{aligned}
\delta(A) &= T_A - \frac{\mathrm{N}_{\cdot A}}{\mathrm{N}_{\cdot A}-1}\hat{T}_A \\
&= \frac{1}{\mathrm{N}_{\cdot A}-1}\cdot\sum_{m=1}^{M}\frac{\left(\mathrm{N}^{(m)}_{r^{(m)}A}-1\right)^2-\left(\mathrm{N}^{(m)}_{r^{(m)}A}\right)^2}{\mathrm{N}^{(m)}_{r^{(m)}\cdot}} \\
&= \frac{1}{\mathrm{N}_{\cdot A}-1}\cdot\sum_{m=1}^{M}\frac{-2\cdot\mathrm{N}^{(m)}_{r^{(m)}A}+1}{\mathrm{N}^{(m)}_{r^{(m)}\cdot}}, \\
\delta(B) &= T_B - \frac{\mathrm{N}_{\cdot B}}{\mathrm{N}_{\cdot B}+1}\hat{T}_B \\
&= \frac{1}{\mathrm{N}_{\cdot B}+1}\cdot\sum_{m=1}^{M}\frac{\left(\mathrm{N}^{(m)}_{r^{(m)}B}+1\right)^2-\left(\mathrm{N}^{(m)}_{r^{(m)}B}\right)^2}{\mathrm{N}^{(m)}_{r^{(m)}\cdot}} \\
&= \frac{1}{\mathrm{N}_{\cdot B}+1}\cdot\sum_{m=1}^{M}\frac{2\cdot\mathrm{N}^{(m)}_{r^{(m)}B}-1}{\mathrm{N}^{(m)}_{r^{(m)}\cdot}}.
\end{aligned} \tag{11}$$

Thus, to obtain the new SCS value, we only need to update T_A and T_B based on $\hat{T}_A$ and $\hat{T}_B$ as follows:

$$\begin{aligned}
T_A &= \frac{\mathrm{N}_{\cdot A}}{\mathrm{N}_{\cdot A}-1}\hat{T}_A+\delta(A), \\
T_B &= \frac{\mathrm{N}_{\cdot B}}{\mathrm{N}_{\cdot B}+1}\hat{T}_B+\delta(B).
\end{aligned} \tag{12}$$

3.3.3 Complexity and Convergence Analysis. As shown in Equation (11), the calculation of $\delta(A)$ and $\delta(B)$ in Equation (12) requires the values of $\mathrm{N}^{(m)}_{r^{(m)}A}$ and $\mathrm{N}^{(m)}_{r^{(m)}B}$. We can retrieve and update the count of x^i_m in each cluster in $O(1)$ when a hash table is utilized to store attribute values and their counts. Thus, we can update each term in SCS in $O(M)$. In Lines 6∼9, we only need to update the T_A once and calculate the T_B value $K-1$ times in $O(KM)$. Therefore, the time complexity for re-assigning an object is $O(KM)$.

In the initialization step of Algorithm 1, we randomly generate N cluster labels in $O(N)$ and scan the dataset once in $O(NM)$ to obtain each $\mathrm{N}^{(m)}_{qk}$ for calculating the $\hat{T}$. In the iteration step (Lines 3∼15), we scan the given dataset $\mathcal{I}$ times and execute the re-assignment procedure (Lines 6∼13) N times in each iteration. Hence, the iteration step of Algorithm 1 has a time complexity $O(\mathcal{I}NKM)$. The overall time complexity of Algorithm 1 is $O(N)+O(NM)+O(\mathcal{I}NKM)=O(\mathcal{I}NKM)$. The scalability evaluation of the algorithm on large-scale simulated data is presented in Section 4.2

The K-MHTC algorithm converges after $\mathcal{I}$ iterations and we prove that $\mathcal{I}$ is a finite number as follows: First, the re-assignment procedure can only generate a finite number of possible partitions. Second, each possible partition cannot appear more than once in the iteration (Line 11) since the objective function in the algorithm is monotonically increasing. Therefore, the K-MHTC algorithm converges in a finite number of iterations. We will show in Section 4.2 that the number of iterations remains small across various datasets.

Table 4. The Properties of 25 Datasets

Dataset	*Abbr.*	N	M	Q	K
Lenses	Ls	24	4	9	3
Lung Cancer	Lc	32	56	159	3
Soybean (Small)	So	47	21	58	4
Photo Evaluation	Pe	66	4	21	3
Assistant Evaluation	Ae	72	4	21	3
Zoo	Zo	101	16	36	7
Promoter Sequences	Ps	106	57	228	2
Hayes-Roth	Hr	132	4	15	3
Lymphography	Ly	148	18	59	4
Heart Disease	Hd	303	13	57	5
Solar Flare	Sf	323	9	25	6
Primary Tumor	Pt	339	17	42	21
Dermatology	De	366	33	129	6
House Votes	Hv	435	16	48	2
Balance Scale	Bs	625	4	20	3
Credit Approval	Ca	690	9	45	2
Breast Cancer	Bc	699	9	90	2
Mammographic Mass	Mm	824	4	18	2
Tic-Tac-Toe	Tt	958	9	27	2
Lecturer Evaluation	Le	1,000	4	20	5
Car Evaluation	Ce	1,728	6	21	4
Titanic	Ti	2,201	3	6	4
Chess (kr vs. kp)	Ch	3,196	36	73	2
Mushroom	Mu	8,124	20	111	2
Nursery	Nu	12,960	8	27	5

4 Experiments

First, we compare K-MHTC[1] with 11 categorical data clustering methods on 25 real-world datasets in terms of three evaluation metrics. Then, we show that the combined p-value via MHT is effective on assessing whether a partition is statistically significant or not. All experiments are conducted on an Intel i7-12700K@3.60 GHz personal computer with 32G RAM.

The properties of 25 real-world datasets are shown in Table 4 where the notations used are consistent with the previous section and Q denotes the total number of all categorical values in a dataset. Among theses datasets, Photo Evaluation, Assistant Evaluation, and Lecturer Evaluation are collected from [89], others (except for Titanic[2]) are downloaded from the UCI Machine Learning Repository [32].

The baseline methods can be divided into three types as listed below:

—Classic partitioning methods based on objective functions [6] defined on the original categorical data: CU [35], K-modes [51], and Entropy [62]. We use the same procedure in our algorithm to maximize the CU objective function and a Monte Carlo procedure to minimize the Entropy objective function.

—SOTA partitioning methods: CDC_DR [7], CMS [58], CDE [59], Het2Hom [91], HD-NDW [89], and COForest [92]. In CDC_DR, spectral embedding and joint operation are used since

[1]https://github.com/hulianyu/MHTC.

[2]https://perso.liris.cnrs.fr/marc.plantevit/ENS/TP/dataTP2/.

this combination can achieve better performance than other candidates [7]. As suggested in the original paper, the parameter adjusting intra- and inter-attribute couplings in CMS is set to 0.5 for each dataset, and spectral clustering [75] is employed. Since there are no ordinal attributes in the datasets used in the experiment, we set the number of ordinal attributes to be 0 on each dataset in Het2Hom and HD-NDW. According to [89], HD-NDW is competitive to other SOTA algorithms such as UNTIE [95], WOC [56], and SBC [73] under this parameter setting. Therefore, those methods can achieve the same level performance as HD-NDW are not included in the performance comparison.

— Significance-based methods: DV [88] and SigDT [47]. These algorithms automatically determine the number of clusters. The former extracts clusters one by one, while the latter constructs an unsupervised decision tree. In some cases, DV may not output a result if it fails to identify any statistically significant clusters. For SigDT, even when the initial split of a given dataset is not statistically significant, it still enforces a cut into two clusters based on its splitting criterion. These significance-based methods can assist in determining whether a dataset is clusterable, which can be verified by DV if it detects at least one statistically significant individual cluster or by SigDT if the initial split of the data is statistically significant.

The source codes of above baseline methods are publicly available, and the codes of representation methods are provided by the original authors. In the performance comparison, we independently run each algorithm 50 times on each dataset and report the average results. We set the number of cluster K used in each algorithm (except for DV) to be the GT cluster number on each dataset.

To verify the clustering results, we employ three widely used external evaluation metrics: Clustering ACC [19], **normalized mutual information (NMI)** [82], and **adjusted Rand index (ARI)** [52]. These external metrics calculate the similarity between a clustering result π and the GT partition π_* on a dataset. The higher these metrics are, the better the performance of a clustering algorithm. All these metrics provide complementary insights into clustering performance from different perspectives, based on each object's cluster label π^i and GT label π_*^i. Specifically, they assess label matching (ACC), mutual information (NMI), and pair-counting (ARI). The detailed calculations are as follows:

$$\mathrm{ACC} = \frac{\sum_{i=1}^{N} f(\pi^i \rightarrow \pi_*^i)}{N}, \tag{13}$$

where f is a mapping function that aligns π^i with its corresponding GT label π_*^i, assigning $f(\pi^i \rightarrow \pi_*^i) = 1$ if they match and 0 otherwise. The optimal mapping is obtained using the Kuhn-Munkres algorithm [19].

$$\mathrm{NMI} = \frac{1}{2} \cdot \frac{\mathrm{H}(\pi) + \mathrm{H}(\pi_*) - \mathrm{H}(\pi, \pi_*)}{\mathrm{H}(\pi) + \mathrm{H}(\pi_*)}, \tag{14}$$

where both label sets are treated as random variables. Here, $\mathrm{H}(\pi)$ and $\mathrm{H}(\pi_*)$ denote their respective entropies, while $\mathrm{H}(\pi, \pi_*)$ represents their joint entropy.

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathbb{E}[\mathrm{RI}]}, \tag{15}$$

where the **Rand index (RI)** quantifies the proportion of correctly assigned object pairs. Here, $\mathbb{E}[\mathrm{RI}]$ is the expected RI, and $\max(\mathrm{RI})$ is its maximum value, both computed using a permutation model. The ARI ranges from -1 to 1, which can be regarded as a normalized RI. The RI is defined as

follows:

$$\mathrm{RI} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}. \tag{16}$$

In Equation (16), TP (true positive) denotes the number of object pairs that share the same π_* and are correctly clustered in π. FP (false positive) refers to pairs with different π_* that are incorrectly clustered together in π. FN (false negative) represents pairs with the same π_* that are incorrectly assigned to different clusters in π. TN (true negative) counts pairs with different π_* that are correctly assigned to separate clusters in π.

