SPARTAN: Data-Adaptive Symbolic Time-Series Approximation

Fan Yang
yang.7007@osu.edu
The Ohio State University
Columbus, Ohio, USA

John Paparrizos
paparrizos.1@osu.edu
The Ohio State University
Columbus, Ohio, USA

ABSTRACT

Symbolic approximations are dimensionality reduction techniques that convert time series into sequences of discrete symbols, enhancing interpretability while reducing computational and storage costs. To construct symbolic representations, first numeric representations approximate and capture properties of raw time series, followed by a discretization step that converts these numeric dimensions into symbols. Despite decades of development, existing approaches have several key limitations that often result in unsatisfactory performance: they (i) rely on data-agnostic numeric approximations, disregarding intrinsic properties of the time series; (ii) decompose dimensions into equal-sized subspaces, assuming independence among dimensions; and (iii) allocate a uniform encoding budget for discretizing each dimension or subspace, assuming balanced importance. To address these shortcomings, we propose SPARTAN, a novel data-adaptive symbolic approximation method that intelligently allocates the encoding budget according to the importance of the constructed uncorrelated dimensions. Specifically, SPARTAN (i) leverages intrinsic dimensionality reduction properties to derive non-overlapping, uncorrelated latent dimensions; (ii) adaptively distributes the budget based on the importance of each dimension by solving a constrained optimization problem; and (iii) prevents false dismissals in similarity search by ensuring a lower bound on the true distance in the original space. To demonstrate SPARTAN's robustness, we conduct the most comprehensive study to date, comparing SPARTAN with seven state-of-the-art symbolic methods across four tasks: classification, clustering, indexing, and anomaly detection. Rigorous statistical analysis across hundreds of datasets shows that SPARTAN outperforms competing methods significantly on *all* tasks in terms of downstream accuracy, given the same budget. Notably, SPARTAN achieves up to a 2x speedup compared to the most accurate rival. Overall, SPARTAN effectively improves the symbolic representation quality without storage or runtime overheads, paving the way for future advancements.

KEYWORDS

Time-series Analysis, Symbolic Representation

ACM Reference Format:
Fan Yang and John Paparrizos. 2018. SPARTAN: Data-Adaptive Symbolic Time-Series Approximation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 16 pages. https://doi.org/XXXXXXX.XXXXXXX

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference acronym 'XX, June 03–05, 2018, Woodstock, NY
© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

Table 1: Critical features of symbolic methods. Complexity results are estimated by n time series with length m, alphabet size α, and word length ω. s denotes the number of downsampled data. k denotes the number of iterations for clustering. The "Lower Bounding" column displays whether the representation guarantees no false dismissals in similarity search. "Data-dependent Approximation" shows whether the numeric approximation learns characteristics from the entire dataset. "Dynamic Discretization" indicates whether the budget can be dynamically allocated across each symbol.

Method	Lower Bounding	Data-dependent Approximation	Dynamic Discretization	Complexity (training-stage)	Complexity (inference-stage)
SAX [42]	✓	–	–	$O(nm)$	$O(nm)$
ESAX [46]	–	–	–	$O(nm)$	$O(nm)$
TFSAX [79]	–	–	–	$O(nm)$	$O(nm)$
SAX-DR [39]	–	–	–	$O(nm)$	$O(nm)$
SAX-VFD [76]	–	✓	–	$O(nm\log(nm))$	$O(nm\log(nm))$
1d-SAX [49]	–	–	–	$O(nm)$	$O(nm)$
SFA [65]	✓	–	–	$O(nm\log(m))$	$O(nm\log(m))$
ABBA [17]	–	–	–	$O(nm + n\alpha^2\omega k)$	$O(nm + n\alpha^2\omega k)$
fABBA [12]	–	–	–	$O(nm + n\omega\log\omega)$	$O(nm + n\omega\alpha)$
Newly Proposed Symbolic Representation Solution					
SPARTAN	✓	✓	✓	$O(nm^2)$	$O(nm\omega)$
SPARTAN-R	✓	✓	✓	$O(nm\omega)$	$O(nm\omega)$
SPARTAN-S	✓	✓	✓	$O(sm\omega)$	$O(nm\omega)$

1 INTRODUCTION

Time series, an ordered sequence of real-valued observations, have become prevalent across a wide range of domains, including environmental science, biology, engineering, astronomy, and finance [3, 8, 13, 30, 40, 72, 74, 77]. Over the past few decades, the ubiquity of time series has driven the development of methods for diverse analytical tasks such as classification [7, 13], clustering [58], forecasting [26], anomaly detection [59, 66], motif discovery [43, 47], and similarity search [16, 56], highlighting the importance of processing and mining such data. However, the proliferation of high-dimensional time-series data from expanding Internet of Things (IoT) deployments [48] poses significant challenges in computational and storage costs [48, 56, 57], emphasizing the necessity of efficient and scalable analysis techniques for downstream tasks.