4.1 Performance Comparison

The performance comparison results in terms of three external metrics are listed in Table 5, where the mean and average rank assess the overall performance of each method. The faced values indicate the best results, and the corresponding average values are highlighted. The running time of each method on each dataset is displayed in Table 6. The total running time of DV in 50 executions is about 300 hours, which is much more time consuming than other methods. From Tables 5 and 6, we have the following key observations.

— *Overall Performance*: K-MHTC achieves the best overall performance in terms of all three metrics and ranks fourth in overall running time, with a notable second-fastest ranking on Ps (which contains the largest number of categorical values among all datasets). This demonstrates that the MHT-based objective function and our optimization procedure effectively identify meaningful categorical clusters. Compared to the best-performing competitor among the eleven methods, our algorithm achieves an overall improvement of more than 3.5% in NMI and 4.5% in ARI. Meanwhile, K-MHTC is highly reliable, as it never produces partitions that are worse than random ones. In contrast, as shown in Table 5, competing algorithms (except Entropy, CDE, HD-NDW, and COForest) commonly generate partitions that are worse than arbitrary partitions on certain datasets, such as Hr.

— *Comparison with DV and SigDT:* Since K-MHTC is a clustering method based on significance testing, we include DV and SigDT in the performance comparison despite the fact that they do not belong to k-partition methods. Regarding DV, our method runs significantly faster across all datasets, whereas DV often fails to report clustering results on datasets that may contain meaningful clusters, at least those clusterable datasets recognized by SigDT. As shown in Figure 2(a), on DV-verified datasets (where DV can report a result), K-MHTC significantly outperforms DV across all three metrics. As indicated in Table 6, SigDT ranks as the fastest competing method, suggesting that significance testing can potentially be applied efficiently in clustering algorithms. However, its clustering quality is considerably worse than that of K-MHTC, with ARI being particularly inferior, as shown in Figure 2(a). Even on SigDT-verified datasets (where SigDT can produce at least a statistically significant initial split), it still fails to outperform K-MHTC, as evidenced in Figure 2(b).

— *Comparison with Classic Partitioning Algorithms*: In terms of three metrics, K-MHTC can achieve better performance than the best-performing methods based on validity functions on 13 datasets (except for Ls, Lc, Pe, Pt, Ca, Mm, Tt, Ce, Ti, Ch, Mu, Nu), while one method among CU, Entropy and K-modes can achieve better results than K-MHTC on nine datasets. When the same 50 initial random partitions are utilized in both K-MHTC and CU, CU usually requires more iterations (as evidenced in Section 4.2). This could partially explain why CU method takes more running time than K-MHTC on most datasets, even the CU function is a special case of our SCS function. The performance contrast is more pronounced on datasets with verified clustering tendency. As shown in Figure 2(a), K-MHTC is statistically significantly

Table 5. The Performance of K-MHTC, CU, K-Modes, Entropy, DV, SigDT, CDC_DR, CMS, CDE, Het2Hom, HD-NDW, and COForest Is Compared in Terms of ACC, NMI, and ARI across 25 Datasets

Metric	Method	Ls	Lc	So	Pe	Ae	Zo	Ps	Hr	Ly	Hd	Sf	Pt	De	Hv	Bs	Ca	Bc	Mm	Tt	Le	Ce	Ti	Ch	Mu	Nu	Mean	Rank
	Random	0.475	0.443	0.360	0.421	0.407	0.260	0.540	0.391	0.311	0.251	0.223	0.148	0.220	0.519	0.358	0.516	0.517	0.515	0.513	0.229	0.266	0.269	0.507	0.514	0.212	0.375	11.64
	K-MHTC	0.507	0.531	0.984	0.544	0.549	0.809	0.753	0.430	0.592	0.437	**0.524**	0.286	0.745	**0.880**	0.459	0.620	**0.974**	0.790	0.556	0.322	0.314	0.420	0.524	0.814	0.344	**0.588**	**4.88**
	CU	0.564	0.513	0.892	0.531	0.532	0.599	0.709	**0.353**	0.468	0.352	0.463	0.277	0.740	0.869	0.417	0.772	0.914	**0.827**	0.560	0.297	0.351	0.330	0.544	0.815	0.311	0.560	6.96
	K-modes	0.543	0.517	0.826	**0.564**	0.509	0.691	0.557	**0.370**	0.441	0.393	0.473	0.295	0.617	0.866	0.424	0.749	0.904	0.818	0.548	0.319	0.380	0.433	0.544	0.733	0.353	0.555	6.80
	Entropy	0.543	**0.596**	0.871	0.528	0.536	0.705	0.723	0.422	0.522	0.380	0.444	0.295	0.622	0.869	0.442	0.659	0.962	0.749	0.567	0.313	0.383	0.399	0.521	0.757	0.331	0.566	6.30
	DV	–	–	**1**	–	–	**0.950**	–	–	0.345	**0.168**	–	–	0.497	0.756	–	–	0.738	–	–	–	–	–	–	**0.245**	–	–	–
ACC	SigDT	0.542	0.563	**1**	0.470	0.472	0.723	0.660	**0.485**	0.588	**0.479**	0.495	0.301	**0.833**	0.582	**0.590**	0.683	0.763	0.824	**0.401**	**0.401**	**0.534**	**0.438**	**0.164**	0.530	**0.402**	0.557	5.14
	CDC_DR	**0.417**	0.534	0.859	0.525	0.535	0.757	0.763	**0.374**	0.604	0.416	0.490	0.302	0.689	0.846	0.517	0.681	0.973	0.817	0.557	0.309	0.335	0.428	0.533	0.533	0.304	0.564	6.04
	CMS	**0.574**	0.576	**1**	0.448	0.484	0.641	**0.765**	**0.341**	0.503	0.324	0.437	0.262	0.808	0.878	0.381	0.678	0.964	0.826	0.517	0.305	0.399	0.416	0.509	**0.910**	0.310	0.570	6.78
	CDE	0.508	0.526	0.893	0.500	0.506	0.746	0.699	0.477	0.517	0.400	0.489	**0.305**	0.685	0.863	0.475	0.650	0.967	0.792	**0.573**	0.316	0.364	0.426	0.527	0.873	0.320	0.576	6.08
	Het2Hom	0.535	0.538	0.988	0.524	0.522	0.719	**0.511**	**0.333**	0.489	0.350	0.442	0.285	0.754	0.874	0.468	**0.836**	0.967	0.812	0.555	0.333	0.376	0.428	0.521	0.883	0.311	0.574	6.24
	HD-NDW	0.552	0.535	0.827	0.541	**0.555**	0.741	0.678	0.408	**0.668**	0.453	0.521	0.300	0.715	0.867	0.431	0.542	0.969	0.794	**0.573**	0.319	0.369	0.435	0.512	0.811	0.326	0.578	5.16
	COForest	0.543	0.539	0.894	0.516	0.524	0.700	0.638	0.415	0.506	0.362	0.410	0.272	0.735	0.877	0.448	0.712	0.920	0.806	0.566	0.323	0.384	0.428	**0.558**	0.783	0.319	0.567	5.98
	Random	0.130	0.069	0.074	0.035	0.026	0.109	0.008	0.015	0.024	0.016	0.024	0.170	0.019	0.002	0.003	0.001	0.001	0.001	0.001	0.005	0.002	0.002	0.000	0.000	0.000	0.029	11.64
	K-MHTC	0.179	0.274	0.986	0.202	0.201	0.813	0.264	0.059	**0.256**	0.189	0.389	0.325	**0.826**	0.483	0.046	0.084	**0.820**	0.294	0.008	0.067	0.051	**0.109**	0.003	0.380	0.051	**0.294**	**4.32**
	CU	0.279	0.163	0.838	0.165	0.158	0.673	0.205	**0.003**	0.171	0.173	0.297	0.344	0.748	0.482	0.026	0.276	0.625	0.339	0.019	0.049	0.088	0.068	0.008	0.283	0.090	0.263	6.44
	K-modes	0.241	0.201	0.822	0.191	0.130	0.736	0.019	**0.007**	0.110	0.153	0.288	0.337	0.570	0.462	0.014	0.207	0.568	0.325	0.010	0.049	0.073	0.067	0.010	0.205	0.063	0.234	7.80
	Entropy	0.253	**0.310**	0.874	0.177	0.177	0.760	0.232	0.049	0.206	0.177	0.272	0.361	0.766	0.470	0.032	0.136	0.774	0.234	0.011	0.052	0.060	0.083	0.003	0.271	0.088	0.273	6.08
	DV	–	–	**1**	–	–	**0.916**	–	–	0.161	0.153	–	–	0.275	0.331	–	–	0.371	–	–	–	–	–	–	0.226	–	–	–
NMI	SigDT	**0.020**	0.189	**1**	0.047	0.049	0.699	0.193	0.037	0.191	0.111	0.316	0.175	0.811	0.327	0.068	0.293	0.353	**0.348**	**0.048**	0.040	0.034	0.026	**0.019**	0.274	0.025	0.228	7.18
	CDC_DR	**0.025**	0.247	0.886	0.187	0.165	0.784	0.238	**0.012**	0.228	0.171	0.333	0.350	0.792	0.401	**0.072**	0.167	0.813	0.325	0.005	0.062	0.036	0.094	0.004	0.007	0.086	0.260	5.72
	CMS	**0.282**	0.225	**1**	0.134	0.149	0.640	**0.297**	**0.000**	0.181	0.128	0.230	0.309	0.804	**0.490**	0.014	0.132	0.778	0.338	**0.000**	0.040	**0.118**	0.104	**0.000**	**0.563**	0.060	0.281	6.54
	CDE	0.184	0.210	0.905	0.161	0.145	0.778	0.158	**0.103**	0.191	**0.190**	0.331	**0.364**	0.769	0.433	0.058	0.130	0.786	0.299	0.013	0.052	0.066	0.073	0.005	0.505	0.076	0.279	5.60
	Het2Hom	0.234	0.219	0.989	0.183	0.197	0.798	0.012	**0.000**	0.157	0.172	0.248	0.345	0.771	0.480	0.003	**0.369**	0.777	0.321	0.012	**0.073**	0.082	0.104	0.001	0.452	0.055	0.282	6.04
	HD-NDW	0.265	0.228	0.843	**0.231**	**0.258**	0.777	0.147	0.028	0.254	0.179	**0.408**	0.364	0.779	0.461	0.034	0.021	0.785	0.302	0.007	0.055	0.095	0.057	0.004	0.418	**0.113**	0.284	5.04
	COForest	0.240	0.234	0.908	0.187	0.179	0.770	0.103	0.035	0.135	0.138	0.224	0.322	0.805	0.482	0.028	0.212	0.669	0.317	0.011	0.063	0.089	0.097	0.017	0.300	0.093	0.266	5.60
	Random	0.029	−0.003	−0.008	0.006	−0.001	0.002	0.001	0.001	0.000	−0.001	0.000	0.000	−0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	11.60
	K-MHTC	0.092	0.174	0.977	0.143	0.153	0.768	**0.291**	0.037	**0.246**	0.203	**0.315**	0.100	0.711	**0.578**	0.048	0.093	**0.899**	0.366	0.013	0.044	0.014	**0.107**	0.005	0.421	0.041	**0.274**	**4.16**
	CU	0.144	0.079	0.767	0.119	0.120	0.495	0.245	**−0.012**	0.164	0.125	0.232	0.107	0.670	0.545	0.032	0.340	0.686	**0.428**	0.024	0.031	0.052	0.058	0.010	0.366	0.074	0.236	6.12
	K-modes	0.123	0.124	0.736	0.150	0.093	0.640	0.013	**−0.010**	0.077	0.141	0.212	0.091	0.479	0.534	0.015	0.257	0.676	0.403	0.016	0.025	0.047	0.089	0.012	0.229	0.058	0.209	7.48
	Entropy	0.120	**0.203**	0.822	0.120	0.127	0.636	0.237	0.021	0.203	0.158	0.199	0.120	0.594	0.544	0.031	0.151	0.853	0.271	0.012	0.024	0.035	0.075	0.003	0.261	0.070	0.236	6.64
	DV	–	–	**1**	–	–	**0.953**	–	–	0.120	0.057	–	–	0.194	0.449	–	–	0.653	–	–	–	–	–	–	0.140	–	–	–
ARI	SigDT	0.030	0.147	**1**	**−0.002**	0.008	0.592	0.271	0.032	0.161	0.166	0.206	0.083	**0.818**	0.365	0.064	0.347	0.509	0.419	**0.092**	0.021	**−0.068**	0.014	0.010	0.315	0.031	0.225	6.94
	CDC_DR	**−0.043**	0.162	0.810	0.112	0.108	0.720	0.281	**−0.003**	0.232	0.157	0.243	0.115	0.646	0.499	**0.086**	0.201	0.894	0.401	0.012	0.034	0.015	0.094	0.004	0.003	0.074	0.234	6.00
	CMS	**0.167**	0.160	**1**	0.075	0.093	0.494	0.277	**−0.015**	0.188	0.090	0.181	0.089	0.744	0.571	0.015	0.158	0.861	0.426	**−0.001**	0.028	**0.076**	0.102	**0.000**	**0.627**	0.052	0.258	6.26
	CDE	0.079	0.131	0.851	0.102	0.104	0.704	0.191	**0.096**	0.188	0.180	0.247	**0.127**	0.641	0.529	0.056	0.146	0.871	0.369	0.022	0.033	0.028	0.074	0.007	0.546	0.057	0.255	6.00
	Het2Hom	0.098	0.145	0.984	0.131	0.153	0.672	0.004	**−0.015**	0.167	0.126	0.176	0.118	0.711	0.557	0.001	**0.457**	0.871	0.396	0.015	**0.045**	0.047	0.100	0.001	0.557	0.034	0.262	5.92
	HD-NDW	0.140	0.136	0.775	**0.171**	**0.211**	0.700	0.145	0.017	0.243	**0.221**	0.312	0.124	0.662	0.537	0.038	0.003	0.877	0.371	0.018	0.035	0.056	0.082	0.004	0.416	**0.091**	0.255	4.72
	COForest	0.121	0.132	0.869	0.117	0.122	0.655	0.112	0.019	0.128	0.115	0.129	0.086	0.689	0.566	0.030	0.250	0.759	0.396	0.018	0.039	0.049	0.099	**0.020**	0.355	0.066	0.238	6.16