Symbolic approximation, which transforms raw time series into sequences of discrete symbols like "AABCBBCD," has been proposed as a solution. Beyond dimensionality reduction benefits (e.g., low computational and storage cost) shared with compression techniques [2, 11, 20, 22, 35, 56], symbolic approximation methods (referred to as symbolic methods hereafter) provide additional advantages stemming from their symbolic nature. In particular, symbolic methods: (i) offer high interpretability by generating symbolic representation; (ii) avail the wealth of diverse techniques from other domains, such as text processing [69, 70], facilitating the discovery of

meaningful patterns [42, 43]; (iii) prevent false dismissals in similarity search tasks by preserving a lower bound on the true distance in the original space. For the past two decades, symbolic methods have become a subroutine in diverse analytical tasks, such as dictionary classifiers [44, 45, 54, 64, 67, 71], anomaly detection [10, 43, 68–70], indexing [42, 65, 73], and motif discovery [41–43]. Nowadays with the flourishing of Large Language Models (LLMs) [24, 31, 83], symbolic methods, by their nature, hold potential as a bridge between the continuous time series and discrete text data, which highlights the importance of attention paid to their development.

To extract meaningful temporal information, symbolic methods typically involve two crucial steps: (i) *numeric approximation*, which approximates the raw data with a numeric representation while preserving the essential information; (ii) *discretization*, which applies a predefined or learned vocabulary to transform the approximated data to a sequence of symbols. For example, as one well-established symbolic method, Symbolic Aggregate Approximation (SAX) [42, 43] reduces the dimensionality by evenly segmenting the data using Piecewise Aggregate Approximation (PAA) [37]. The mean values of PAA segments are mapped to discrete symbols based on a predefined lookup table, assuming a Gaussian distribution. However, relying solely on the average values of input segments can lead to suboptimal performance, especially with higher dimensionality. This drawback has prompted the development of several SAX variants [39, 46, 49, 76, 79], which incorporate additional representative features (e.g., statistics and trends). Yet most of these solutions have improved representation power by sacrificing space/increasing storage requirements and lack a proven lower-bounding property. Other methods reduce the reconstruction error by adopting adaptive segments but produce variable-sized representations per time series [12, 17], making them less suitable for batch processing in downstream analysis. In contrast, Symbolic Fourier Analysis (SFA) [65] leverages the advantage of Discrete Fourier Transform (DFT) approximation and retains the desirable lower bounding property, making it one of the most widely used methods in similarity search and dictionary classifiers [51, 52, 54, 64, 65, 67].

Despite decades of progress, current approaches still exhibit several key limitations that may lead to unsatisfactory performance. Specifically, most existing methods: (i) rely on data-agnostic approximation strategies, neglecting the intrinsic properties of the time series; (ii) decompose raw dimensions into equal-sized subspaces, assuming independence across the dimensions; and (iii) assign a uniform budget across dimensions or subspaces, assuming equal importance for all dimensions. While the first limitation prevents methods from effectively modeling intrinsic data properties, the latter two overlook the temporal dependencies and the disproportionate contributions of different dimensions (as illustrated in Section 3.1). Consequently, these shortcomings lead to inefficient resource allocation by wasting budgets on insignificant dimensions. Furthermore, no comprehensive comparative evaluation of symbolic methods currently exists, leaving their impact on downstream accuracy unclear. We challenge and address this as part of our work.

In this work, we present the **S**ymbolic **P**C**A** **R**epresentation for **T**ime-series **A**pproximatio**N** (**SPARTAN**), a novel symbolic approximation method that overcomes the aforementioned three key limitations in both numeric approximation and discretization phases. Specifically, SPARTAN (i) exploits intrinsic dimensionality reduction to extract uncorrelated latent dimensions for approximation; (ii) formulates a constrained optimization problem to adaptively distribute the encoding budget across dimensions, in proportion to their importance; and (iii) guarantees a lower bound on the Euclidean distance in the original space, preventing false dismissals in the similarity search tasks [43, 65, 73]. By intelligently allocating the encoding budget according to the significance of the uncorrelated latent dimensions, SPARTAN efficiently enhances symbolic representation quality without sacrificing the encoding budget.