Each algorithm is run for 50 executions (except for DV and SigDT), and the average clustering results are reported. Metric values lower than those from the random partitions (using the same initialization as in K-MHTC and CU) are marked in blue. If DV fails to find any statistically significant clusters for a given dataset, its result is denoted as "–." For CMS and Het2Hom & HD-NDW, the parameters are fixed to be 0.5 and 0, respectively.

Table 6. The Running Time for 50 Executions of Each Algorithm across 25 Datasets, Along with the Total Running Time of Each Algorithm

Method	Ls	Lc	So	Pe	Ae	Zo	Ps	Hr	Ly	Hd	Sf	Pt	De	Hv	Bs	Ca	Bc	Mm	Tt	Le	Ce	Ti	Ch	Mu	Nu	Total
K-MHTC	<1	<1	<1	<1	<1	1.2	1.6	<1	1.7	4.3	2.5	21.5	9.5	1.2	<1	1.9	1.2	<1	2.5	3.9	3.9	3.9	34.3	51.7	43.9	193.8
CU	<1	<1	<1	<1	<1	1.6	2.5	<1	1.9	5.1	3.8	37.8	21.5	1.0	1.1	2.3	1.2	<1	4.2	3.3	5.7	5.1	94.1	79.3	84.9	358.5
K-modes	<1	1.5	<1	<1	<1	1.2	7.3	<1	2.6	3.9	2.6	7.2	14.0	7.0	3.4	9.1	10.2	5.3	17.3	6.5	18.6	12.4	424.1	2.E+03	651.6	3.E+03
Entropy	<1	2.5	2.4	<1	<1	10.6	8.0	1.5	13.3	37.2	23.9	248.8	105.7	11.2	9.6	8.1	12.4	6.2	12.5	35.9	60.4	44.6	196.6	339.3	993.2	2.E+03
DV	<1	87.4	37.6	5.1	5.1	47.7	2.E+03	2.2	557.1	3.E+03	16.4	359.6	4.E+03	417.0	16.0	210.2	3.E+03	2.9	543.8	6.3	118.2	<1	4.E+04	1.E+06	7.E+03	1.E+06
SigDT	<1	<1	<1	<1	<1	<1	1.9	<1	<1	<1	<1	<1	1.4	<1	<1	<1	<1	<1	<1	<1	<1	<1	13.5	10.4	0.8	**32.0**
CDC_DR	<1	<1	<1	<1	<1	<1	1.1	<1	<1	<1	<1	<1	<1	<1	<1	<1	<1	<1	<1	<1	<1	<1	7.9	25.5	49.5	90.2
CMS	<1	3.5	<1	<1	<1	<1	9.3	<1	<1	1.3	<1	1.1	5.0	1.6	1.5	2.8	4.3	2.7	4.4	3.9	14.9	19.2	95.1	501.9	884.5	2.E+03
CDE	<1	8.5	1.6	<1	<1	<1	19.7	<1	1.8	1.7	<1	1.4	7.7	2.0	<1	1.6	3.7	<1	1.4	<1	<1	<1	19.1	39.5	6.3	122.3
Het2Hom	<1	12.8	3.4	3.9	4.0	3.3	34.8	<1	12.6	71.4	7.6	93.2	109.9	15.9	20.9	24.6	32.2	7.0	8.6	17.5	16.3	1.8	122.7	832.9	380.8	2.E+03
HD-NDW	<1	2.6	<1	<1	<1	1.3	9.3	<1	3.4	11.8	3.9	34.7	23.0	2.2	2.2	3.1	13.3	1.6	6.4	9.5	13.1	1.7	63.3	138.2	148.3	494.6
COForest	<1	1.5	<1	<1	<1	1.2	3.6	<1	3.2	7.2	3.5	30.1	21.0	1.7	28.1	1.4	6.2	1.3	4.0	7.7	7.9	1.2	52.2	87.5	79.4	352.3

For DV and SigDT, executed only once, their running time is estimated by multiplying the single-execution time by 50. Running times of less than 1 second are denoted as "<1," while those exceeding 1,000 seconds are expressed in exponential form.

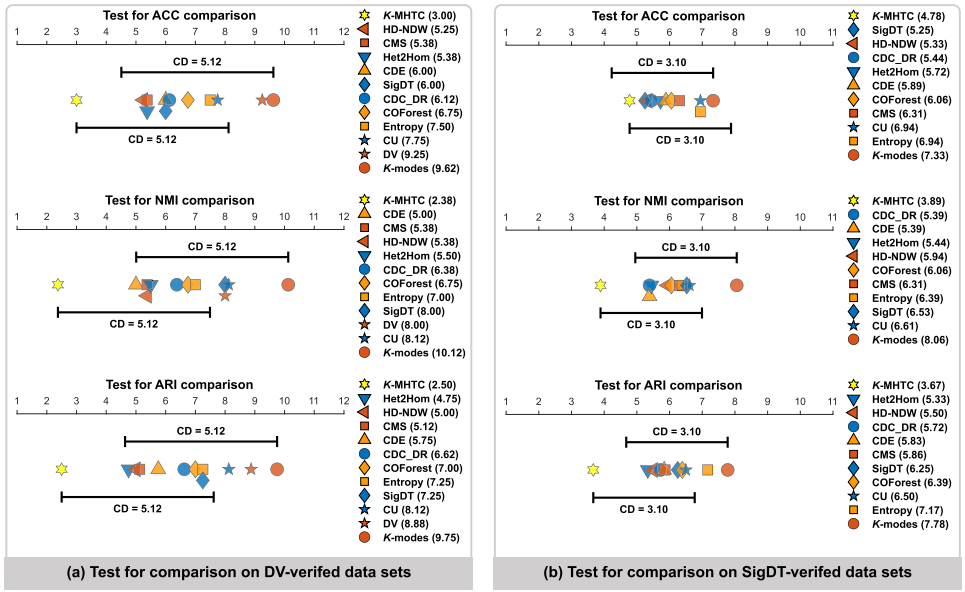


Fig. 2. Comparison of K-MHTC vs. other methods based on two-tailed Bonferroni-Dunn test [30] at the 95% CI. The Critical Difference (CD) is calculated by using the number of comparison methods, the corresponding critical value, and the number of datasets in which these methods are performed. In every subfigure, each method has a position in the diagram based on its average rank, where the CD interval can be located. Taking the best-performing K-MHTC and the worst-performing K-modes as reference points, any method ranking outside the marked CD intervals is believed to be statistically significantly worse than K-MHTC or not significantly different from K-modes, respectively. (a) Comparison with all methods on eight DV-verified datasets, where DV reports statistically significant individual clusters. (b) Comparison with all methods (except DV) on 18 SigDT-verified datasets, where SigDT reports statistically significant partitions.

superior to K-modes across all three metrics and outperforms CU in terms of NMI and ARI. On the 18 SigDT-verified datasets (Lc, So, Zo, Ps, Ly, Hd, Sf, Pt, De, Hv, Ca, Bc, Mm, Tt, Le, Ti, Ch, Mu), as shown in Figure 2(b), K-MHTC demonstrates statistically significant improvements over Entropy in ARI.

— *Comparison with SOTA Methods*: In terms of all three metrics, K-MHTC can beat best-performing methods CDE, Het2Hom, and HD-NDW on 16, 16, and 14 datasets, respectively. Although CDC_DR runs more efficiently than K-MHTC by employing a graph-based representation and using k-means, it can only achieve fully superior clustering results on three datasets (Pt, Bs, Ca). Moreover, it seems that CDC_DR fails to detect a meaningful clustering structure on Ls and Hr since the ARI values are negative. Notably, whether on DV-verified or SigDT-verified datasets, none of these SOTA methods exhibit statistically significant superiority over the worst-performing K-modes in the comparison, within the 95% CI shown in Figure 2. In addition, it is not an easy task to specify ideal parameters for CMS, Het2Hom, and HD-NDW. For example, in order to obtain the number of ordinal attributes, we need to acquire domain knowledge that goes beyond the information encoded in the datasets.

4.2 Convergence and Scalability Comparison

To evaluate the convergence of K-MHTC in finding locally optimal clusters, we examine the number of iterations ($\mathcal{I}$, as recorded in Algorithm 1) required for its objective function (SCS) to reach a local

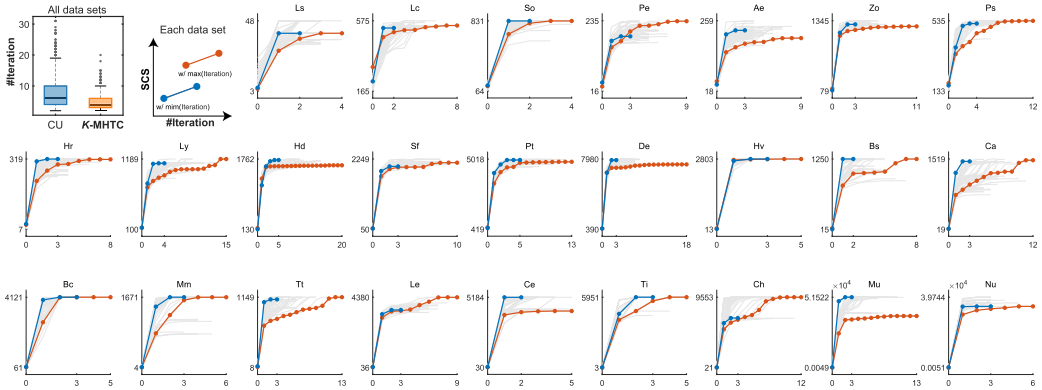


Fig. 3. Convergence curves of K-MHTC on 25 datasets. In each plot, all convergence curves across 50 executions are displayed, in which the curves with the minimum and maximum iteration counts are marked in blue and red, respectively. The boxplot in the upper-left corner illustrates the distribution of iteration counts across all 50×25 cases, compared to those of CU.

optimum, as shown in Figure 3. For each dataset, we run K-MHTC 50 times with different random partitions in the initialization, with each run yielding varying iteration counts and final SCS values. Despite the random initialization, the distribution of iteration counts across all 25 datasets shows that the median does not exceed 5, while the maximum iteration count reaches 20 in a single run on Hd (as indicated by an outlier in the boxplot in Figure 3). Overall, the number of iterations is considerably lower than that required by CU.

To evaluate the scalability of K-MHTC on large-scale datasets, we adjust the scale by randomly generating synthetic datasets with proportionally increasing numbers of objects (N) and attributes (M), varying one while keeping the other fixed. Following this strategy, the execution time curves are plotted in Figure 4, where we also include six most time-efficient competing methods according to the results in Table 6. From the execution time curves with varying N, we observe that K-MHTC performs similarly to COForest, with both ranking third among all compared methods. While increasing M quickly raises the computational cost for K-MHTC due to the growing number of Chi-square statistic calculations in the MHT-based objective function. Note that the running time curve for varying M exhibit a clear linear growth trend, aligning with the time complexity analysis in Section 3.3.3. Among the SOTA algorithms, CDE is particularly sensitive to the increase of M, suggesting it may not be well-suited for high-dimensional data.

4.3 Validity of SCS and the Combined p-Value

4.3.1 Validity of SCS. As a significance-based objective function, it is designed to obtain clustering results that are distinguished from random partitions. From the left subfigure in Figure 5, we observe that all 10 k-partition clustering algorithms yield results with SCS values greater than those of random partitions. However, such random partitions may be overly idealized, and further study is needed to examine how varying degrees of randomness affect SCS, from completely randomized partitions (as used in Section 4.1) to partially randomized partitions (in the upcoming simulation experiments).

To generate a partially randomized partition, we first treat the set of cluster labels as an ordered vector of length N. Next, we randomly select a fraction of N positions, specifically $y\%$ of N positions (i.e., $(y/100) \times N$ positions), and shuffle the labels at these positions. This process reassigns the cluster labels of the selected objects. For example, consider a dataset with $N = 500$ objects. If we set

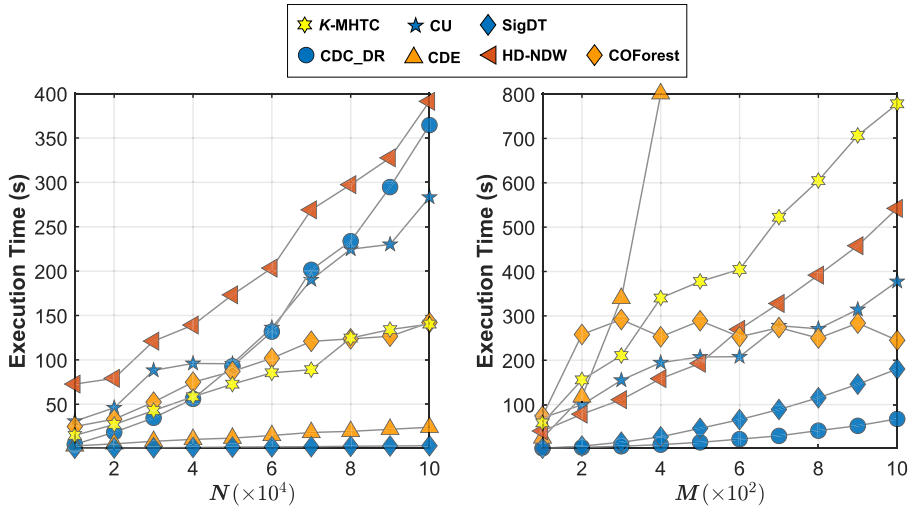


Fig. 4. Execution time curves (averaged over 10 runs) of each algorithm on synthetic datasets. Each dataset is generated along attributes, with each attribute having five categories randomly assigned to objects, and clustering algorithms are executed with fixed $K = 5$. Left subfigure: $N = 10{,}000 \longrightarrow 100{,}000$ (step 10,000), with fixed $M = 20$. Right subfigure: $M = 100 \longrightarrow 1{,}000$ (step 100), with fixed $N = 2{,}000$.

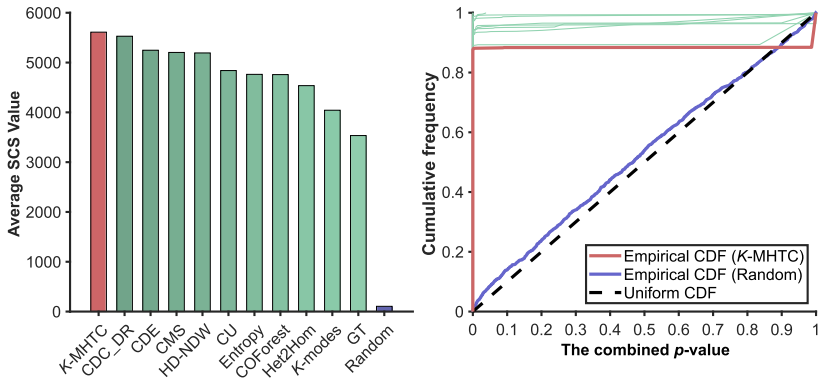


Fig. 5. Comparison of SCS and the combined p-value between algorithm-derived and random partitions. All partition results are consistent with those used in Table 5. The left subfigure presents the average SCS value per run on each dataset for different methods, with each method based on 25×50 partitions. The average SCS of GT partitions across 25 datasets is included as a reference. The right subfigure displays the empirical CDF of the combined p-value for each set of 25×50 algorithm-derived partitions, highlighting the curves of K-MHTC, random partitions, and the theoretical uniform CDF. CDF, cumulative distribution function.