To demonstrate the robustness of SPARTAN, we perform the most comprehensive experimental study on symbolic representations to date, along with seven state-of-the-art methods. Specifically, we evaluate SPARTAN and leading methods across four analytical tasks: classification, clustering, and the tightness of lower bound (a proxy for indexing) [42, 43, 65] across 128 UCR datasets [13], and anomaly detection on about 2000 time series from TSB-UAD datasets [59]. Existing symbolic distances are often tailored for their respective methods, which complicates the fair evaluation of different representations. Given the absence of a unified testbed, we propose a generic symbolic distance, *Symbolic* L_1, to measure dissimilarity between symbolic representations. This approach requires no prior knowledge of the underlying methods, which helps ensure fairness. Additionally, we evaluate Euclidean distance (ED) on raw time series as a valuable reference point in comparison with symbolic methods. Importantly, we validate our results through rigorous statistical tests to assess the significance of performance differences. We make the code available for reproducibility [1].

In summary, our evaluation study demonstrates: (i) SAX variants outperform SAX by sacrificing storage, yet none surpass SAX under the same budget, reinforcing SAX as a strong baseline; SFA is the only method that strongly outperforms SAX under the same budget, showing its robust representation quality; (ii) SPARTAN demonstrates superior representation power over two leading methods, SAX and SFA, across all analytical tasks under the same storage budget; and (iii) SPARTAN also offers accelerated versions that deliver up to a 2x speedup over SFA on large-scale databases with millions of time series, without compromising its representation quality. As shown above, SPARTAN emerges as a robust symbolic method, advancing the state of the art. Table 1 summarizes the characteristics of SPARTAN against baselines.

In this paper, we begin with a review of the preliminary and related work (Section 2). We then present the new method as follows:

- We show the disproportionate nature of importance distribution across dimensions using representative samples (Section 3.1).
- We derive non-overlapping latent dimensions and efficiently measure the informativeness by leveraging the intrinsic property of linear dimensionality reduction (Section 3.1).
- We formulate a constrained optimization problem with a dynamic programming solution to adaptively assign encoding budget space in proportion to the importance (Section 3.2).
- We formally prove the lower bound property of SPARTAN to Euclidean distance on the original space. (Section 3.3).
- We demonstrate the strength of SPARTAN by conducting a comprehensive evaluation against the current state-of-the-art methods under four analytical tasks (Section 4 and 5).

Finally, we discuss the conclusions of the work and potential directions for future symbolic representations (Section 6).

2 BACKGROUND AND PRELIMINARIES

We first review the terminology for time series (Section 2.1), and we introduce the time-series representation across different categories (Section 2.2). We will then present the current state-of-the-art method in time-series symbolic representation (Section 2.3).

2.1 Terminology and Definitions

Time Series: A set of n time-series, denoted as $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$, where each $X_i \in \mathbb{R}^{m \times d}$, with m time steps and d channels. A time-series is *univariate* when $d = 1$, and *multivariate* when $d > 1$. In this paper, we focus on the univariate case for simplicity.

Sliding Window: Given a univariate time-series X with length m, sliding windows are defined as a set of subsequences, i.e., a sequence of consecutive time steps, extracted by sliding a "window" of width w across all time steps, with a stride size c ($1 \leq w \leq m-1$ and $1 \leq c \leq m-w$). The shape of the output matrix is $(\lfloor \frac{m-w}{c} \rfloor + 1, w)$.

Symbolic Representation: Given a time-series $X \in \mathbb{R}^{m \times d}$, we want to find a transformation f from the raw data to a word of ω symbols: $\widetilde{X} = f(X)$, where $\widetilde{X} \in \Phi^{\omega}$ and $\Phi = \{s_1, s_2, \cdots, s_\alpha\}$ denotes the vocabulary set given alphabet size α ($\alpha, \omega \ll m$). The transformation typically consists of two critical steps: (i) *approximation*, which approximates the data with a low-dimensional representation $\hat{X} = \psi(X)$, $\hat{X} \in \mathbb{R}^{\omega}$; (ii) *discretization*, which transforms the approximated data to a sequence of symbols $\widetilde{X} = \phi(\hat{X})$, $\widetilde{X} \in \Phi^{\omega}$.

2.2 Time-Series Representation

Curse of Dimensionality [36] inevitably raises performance concerns in both runtime and accuracy. It is useful to develop transformations that reduce the dimensions while retaining the essential information. Depending on the type of data, time-series representation can be classified into: *continuous* and *discrete* representations.