$y = 1$, we randomly select $1\% \times 500 = 5$ positions. Using the MATLAB function `randperm(500,5)`, we might obtain the positions [323, 189, 405, 265, 174]. We then sort these selected positions to obtain [174, 189, 265, 323, 405] and reassign the cluster labels accordingly. As a result, only the cluster labels of these five objects are randomized. By varying y between 1 and 100, we can generate partitions that range from being very similar to the original cluster labels (low randomness) to completely randomized (high randomness). This approach allows us to control the degree of randomness introduced into the partitions.

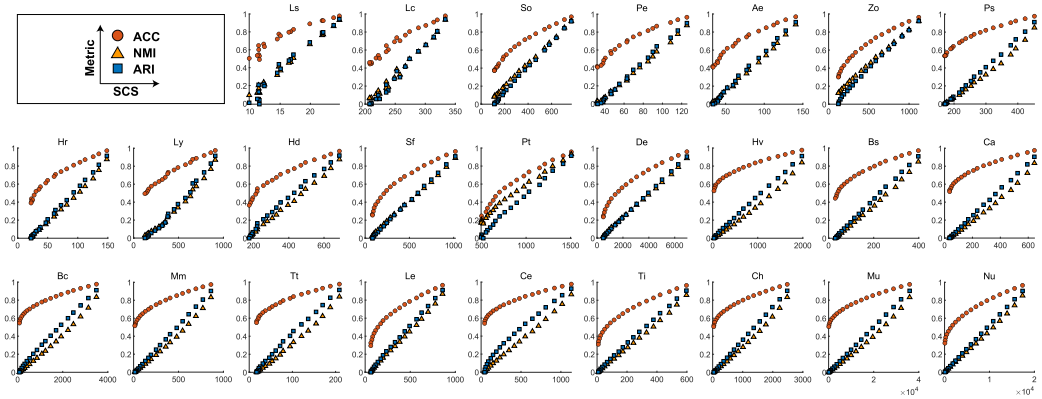


Fig. 6. Scatter plot of SCS vs. metrics with varied randomized GT partitions on each dataset.

We conduct simulation experiments on 25 datasets by perturbing cluster labels in their corresponding GT partitions. The parameter y is varied from 5 to 100 in 20 steps. To ensure stable numerical results at each step, we perform 50 randomized trials and compute the average ACC, NMI, ARI, and SCS. As shown in Figure 6, the increase of SCS values exhibits an approximately linear relationship with the increase of performance metrics. Notably, for each dataset, the values of NMI and ARI are distributed between 0 and 1, indicating that our simulation procedure is reasonable for validating SCS across partitions with varied clustering quality.

4.3.2 Validity of the Combined p-Value. To assess whether a partition is statistically significant or not, we can directly calculate the combined p-value by using the rOP method in which r is set to be $\lfloor 0.5 \cdot M \rfloor$ for the given dataset. In order to confirm the validity of the combined p-value, we first show that a random partition typically is associated with a quite large p-value, while the GT or algorithm-optimized partition generally has a small p-value, often close to 0. Then, we further demonstrate that the combined p-value is effective in assessing clustering quality, as measured by external metrics such as ACC, NMI, and ARI.

We collect the combined p-values of algorithm-optimized and random partitions across all 25 datasets and show their empirical **cumulative distribution function (CDF)** curves in the right subfigure of Figure 5. It is observed that the combined p-values of random partitions are approximately uniformly distributed, meaning that the probability of obtaining a value smaller than 0.01 is close to only 1%. In contrast, all CDF curves of algorithm-optimized partitions exhibit clear deviations from uniformity and a skew toward smaller p-values or zeros. Notably, SCS-optimized partitions (the empirical CDF curve of K-MHTC) also exhibit sensitivity to datasets lacking clustering tendency, particularly as multiple partitions yield the combined p-values equal to 1 on unclusterable datasets (Ls, Bs, Ce, and Nu). These four datasets are identified as unclusterable by TestCat [45], a well-designed clusterability test method for categorical data.

As shown in Figure 7, the combined p-values for random partitions are generally greater than the significance level of 0.01 across all datasets, indicating that such arbitrary partitions are not statistically significant. In contrast, clustering algorithms predominantly produce valid partitions, consistently generating partitions with a combined p-value ≤ 0.01 in nearly all runs on most datasets. However, on unclusterable datasets (Ls, Bs, Ce, and Nu), even algorithm-optimized partitions are regarded as being statistically invalid in approximately 5% to 20% of cases. This suggests that optimizing objective functions does not guarantee finding a true clustering structure, and

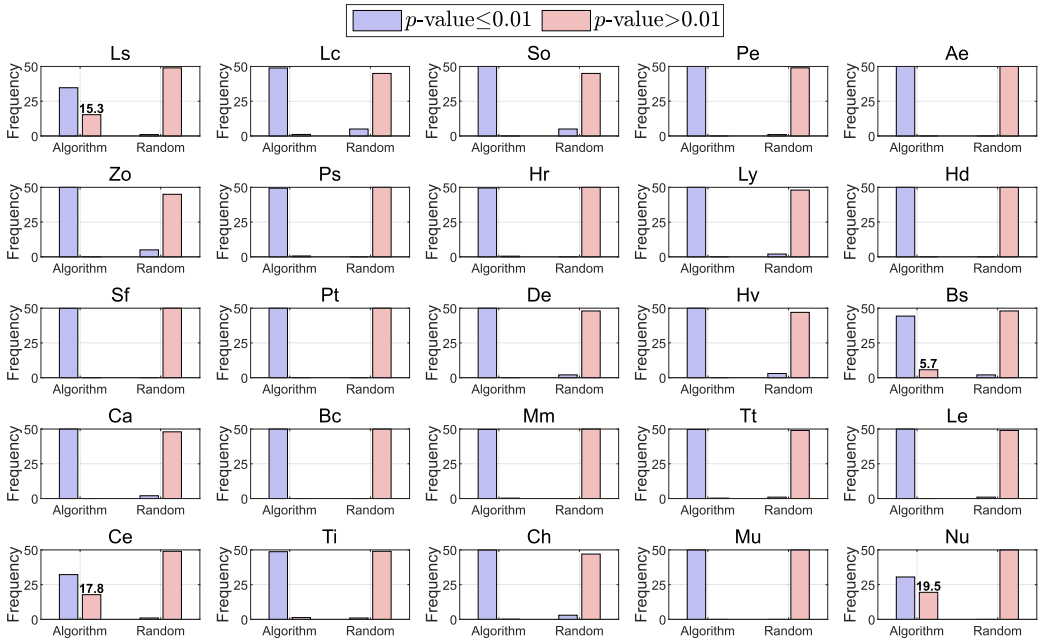


Fig. 7. The combined p-values in algorithm-optimized partitions vs. random partitions. The distribution of resulting p-values for each dataset is profiled based on its frequency over 50 runs, categorized by a significance level of 0.01 and separately displayed for the two types of partitions. For algorithm-optimized partitions, the reported frequency represents the average value across 10 k-partition clustering algorithms, excluding DV and SigDT.

any algorithm-derived partitions, especially on completely randomized datasets, require further validation.

All GT partitions on 25 datasets are statistical significant where the p-value < 0.01. For example, the p-values of the GT partitions on Ls, Lc, Pe, Ae, Ps and Tt are 6.83E−4, 2.03E−5, 4.46E−11, 2.66E−12, 7.38E−9, and 1.84E−14, respectively. In addition, the p-values of the GT partitions on So, Zo, Ly, Hd, De, Hv, Bc, and Mu are zeros on which the DV method can find statistically significant clusters.

If one dataset has a clear and strong clustering structure, it can be expected that the corresponding optimized partitions will yield statistically significant results, while most clustering algorithms can achieve relatively good performance. To aid both illustration and interpretation, we first group the datasets using our method as well as other significance-based methods. As shown in the last row of Table 7, we use the combined p-value as a validation indicator for data-related k-partition results. Based on whether all partitions are claimed to be statistically significant by our method, the dataset is classified as either X(sig) (marked with "✓" by "Ours" in Table 7) or X(unsig) as follows:

—X(sig): {So, Pe, Ae, Zo, Ly, Hd, Pt, De, Hv, Ca, Bc, Le, Mu}.
—X(unsig): {Ls, Lc, Ps, Hr, Sf, Bs, Mm, Tt, Ce, Ti, Ch, Nu}.

Similarly, datasets verified by DV and SigDT (marked with "✓" by DV and SigDT in Table 7) are grouped as follows:

—X(DV): {So, Zo, Ly, Hd, De, Hv, Bc, Mu}.
—X(SigDT): {Lc, So, Zo, Ps, Ly, Hd, Sf, Pt, De, Hv, Ca, Bc, Mm, Tt, Le, Ti, Ch, Mu}.

Table 7. Partition Quality and Validation on Each Dataset, Assessed Separately by Performance Metrics and Significance-Based Algorithms

		Ls	Lc	So	Pe	Ae	Zo	Ps	Hr	Ly	Hd	Sf	Pt	De	Hv	Bs	Ca	Bc	Mm	Tt	Le	Ce	Ti	Ch	Mu	Nu
Metric	ACC	0.528	0.541	0.903	0.522	0.525	0.711	0.680	0.392	0.531	0.387	0.469	0.288	0.711	0.869	0.446	0.690	0.951	0.803	0.557	0.315	0.365	0.414	0.529	0.791	0.323
	NMI	0.218	0.231	0.905	0.182	0.176	0.753	0.167	0.030	0.189	0.167	0.302	0.342	0.763	0.464	0.033	0.173	0.740	0.309	0.010	0.056	0.076	0.086	0.005	0.338	0.078
	ARI	0.104	0.145	0.859	0.124	0.128	0.648	0.179	0.014	0.183	0.152	0.225	0.108	0.655	0.546	0.035	0.206	0.825	0.383	0.015	0.034	0.042	0.088	0.007	0.378	0.062
	avgRank	13.53	**11.97**	**1.43**	13.97	13.70	**4.70**	**11.70**	21.97	**12.50**	15.77	**11.50**	15.83	**4.60**	**4.40**	20.33	**11.43**	**1.93**	**6.57**	19.03	21.57	20.17	18.13	20.63	**7.27**	20.37
Validation	DV-verified			✓			✓			✓	✓			✓	✓			✓							✓	
	SigDT-verified		✓	✓			✓	✓		✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓		✓	✓	✓	
	TestCat		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓		✓	✓	✓	
	Ours	69.4%	97.8%	✓	✓	✓	✓	98.8%	99.0%	✓	✓	99.8%	✓	✓	✓	88.6%	✓	✓	99.4%	99.4%	✓	64.4%	97.4%	99.8%	✓	61.0%

Note that we consider k-partition results from Table 5, excluding those produced by DV and SigDT. In the metric row, average ranks across three metrics are reported, with top-half ranks marked in bold. In the validation row, a "✓" indicates that a method determines the existence of clustering tendency in the dataset. The final row presents the validation results of our combined p-value (ours), where a "✓" is assigned only to a dataset in which all 50 partitions consistently (100%) yield small p-values (≤ 0.01).

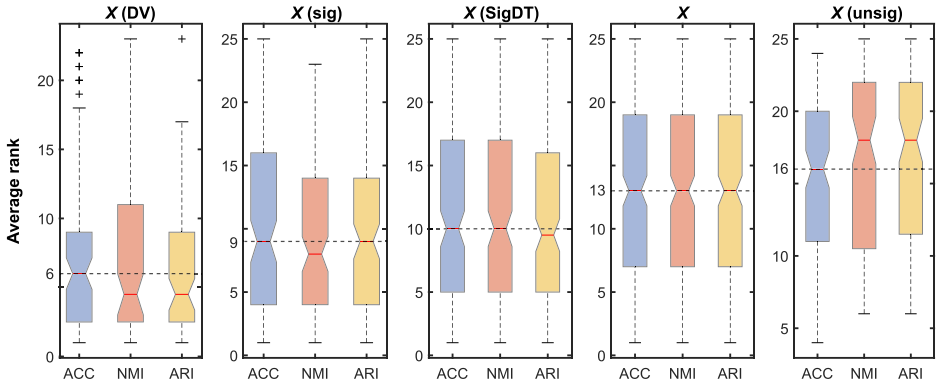


Fig. 8. The box-plots of the average ranks based on the results in Table 7.

Our method shares similar characteristics with DV and SigDT. We observe that DV is a subset of X(sig), while all datasets in X(SigDT) either belong to X(sig) or are assigned high percentages (ranging from 97.4% to 99.4%) as measured by the combined p-value shown in the last row of Table 7. Next, we leverage the metric row of Table 7 to assess whether significance-verified clusterable datasets are more likely to yield high-quality k-partition results. In summary, the overall means of ACC, NMI, and ARI across all datasets in X(unsig) are 0.504, 0.129, and 0.108, respectively, while the corresponding overall means in X(sig) increase to 0.630, 0.404, and 0.373. From Figure 8, we conclude that clustering methods achieve better overall performance on datasets associated with a statistically significant partition. Moreover, the existence of statistically significant individual clusters or splits also contribute to improved performance. X(DV), X(sig), and X(SigDT) all show a higher median average rank with a clear distinction from X(unsig) across all metrics.

4.3.3 Robustness to Parametric Assumptions. There are two main parametric assumptions underlying K-MHTC and the combined p-value:

- The rOP method, which aggregates multiple single p-values across all attributes, depends on the selection of a parameter r. We set $r = \lfloor 0.5 \cdot M \rfloor$ by default and have demonstrated its effectiveness in Section 4.3.2. However, it remains to be investigated whether the combined p-value remains effective under alternative choices of r, i.e., how robust the method is to variations in r when different degrees of randomness are introduced.
- The Chi-square test requires a sufficient sample size in each cell of the contingency table. This assumption may be violated in some real-world categorical data, particularly in small datasets (i.e., when N is small). To assess this, we perform simulation experiments on small datasets with attached GT cluster labels that should consistently yield statistically significant partitions, and empirically evaluate how small N can be before the validity of the combined p-value is degraded.

Simulation Study for Various r: We adopt the same strategy used in Section 4.3.1 to generate partially randomized partitions with varying degrees of randomness. For each selected value of r, we plot a curve showing the median combined p-values as the randomization process is applied to the 25 GT partitions across all datasets. The parameter r is varied from $r = \lfloor 0.01 \cdot M \rfloor$ to $r = M$ over 100 values. As shown in the left subfigure of Figure 9, we observe that when $r = \lfloor 0.5 \cdot M \rfloor$, the method remains robust to up to 75% randomness introduced into the GT partitions. In particular, the median of the combined p-value remains below 0.01 at the 75% randomization level, indicating

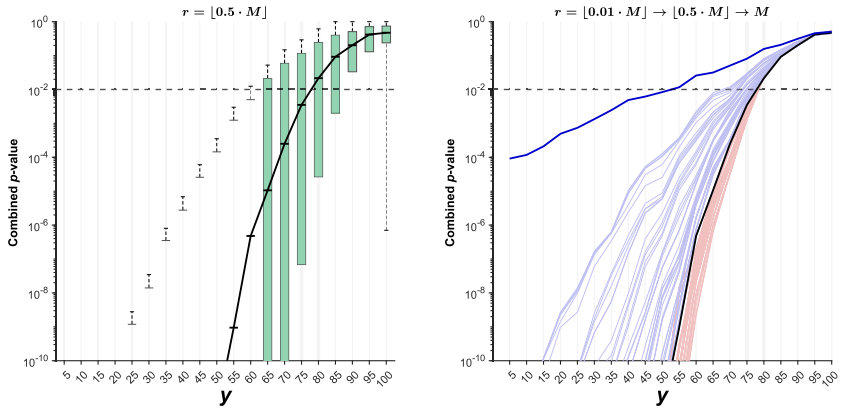


Fig. 9. Robustness of the combined p-values calculated via rOP against increasing randomness measured by the percentage (y%) of shuffled cluster labels, with the parameter set to the default $r = \lfloor 0.5 \cdot M \rfloor$ vs. various r. In the right subplot, r varies from $\lfloor 0.01 \cdot M \rfloor$ to $\lfloor 1 \cdot M \rfloor$ in 100 steps. Curves with $r < \lfloor 0.5 \cdot M \rfloor$ are shown in light red, those with $r > \lfloor 0.5 \cdot M \rfloor$ are shown in light blue, and the curve for $r = \lfloor 1 \cdot M \rfloor$ is marked in dark blue.

that such partially randomized partitions still tend to be identified as statistically significant. When the median curve for $r = \lfloor 0.5 \cdot M \rfloor$ is compared with those in the right subfigure of Figure 9, we observe that the combined p-values tend to be slightly smaller when $r < \lfloor 0.5 \cdot M \rfloor$ (light red curves), and slightly larger when $r > \lfloor 0.5 \cdot M \rfloor$ (light blue curves). Notably, when $r = M$ (dark blue curve), the combined p-values become substantially larger than those with $r \leq \lfloor 0.99 \cdot M \rfloor$. Nevertheless, the method still remains robust up to approximately 50% randomization, which is acceptable for potential practical use.

Simulation Study for Small Datasets: We examine three clusterable datasets, each consisting of two clusters, to isolate the effect of K. A small number of objects are sampled from each, and their GT partitions are evaluated using the combined p-value. As shown in Figure 10, the rate of misidentification varies with sample size. In general, the combined p-value becomes less reliable when the sample size N falls below 24. In particular, at $N = 4$, the misidentification rate can exceed 50%, rendering the result no longer usable. In contrast, as N increases, for instance when it exceeds 26, the combined p-value remains consistently valid. This is supported by the results for N between 26 and 200 shown in the figure, where almost no failures in identifying significant partitions are observed.

4.4 Case Study

To demonstrate the practical utility of our method in real-world clustering analysis, we select a representative dataset from our experiments that contains two well-defined clusters and interpret the clustering results in the context of attribute semantics. In the field of social science, the House Votes dataset is commonly used to quantify ideological divergence between two major groups, typically corresponding to political parties.

As shown in the left subfigure of Figure 11, the individual p-values associated with each voting issue (i.e., attribute) are displayed as bar plots, corresponding to the K-MHTC-derived partition and the GT partition, respectively. We observe that different voting issues contribute unequally to the disagreement between the two political groups (i.e., clusters). In other words, some issues play

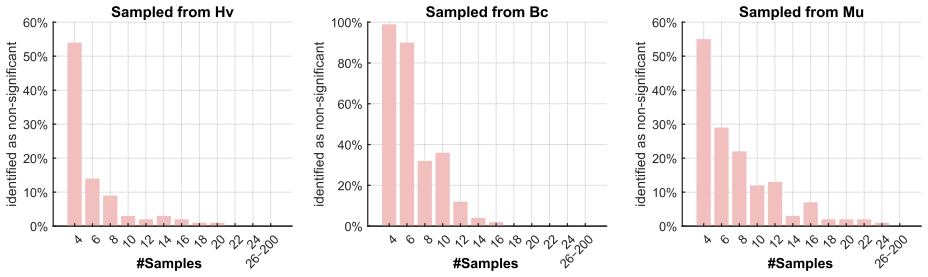
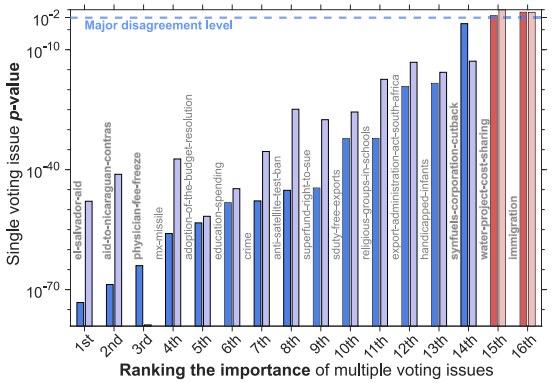


Fig. 10. Percentage of GT partitions misidentified as being non-significant on sampled small datasets. Each sample size is randomly drawn 100 times, generating 100 GT partitions. Hv, Bc, and Mu are chosen for their verified clustering tendency and sufficiently large samples per cluster. Objects from both clusters (each original dataset contains two clusters), along with their GT labels, are evenly sampled, with 2 to 100 samples per cluster, resulting in total sample sizes ranging from 4 to 200.