Continuous Representation: Examples of approximation techniques include Piecewise Aggregate Approximation (PAA) [35], Discrete Fourier Transform (DFT) [2], Discrete Wavelet Transform (DWT) [11], and Principal Component Analysis (PCA) [33], and Generic RepresentAtIon Learning (GRAIL) [57]. While data-agnostic methods like DFT offer reliable approximation by providing a universal basis across domains, data-adaptive methods like PCA can efficiently model underlying data distribution. With the advent of deep neural networks, unsupervised representation-learning strategies [75, 78, 80–82] and foundation models [5, 23, 31, 83] have attracted significant attention. However, their reliance on large training sets and high computational cost makes them less suitable for data-scarce or time-sensitive tasks in practical applications.

Discrete Representation: There are numerous studies on discrete time-series representations (symbolizing, tokenizing, quantizing) [4, 14, 20, 29, 43, 56, 73]. These techniques take advantage of the fast processing speed and low storage cost from the discrete data structure. Compared with these discrete solutions, symbolic methods offer high interpretability with human-readable symbols and leverage the wealth of text-based techniques [6, 43, 44, 70]. Representative approaches such as SAX [42, 43] and SFA [65] also preserve the lower bounding property, preventing the false dismissal in similarity search tasks [42, 65, 73]. In the next section, we review the current state-of-the-art symbolic methods.

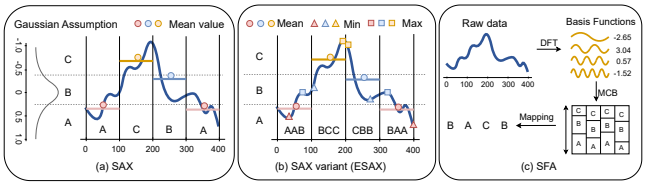

Figure 1: An overview of symbolic approximation methods. (a) SAX, a classical symbolic method. (b) An example of SAX variant method (ESAX). (c) SFA, a frequency-based method.

2.3 Symbolic Representation Methods

Symbolic representations have emerged as an effective tool for various downstream tasks. In the following, we review well-established symbolic methods, including SAX, SAX variants, and SFA.

Symbolic Aggregate Approximation (SAX): SAX [42] is a well-established symbolic method known for its robust performance and high efficiency. As shown in Figure 1, SAX reduces the dimensionality by evenly segmenting the raw data using PAA [37], and then maps the mean value of each PAA segment to a discrete symbol based on a predefined look-up table, assuming a Gaussian distribution. A MINDIST function [42, 43] is introduced to provide a symbolic distance that guarantees a lower bound on the Euclidean distance in the original space [43]. As mentioned earlier, this property serves as a cornerstone by preventing false dismissals in similarity search [43, 73]. Building on SAX, iSAX [73] has achieved great success in indexing tasks. iSAX attempts to create a multiresolution representation, with the goal of increasing precision by requesting additional budget when splitting nodes. However, as discussed in Section 3.2, our approach differs substantially from this logic: SPARTAN intelligently allocates the budget across dimensions *within the same budget*. Instead, iSAX doubles the budget at each level, which is orthogonal to the problem we study here.

SAX Variants: While PAA offers an effective approximation for each segment, it may incur significant information loss when applied to high-dimensional data. To address this issue, numerous variants of SAX have been proposed to encode additional features (Figure 1). For example, ESAX [46] utilizes similar techniques as PAA and additionally includes the maximum and minimum value for representation. To capture the trend feature, approaches like 1d-sax [49], SAX-DR [39], and TFSAX [79] introduce an additional feature derived from the linear approximation idea within the segment, enhancing the ability to represent the trend of the data. SAX-VFD [76] adopts 18 features from three categories, with an optimal feature selection algorithm proposed for informative features. Recently, ABBA [17] and fABBA [12] explored the direction of uneven segmentation strategies in the approximation, achieving lower reconstruction error under a given approximation tolerance. Yet, most of these variant solutions lack a proven lower-bounding property similar to SAX, limiting their applicability in downstream tasks.