Rank	Issue	Republican	Democrat
1st	"el-salvador-aid"	Yea (96.5%)	Nay (86.9%)
2nd	"aid-to-nicaraguan-contras"	Nay (85.4%)	Yea (94.1%)
3rd	"physician-fee-freeze"	Yea (84.9%)	Nay (92.8%)
14th	"synfuels-corporation-cutback"	Nay (70.7%)	Nay (52.3%)
15th	"water-project-cost-sharing"	Yea (51.5%)	Yea (48.1%)
16th	"immigration"	Yea (50.0%)	Nay (49.4%)

Intra-group dominant opinion (**Yea** or **Nay**) per issue

Fig. 11. Case study: Interpreting clustering results on the House Votes dataset (Hv) by ranking the importance of voting issues.

a more critical role in shaping the division between clusters, while others exhibit less disagreement and thus provide limited discriminative power.

From the perspective of hypothesis testing, the significance level can be interpreted as a threshold for major disagreement: voting issues with individual p-values below this threshold are considered to reflect strong opposition between the two groups, whereas those above the threshold indicate relatively weak or non-salient disagreement. This observation is further illustrated in the right subfigure of Figure 11, where we compare the intra-group support rates (i.e., the proportion of members within each cluster that vote consistently) for the top three and bottom three ranked attributes. These examples intuitively demonstrate how attribute importance relates to intra-cluster compactness. For the two voting issues with non-significant p-values, we find that neither party exhibits a clear majority preference (i.e., support for either "Yea" or "Nay"), suggesting that these issues contribute insufficiently to the formation of opinion-based clusters.

5 Conclusion

To obtain a set of meaningful clusters from categorical data, we propose a new clustering algorithm, K-MHTC, which achieves high ACC and computational efficiency based on MHT. The statistical

significance of the resulting clustering structure is assessed by a combined p-value via meta-analysis. In contrast to existing significance-based clustering methods, it is the first time that the cluster analysis issue is formulated as an MHT problem.

Through extensive empirical studies, the benefit of our p-value-based method has been experimentally demonstrated in two main aspects: (1) If there is no clustering structure in a target categorical dataset, a quite large p-value will be delivered. It indicates that the clustering results are not recommended for further use. (2) If many standard and SOTA methods can achieve good clustering results in terms of extremely small p-values, then it means that a clustering structure really exists in the given dataset.

References

[1] Andreas Adolfsson, Margareta Ackerman, and Naomi C Brownstein. 2019. To cluster, or not to cluster: An analysis of clusterability methods. *Pattern Recognition* 88 (2019), 13–26.

[2] Amir Ahmad and Lipika Dey. 2007. A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set. *Pattern Recognition Letters* 28, 1 (2007), 110–118.

[3] Murat O. Ahmed and Guenther Walther. 2012. Investigating the multimodality of multivariate data with principal curves. *Computational Statistics & Data Analysis* 56, 12 (2012), 4462–4469.

[4] Naomi Altman and Martin Krzywinski. 2017. Tabular data. *Nature Methods* 14, 4 (2017), 329–331.

[5] Razia Azen and Cindy M. Walker. 2021. *Categorical Data Analysis for the Behavioral and Social Sciences*. Routledge.

[6] Liang Bai and Jiye Liang. 2015. Cluster validity functions for categorical data: A solution-space perspective. *Data Mining and Knowledge Discovery* 29, 6 (2015), 1560–1597.

[7] Liang Bai and Jiye Liang. 2022. A categorical data clustering framework on graph representation. *Pattern Recognition* 128 (2022), 108694.

[8] Liang Bai, Jiye Liang, Chuangyin Dang, and Fuyuan Cao. 2013. The impact of cluster representatives on the convergence of the K-modes type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 6 (2013), 1509–1522.

[9] A. Banerjee and R. N. Dave. 2004. Validating clusters using the Hopkins statistic. In *Proceedings of the 2004 IEEE International Conference on Fuzzy Systems*, Vol. 1, 149–153.

[10] Daniel Barbará, Yi Li, and Julia Couto. 2002. COOLCAT: An entropy-based algorithm for categorical clustering. In *Proceedings of the 11th International Conference on Information and Knowledge Management*, 582–589.

[11] Debajyoti Bera, Rameshwar Pratap, and Bhisham Dev Verma. 2023. Dimensionality reduction for categorical data. *IEEE Transactions on Knowledge and Data Engineering* 35, 4 (2023), 3658–3671.

[12] Hans-Hermann Bock. 1985. On some significance tests in cluster analysis. *Journal of Classification* 2 (1985), 77–108.

[13] Hans H. Bock. 1996. Probabilistic models in cluster analysis. *Computational Statistics & Data Analysis* 23, 1 (1996), 5–28.

[14] Michael Borenstein, Larry V. Hedges, Julian P. T. Higgins, and Hannah R. Rothstein. 2021. *Introduction to Meta-Analysis*. John Wiley & Sons.

[15] Shyam Boriah, Varun Chandola, and Vipin Kumar. 2008. Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the 2008 SIAM International Conference on Data Mining*. SIAM, 243–254.

[16] Mohamed Bouguessa. 2015. Clustering categorical data in projected spaces. *Data Mining and Knowledge Discovery* 29, 1 (2015), 3–38.

[17] Stephen P. Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

[18] Dario Bruzzese and Domenico Vistocco. 2015. DESPOTA: DEndrogram slicing through a permutation test approach. *Journal of Classification* 32, 2 (2015), 285–304.

[19] Deng Cai, Xiaofei He, and Jiawei Han. 2005. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering* 17, 12 (2005), 1624–1637.

[20] Lun-Ching Chang, Hui-Min Lin, Etienne Sibille, and George C. Tseng. 2013. Meta-analysis methods for combining multiple expression profiles: Comparisons, statistical characterization and an application guideline. *BMC Bioinformatics* 14, 1 (2013), 1–15.

[21] Paraskevi Chasani and Aristidis Likas. 2022. The UU-test for statistical modeling of unimodal data. *Pattern Recognition* 122 (2022), 108272.

[22] Peter Cheeseman and John Stutz. 1996. Bayesian classification (AutoClass): Theory and results. *In Advances in Knowledge Discovery and Data Mining*, 153–180.

[23] Chao Chen and Novi Quadrianto. 2016. Clustering high dimensional categorical data via topographical features. In *Proceedings of the International Conference on Machine Learning*, PMLR, 2732–2740.

[24] Hung-Leng Chen, Kun-Ta Chuang, and Ming-Syan Chen. 2008. On data labeling for clustering categorical data. *IEEE Transactions on Knowledge and Data Engineering* 20, 11 (2008), 1458–1472.
[25] Keke Chen and Ling Liu. 2009. He-tree: A framework for detecting changes in clustering structure for categorical data streams. *The VLDB Journal* 18, 6 (2009), 1241–1260.
[26] Tao Chen, Nevin L. Zhang, Tengfei Liu, Kin Man Poon, and Yi Wang. 2012. Model-based multidimensional clustering of categorical data. *Artificial Intelligence* 176, 1 (2012), 2246–2269.
[27] Yiqun T. Chen and Daniela M. Witten. 2023. Selective inference for k-means clustering. *Journal of Machine Learning Research* 24, 152 (2023), 1–41.
[28] M.-Y. Cheng and Peter Hall. 1998. Calibrating the excess mass and dip tests of modality. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 60, 3 (1998), 579–589.
[29] Eugene Demidenko. 2018. The next-generation K-means algorithm. *Statistical Analysis and Data Mining* 11, 4 (2018), 153–166.
[30] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7 (2006), 1–30.
[31] Tai Dinh, Wong Hauchi, Philippe Fournier-Viger, Daniil Lisik, Minh-Quyet Ha, Hieu-Chi Dam, and Van-Nam Huynh. 2025. Categorical data clustering: 25 years beyond K-modes. *Expert Systems with Applications* 272 (2025), 126608.
[32] Dheeru Dua and Casey Graff. 2017. UCI machine learning repository. Retrieved from http://archive.ics.uci.edu/ml
[33] Richard C. Dubes and Guangzhou Zeng. 1987. A test for spatial homogeneity in cluster analysis. *Journal of Classification* 4, 1 (1987), 33–56.
[34] Kirill Efimov, Larisa Adamyan, and Vladimir Spokoiny. 2019. Adaptive nonparametric clustering. *IEEE Transactions on Information Theory* 65, 8 (2019), 4875–4892.
[35] Douglas H. Fisher. 1987. Knowledge acquisition via incremental conceptual clustering. *Machine Learning* 2, 2 (1987), 139–172.
[36] Jerome H. Friedman and Lawrence C. Rafsky. 1981. Graphics for the multivariate two-sample problem. *Journal of the American Statistical Association* 76, 374 (1981), 277–287.
[37] Claudio Fuentes and George Casella. 2009. Testing for the existence of clusters. *SORT (Barcelona)* 33, 2 (2009), 115.
[38] María Teresa Gallegos and Gunter Ritter. 2018. Probabilistic clustering via Pareto solutions and significance tests. *Advances in Data Analysis and Classification* 12, 2 (2018), 179–202.
[39] Venkatesh Ganti, Johannes Gehrke, and Raghu Ramakrishnan. 1999. CACTUS—Clustering categorical data using summaries. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 73–83.
[40] Lucy L. Gao, Jacob Bien, and Daniela Witten. 2024. Selective inference for hierarchical clustering. *Journal of the American Statistical Association* 119, 545 (2024), 332–342.
[41] Allan D. Gordon. 1996. Null models in cluster validation. In *From Data to Knowledge*. Springer, 32–44.
[42] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. 2000. ROCK: A robust clustering algorithm for categorical attributes. *Information Systems* 25, 5 (2000), 345–366.
[43] Erika S. Helgeson, David M. Vock, and Eric Bair. 2021. Nonparametric cluster significance testing with reference to a unimodal null distribution. *Biometrics* 77, 4 (2021), 1215–1226.
[44] Christian Hennig. 2015. What are the true clusters? *Pattern Recognition Letters* 64 (2015), 53–62.
[45] Lianyu Hu, Junjie Dong, Mudi Jiang, Yan Liu, and Zengyou He. 2025. Clusterability test for categorical data. *Knowledge and Information Systems* 67, 5 (2025), 4113–4138.
[46] Lianyu Hu, Mudi Jiang, Junjie Dong, Xinying Liu, and Zengyou He. 2025. Interpretable categorical data clustering via hypothesis testing. *Pattern Recognition* 162 (2025), 111364.
[47] Lianyu Hu, Mudi Jiang, Xinying Liu, and Zengyou He. 2025. Significance-based decision tree for interpretable categorical data clustering. *Information Sciences* 690 (2025), 121588.
[48] Hanwen Huang, Yufeng Liu, Ming Yuan, and J. S. Marron. 2015. Statistical significance of clustering using soft thresholding. *Journal of Computational and Graphical Statistics* 24, 4 (2015), 975–993.
[49] J. Z. Huang, M. K. Ng, Hongqiang Rong, and Zichen Li. 2005. Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 5 (2005), 657–668.
[50] Jincai Huang and Jianbo Tang. 2021. Discovery of arbitrarily shaped significant clusters in spatial point data with noise. *Applied Soft Computing* 108 (2021), 107452.
[51] Zhexue Huang. 1998. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery* 2, 3 (1998), 283–304.
[52] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification* 2, 1 (1985), 193–218.
[53] Natthakan Iam-On, Tossapon Boongoen, Simon Garrett, and Chris Price. 2011. A link-based approach to the cluster ensemble problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 12 (2011), 2396–2409.