Symbolic Fourier Approximation (SFA): Proposed in [65], SFA generates a symbolic representation of a time series by first computing the DFT coefficients. For word length of ω, and alphabet size α, the first $\frac{\omega}{2}$ coefficients are kept for approxmiation. Each coefficient is split into real and imaginary parts and discretized into one of α bins based on the process of Multiple Coefficient Binning (MCB) [65], which ensures the preservation of the lower-bounding

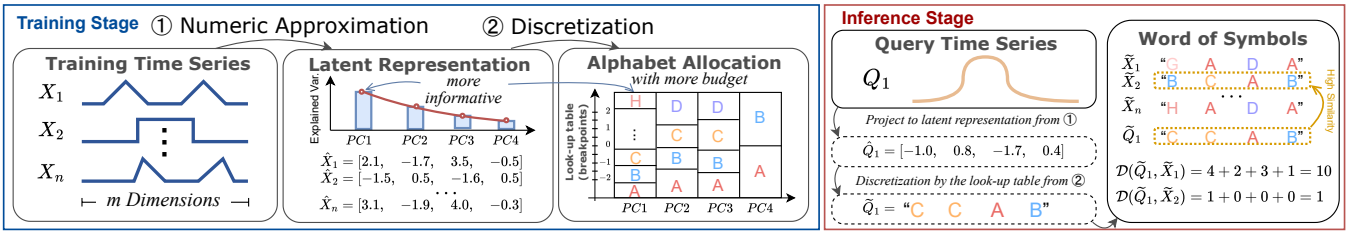

Figure 2: An overview of SPARTAN pipeline: (a) training stage, including the data-dependent approximation and dynamic discretization; and (b) inference stage, where query samples can be transformed based on learned parameters.

property, as proven in [65]. SFA's representation power comes from the DFT's ability to approximate real-valued signals in the frequency domain. However, the default SFA representation may introduce an inherent bias towards low-frequency information. Recent dictionary classifiers utilizing SFA mitigate this by applying supervised feature selection strategies, e.g., ANOVA test to select informative coefficients based on the class labels [51, 67]. However, in our work, we focus on unsupervised solutions for constructing symbolic representations which we then plug into 3 downstream tasks to demonstrate the representation quality. Our goal is not to create a SOTA dictionary-based classifier. While supervision can enhance symbolic methods, it is out of scope for this work.

Existing Limitations: As previously mentioned, existing solutions still share some key shortcomings that may lead to suboptimal performance. Specifically, most existing methods (i) rely on data-agnostic numeric approximation strategies, e.g., PAA and DFT, ignoring the underlying data properties; (ii) partition raw dimensions into equal-sized subspaces, assuming independence across dimensions; and (iii) allocate a uniform budget for each dimension, disregarding the disproportionate importance (as we will show in Section 3.1). Recent advancements made an attempt to address the limitations by applying uneven segmentation strategies [12, 17], or asymmetric alphabet sizes for different features [49], However, these approaches neither capture the intrinsic data properties nor offer an automated solution for adaptive budget allocation. A unified solution to address these challenges simultaneously is still lacking. In the following sections, we demonstrate how the SPARTAN method addresses the existing limitations from both numeric approximation and discretization stages.

3 THE SPARTAN METHOD

We propose SPARTAN, a novel data-adaptive symbolic approximation method that addresses existing limitations. Specifically, we first focus on measuring the informativeness for each dimension and deriving non-overlapping, uncorrelated latent dimensions by exploiting the intrinsic linear dimensionality reduction properties (Section 3.1). Then, we introduce a non-uniform policy to adaptively distribute the budget for discretization proportionally to the importance of each latent dimension, with a dynamic programming solution (Section 3.2). Furthermore, we prove that SPARTAN representations lower-bound the Euclidean distance in the original space, preventing false dismissals in similarity search (Section 3.3).

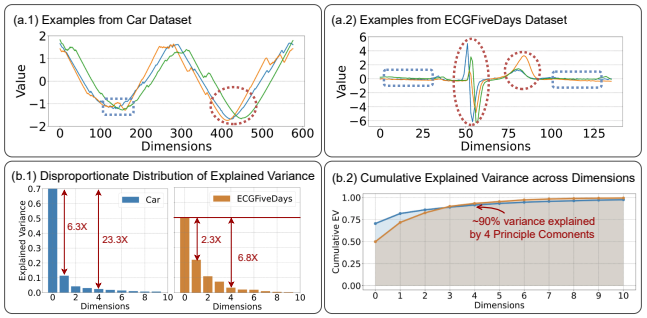

Figure 3: Measure of informativeness across dimensions on Car and ECGFiveDays datasets: (a) representative examples from each dataset, with flat regions highlighted in blue bounding boxes and representative patterns in red ellipses; and (b) explained variance for the first 10 latent dimensions.

3.1 SPARTAN's Numeric Approximation

To capture essential information, symbolic methods need to account for the imbalanced importance across dimensions. To illustrate this, we visualize representative time-series samples from two UCR datasets[13] (Figure 3). From both examples, we can observe that the importance of each dimension is highly disproportionate: (i) many dimensions are flat and less informative (highlighted in blue) compared to representative patterns (in red); and (ii) the majority of the information is concentrated to a few top dimensions indicated by explained variance. This highlights the strong need for an adaptive method that can accurately reflect this characteristic.

To quantify the imbalanced importance, we start by measuring the variance of the original dimensions, a practice that has proven effective in prior studies [21, 56]. As the value of variance indicates the degree of variability within the data, a higher variance in one dimension, suggests a greater likelihood of capturing important features. Given a time series dataset $\mathcal{X} = [X_1, X_2, \cdots, X_n] \in \mathbb{R}^{n \times m}$, we formulate the variance of each dimension as:

$$Var_i(\mathcal{X}) = \frac{1}{n}\sum_{j=1}^{n}(X_j^i - \mu_i)^2, \tag{1}$$

where X_j^i is the value of ith dimension for jth time series X_j and μ_i denotes the mean of ith data dimension. To efficiently measure the variance and address the correlation between dimensions, we leverage the intrinsic property of PCA, an optimal linear dimensionality reduction method [33]. PCA performs the eigen-decomposition over the covariance matrix $cov(\mathcal{X}) = \mathcal{X}^\top\mathcal{X} \in \mathbb{R}^{m \times m}$, whose solution is known to be an orthogonal projection of the data with maximal variance. Specifically, the covariance matrix can be decomposed

into $cov(\mathcal{X}) = \mathcal{W}\Lambda\mathcal{W}^\top$. Here $\Lambda = diag(\lambda_1, \ldots, \lambda_m) \in \mathbb{R}^{m\times m}$ is the diagonal matrix of eigenvalues (explained variance) of $cov(\mathcal{X})$ sorted by descending order, which can serve as a proxy for the importance of each dimension. $\mathcal{W} = [\overrightarrow{w}_1, \overrightarrow{w}_2, \ldots, \overrightarrow{w}_m] \in \mathbb{R}^{m\times m}$ represents the orthonormal matrix of corresponding eigenvectors, serving as the bases for the non-overlapping latent space.

By applying a linear projection $X\mathcal{W}$, we extract the principal components (PCs), a numeric representation that approximates and captures the intrinsic properties of the raw time series in the latent space. As mentioned above, the latent dimensions corresponding to the top-ranked PCs carry the most information (i.e., explained variance) while the remaining tend to be flat and less informative (Figure 3). Next, we reformulate Eq. 1 by normalizing the explained variance $\overrightarrow{EV}$. Formally, given the word length ω, we select the top ω PCs for numeric approximation, which explained the most variance:

$$EV_i(\mathcal{X}) = \frac{|\lambda_i|}{\sum_{j=1}^{\omega} |\lambda_j|}, i \in [1, \omega]. \tag{2}$$

Accordingly, we can transform the original time series X into a low-dimensional representation $\hat{X} = \psi(X) = X\mathcal{W}_\omega$, where $\hat{X} \in \mathbb{R}^{n\times\omega}$ denotes the top PCs and $\mathcal{W}_\omega$ consist of the first ω eigenvectors from $\mathcal{W}$ correspondingly. In the following discretization process, each dimension of $\hat{X}$ will be transformed into one symbol by a learned look-up table. Notably, compared with data-agnostic methods such as PAA or DFT, the numeric approximation step of SPARTAN is *data-adaptive*, where the latent dimensions are constructed with full knowledge of the dataset, capturing intrinsic data properties. In the next, we introduce SPARTAN's dynamic discretization strategy.

3.2 SPARTAN's Discretization

Having derived the latent dimensions for numeric approximation, the next key problem is the discretization, which transforms the numeric representation to a sequence of symbols $\widetilde{X} = \phi(\hat{X})$, $\widetilde{X} \in \Phi^\omega$, where $\Phi = \{s_1, s_2, \cdots, s_\alpha\}$ denotes the vocabulary set of symbols given alphabet size α. In this section, we first introduce the necessity of non-uniform policy under the same budget and then propose *Dynamic Alphabet Allocation*, an adaptive strategy to efficiently distribute the budget guided by the importance of each dimension.