[54] A. K. Jain, Xiaowei Xu, Tin Kam Ho, and Fan Xiao. 2002. Uniformity testing using minimal spanning tree. *In Proceedings of the 2002 International Conference on Pattern Recognition*, Vol. 4, 281–284.
[55] Anil K. Jain. 2010. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters* 31, 8 (2010), 651–666.
[56] Hong Jia and Yiu-Ming Cheung. 2018. Subspace clustering of categorical and numerical data with an unknown number of clusters. *IEEE Transactions on Neural Networks and Learning Systems* 29, 8 (2018), 3308–3325.
[57] Hong Jia, Yiu-Ming Cheung, and Jiming Liu. 2016. A new distance metric for unsupervised learning of categorical data. *IEEE Transactions on Neural Networks and Learning Systems* 27, 5 (2016), 1065–1079.
[58] Songlei Jian, Longbing Cao, Kai Lu, and Hang Gao. 2018. Unsupervised coupled metric similarity for non-iid categorical data. *IEEE Transactions on Knowledge and Data Engineering* 30, 9 (2018), 1810–1823.
[59] Songlei Jian, Guansong Pang, Longbing Cao, Kai Lu, and Hang Gao. 2019. CURE: Flexible categorical data representation by hierarchical coupling learning. *IEEE Transactions on Knowledge and Data Engineering* 31, 5 (2019), 853–866.
[60] Argyris Kalogeratos and Aristidis Likas. 2012. Dip-means: An incremental clustering method for estimating the number of clusters. *In Advances in Neural Information Processing Systems*, Vol. 25.
[61] Patrick K. Kimes, Yufeng Liu, David Neil Hayes, and James Stephen Marron. 2017. Statistical significance for hierarchical clustering. *Biometrics* 73, 3 (2017), 811–821.
[62] Tao Li, Sheng Ma, and Mitsunori Ogihara. 2004. Entropy-based criterion in categorical clustering. In *Proceedings of the 21st International Conference on Machine Learning*, 68.
[63] Robert F. Ling. 1973. A probability theory of cluster analysis. *Journal of the American Statistical Association* 68, 341 (1973), 159–164.
[64] Robert F. Ling and George G. Killough. 1976. Probability tables for cluster analysis based on a theory of random graphs. *Journal of the American Statistical Association* 71, 354 (1976), 293–300.
[65] Hongfu Liu, Junjie Wu, Tongliang Liu, Dacheng Tao, and Yun Fu. 2017. Spectral ensemble clustering via weighted K-means: Theoretical and practical evidence. *IEEE Transactions on Knowledge and Data Engineering* 29, 5 (2017), 1129–1143.
[66] Yufeng Liu, David Neil Hayes, Andrew Nobel, and James Stephen Marron. 2008. Statistical significance of clustering for high-dimension, low–sample size data. *Journal of the American Statistical Association* 103, 483 (2008), 1281–1293.
[67] Ranjan Maitra, Volodymyr Melnykov, and Soumendra N. Lahiri. 2012. Bootstrapping for significance of compact clusters in multidimensional datasets. *Journal of the American Statistical Association* 107, 497 (2012), 378–392.
[68] Samuel Maurus and Claudia Plant. 2016. Skinny-dip: Clustering in a sea of noise. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1055–1064.
[69] Boris Mirkin. 2001. Reinterpreting the category utility function. *Machine Learning* 45, 2 (2001), 219–228.
[70] Michael K. Ng, Mark Junjie Li, Joshua Zhexue Huang, and Zengyou He. 2007. On the impact of dissimilarity measure in k-modes clustering algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 3 (2007), 503–507.
[71] Tin Nguyen, Rebecca Tagett, Michele Donato, Cristina Mitrea, and Sorin Draghici. 2016. A novel bi-level meta-analysis approach: Applied to biological pathway analysis. *Bioinformatics* 32, 3 (2016), 409–416.
[72] Adam Petrie and Thomas R. Willemain. 2013. An empirical study of tests for uniformity in multidimensional data. *Computational Statistics & Data Analysis* 64 (2013), 253–268.
[73] Yuhua Qian, Feijiang Li, Jiye Liang, Bing Liu, and Chuangyin Dang. 2016. Space structure and clustering of categorical data. *IEEE Transactions on Neural Networks and Learning Systems* 27, 10 (2016), 2047–2059.
[74] Gregory Paul M. Rozál and J. A. Hartigan. 1994. The MAP test for multimodality. *Journal of Classification* 11, 1 (1994), 5–36.
[75] Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 8 (2000), 888–905.
[76] Ravid Shwartz-Ziv and Amitai Armon. 2022. Tabular data: Deep learning is not all you need. *Information Fusion* 81 (2022), 84–90.
[77] Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouët. 2018. Are your data gathered? In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2210–2218.
[78] Bernard W. Silverman. 1981. Using kernel density estimates to investigate multimodality. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 43, 1 (1981), 97–99.
[79] S. P. Smith. 1993. Threshold validity for mutual neighborhood clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 1 (1993), 89–92.
[80] Stephen P. Smith and Anil K. Jain. 1984. Testing for uniformity in multidimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1 (1984), 73–81.
[81] Chi Song and George C. Tseng. 2014. Hypothesis setting and order statistic for robust genomic meta-analysis. *The Annals of Applied Statistics* 8, 2 (2014), 777.
[82] Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3 (Dec. 2002), 583–617.

[83] Marcio Valk and Gabriela Bettella Cybis. 2021. U-statistical inference for hierarchical clustering. *Journal of Computational and Graphical Statistics* 30, 1 (2021), 133–143.
[84] Jingshu Wang and Art B. Owen. 2019. Admissibility in partial conjunction testing. *Journal of the American Statistical Association* 114, 525 (2019), 158–168.
[85] Yiyong Xiao, Changhao Huang, Jiaoying Huang, Ikou Kaku, and Yuchun Xu. 2019. Optimal mathematical programming and variable neighborhood search for k-modes categorical data clustering. *Pattern Recognition* 90 (2019), 183–195.
[86] Yiqun Xie, Xiaowei Jia, Shashi Shekhar, Han Bao, and Xun Zhou. 2021. Significant DBSCAN+: Statistically robust density-based clustering. *ACM Transactions on Intelligent Systems and Technology* 12, 5 (2021), 1–26.
[87] Tengke Xiong, Shengrui Wang, André Mayers, and Ernest Monga. 2012. DHCC: Divisive hierarchical clustering of categorical data. *Data Mining and Knowledge Discovery* 24, 1 (2012), 103–135.
[88] Peng Zhang, Xiaogang Wang, and Peter X.-K Song. 2006. Clustering categorical data based on distance vectors. *Journal of the American Statistical Association* 101, 473 (2006), 355–367.
[89] Yiqun Zhang and Yiu-Ming Cheung. 2022. Learnable weighting of intra-attribute distances for categorical data clustering with nominal and ordinal attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 7 (2022), 3560–3576.
[90] Yiqun Zhang and Yiu-Ming Cheung. 2023. Graph-based dissimilarity measurement for cluster analysis of any-type-attributed data. *IEEE Transactions on Neural Networks and Learning Systems* 34, 9 (2023), 6530–6544.
[91] Yiqun Zhang, Yiu-Ming Cheung, and An Zeng. 2022. Het2Hom: Representation of heterogeneous attributes into homogeneous concept spaces for categorical-and-numerical-attribute data clustering. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, 3758–3765.
[92] Mingjie Zhao, Sen Feng, Yiqun Zhang, Mengke Li, Yang Lu, and Yiu-Ming Cheung. 2024. Learning order forest for qualitative-attribute data clustering. In *Proceedings of the 27th European Conference on Artificial Intelligence*, 1943–1950.
[93] Xingwang Zhao, Jiye Liang, and Chuangyin Dang. 2017. Clustering ensemble selection for categorical data based on internal validity indices. *Pattern Recognition* 69 (2017), 150–168.
[94] Qibin Zheng, Xingchun Diao, Jianjun Cao, Yi Liu, Hongmei Li, Junnan Yao, Chen Chang, and Guojun Lv. 2020. From whole to part: Reference-based representation for clustering categorical data. *IEEE Transactions on Neural Networks and Learning Systems* 31, 3 (2020), 927–937.
[95] Chengzhang Zhu, Longbing Cao, and Jianping Yin. 2022. Unsupervised heterogeneous coupling learning for categorical representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 1 (2022), 533–549.

Received 26 November 2024; revised 26 March 2025; accepted 8 May 2025