Non-uniform policy. Thus far in the literature, it has been assumed a uniform policy – every dimension is assigned the same encoding budget (i.e., the number of bits to represent a symbol) for their equal importance. However, as previously mentioned, this uniform policy may potentially waste valuable budget on less informative dimensions, resulting in inefficient use of the budget. To address this limitation, our goal is to develop a more intelligent allocation strategy that optimizes the use of the same overall budget, by prioritizing more important dimensions with more budgets. As demonstrated in the experiment studies (Section 5.2), naive allocation strategies do not necessarily enhance performance and may even underperform the uniform policy. Next, we propose *Dynamic Alphabet Allocation* (DAA), a non-uniform policy that dynamically allocates the budget based on the importance of each dimension.

Dynamic Alphabet Allocation. When constructing SPARTAN representations, we abandon the uniform policy and allow vocabularies Φ with an unequal number of symbols. To achieve this, we introduce the *bit-budget*, a storage budget constraint (in bits) determined by alphabet size α and ω symbols. An *alphabet allocation*

Algorithm 1: Dynamic Alphabet Allocation

Input : Var is the truncated explained variance
Constraint is a tuple of parameters
Output: $BestVal$ is the best value of the objective function
$BitPerSym$ is the optimized bit assignments

Function DynAlphaAlloc(Var,$Constraint$):
  $Budget, WordLen, \lambda = Constraint$
  α = int($Budget/WordLen$)
  $MaxBit$ = int(max(Var)$*Budget$)
  DP = createArray(($WordLen$+1, $Budget$+1))
  $Alloc$ = createArray(($WordLen$+1, $Budget$+1))
  $BitPerSym$ = createArray(($WordLen$, 1))
  /* Initialize DP with value of $-Inf$ */
  /* Initialize $Alloc$ with value of $Budget$ */
  $DP[0][0] = 0$
  for $i \leftarrow 1$ *to* $WordLen$ **do**
    for $j \leftarrow 1$ *to* $Budget$ **do**
      for $x \leftarrow 1$ *to* $MaxBit$ **do**
        if $j - x \geq 0$ & $x \leq Alloc[i-1][j-x]$ **then**
          $ev = Var[i-1]$
          $Reward = x * ev - \lambda * ev * (x-\alpha)^2$
          if $DP[i-1][j-x] + Reward > DP[i][j]$ **then**
            $Alloc[i][j] = x$
            $DP[i][j] = DP[i-1][j-x] + Reward$
  $BestVal = DP[WordLen][Budget]$
  $BitPerSym = traceBack(Alloc)$
  return $BestVal, BitPerSym$

$\vec{a} = [a_1, a_2, \cdots, a_\omega] \in \mathbb{N}_+^\omega$ is a vector of length ω whose element-wise sum is equal to the bit-budget. Specifically, the ith symbol is drawn from a vocabulary Φ with alphabet size 2^{a_i}. The bit-budget is ideal for a standard comparison between two representations: any allocation with bit-budget k has 2^k possible unique symbol combinations for representation and can be stored in k bits.

We aim to find an alphabet allocation $\vec{a}$ which maximizes the total reward, $\mathcal{R}(\vec{a}, \vec{w}) = \vec{w}^\top\vec{a}$, where $\vec{w} = [w_1, w_2, \cdots, w_\omega]$ represents the predefined weights for each symbol. We could faithfully assign the weights as the explained variance $\overrightarrow{EV}$, a proxy for the importance of each dimension. However, as demonstrated in the experimental studies (Section 5.2), we observe that such naive strategy (denoted as `naiveDAA`) can lead to significant performance degradation, often leading to a trivial solution where the majority of bits are allocated to the first few dimensions, leaving the rest with 0s.

To avoid trivial solutions and encourage a more balanced allocation, we introduce a regularization term with a parameter $\lambda > 0$, controlling the "smoothness" of alphabet allocation: it imposes a greater penalty when one symbol consumes an excessive portion of bit-budget. A large λ would encourage more evenly distributed allocations, e.g., $\vec{a} = [2, 2, 2, 2]$, while a small λ would allow more flexibility for a skewed distribution, e.g., $\vec{a} = [3, 3, 1, 1]$. As we demonstrate in Section 5.2, λ is robust to a wide range of values, and one single value such as $\lambda = 0.5$ generally works well across

Algorithm 2: SPARTAN Symbolic Representation

Input : X is a $n \times m$ matrix of preprocessed data
$Budget$ is the # of bit-budget for symbols
$WordLen$ is the total # of symbols for encoding
λ is the parameter to balance the allocation
Output: $Word$ is the symbolic representation output

Function SPARTAN(X,$Budget$,$WordLen$, λ):
  /* TRAINING STAGE */
  $[EigVal, EigVec]$ = solveEIG($X^T * X$)
  $[EigVal, IndexSorted]$ = sortDescending($EigVal$)
  $EigVec = EigVec[IndexSorted]$
  $EigValTruncated = EigVal[:WordLen]$
  $EigVecTruncated = EigVec[:, :WordLen]$
  $Var = EigValTruncated$ / sum($EigValTruncated$)
  $Constraint$ = tuple($Budget$, $WordLen$, λ)
  $BestVal$, $BitPerSym$ = DynAlphaAlloc(Var,$Constraint$)
  $\hat{X} = X * EigVecTruncated$
  $Breakpoints$ = Binning($\hat{X}$, $BitPerSym$)
  $Word$ = Discretize($\hat{X}$, $Breakpoints$)
  /* INFERENCE STAGE */
  $\hat{X} = X * EigVecTruncated$
  $Word$ = Mapping($\hat{X}$, $Breakpoints$)
  return $Word$

various tasks. Given the bit-budget K of ω symbols, we formulate the constrained optimization problem as follows:

$$\vec{a}^* = \underset{\vec{a}}{\text{argmax}} \sum_{i=1}^{\omega} \left(\mathcal{R}(a_i, w_i) - \lambda \cdot w_i \cdot (a_i - \frac{K}{\omega})^2 \right) \tag{3}$$
$$\text{s.t. } a_1 \geq a_2 \geq \cdots \geq a_\omega, \|\vec{a}\|_1 = K, \vec{a} \in \mathbb{N}_+^\omega.$$

This optimization problem can be efficiently solved by dynamic programming-based solutions, as presented in Algorithm 1. Given a word of ω symbols and a bit-budget of K, we define the optimal reward δ, which is computed recursively at each step of ith symbol and j bit-budget, considering all valid allocations:

$$\delta(i, j) = \max_{1 \leq x \leq j, MaxBit} \{\delta(i-1, j-x) + \mathcal{R}'(x, ev_i, \lambda, K, \omega)\}, \text{with}$$

$$\mathcal{R}' = \begin{cases} x \cdot ev_i - \lambda \cdot ev_i \cdot (x - \frac{K}{\omega})^2, & x \leq PrevAlloc, \\ -\infty, & \text{otherwise} \end{cases} \tag{4}$$

Here, $\delta(i, j)$ represents the optimal reward for i symbols with j bit-budgets. $PrevAlloc$ denotes the bit allocation for the previous symbol, typically stored in a 2-D array $Alloc$. The maximum allowable bits for each symbol, $MaxBit$, is naturally bounded by $\max(ev) \cdot K$. The recursion for the reward δ is initialized with $-\inf$, except for $\delta(0, 0) = 0$. By solving for the optimal reward at each step using dynamic programming, the ultimate reward can be obtained from the final step $\delta(\omega, K)$. The corresponding bit allocation can then be traced back from $Alloc$. With appropriate regularization in the reward design, this dynamic strategy achieves a balanced distribution that effectively reflects the importance of each dimension.

We note that this alphabet allocation process constitutes only a minimal portion of the total time (will be shown in Section 5.6), given that $\alpha, \omega \ll m$. After dynamically allocating the alphabet size across dimensions, we construct a histogram for mapping the symbols (binning) and store the breakpoints in a look-up table. In Section 5.2, we present additional experiment results comparing various binning strategies [65]. Notably, SPARTAN is flexible and can accommodate alternative binning strategies as needed.

The overall pipeline of SPARTAN is outlined in Algorithm 2, with an illustrative example provided in Figure 2. The key procedures are summarized as follows: (i) during the training stage, SPARTAN first derives the non-overlapping and uncorrelated latent dimensions, and measures the importance across dimensions. It then performs dynamic discretization, adaptively assigning the bit budget based on the explained variance of dimensions, followed by constructing the look-up table for mapping the approximation values to symbols (using binning strategies); and (ii) in the inference stage, SPARTAN can efficiently approximate the queried data using linear projection to the latent dimensions (using eigenvector basis) and discretize the approximated data into symbols by the look-up table learned in the training stage. In the following section, we are going to introduce a desirable lower-bounding property and provide a formal proof.

3.3 SPARTAN's Lower Bounding Property

Apart from the representation capability, it is also ideal to build a strong connection between the symbolic space and the original space. SAX first formally shows to lower bound the Euclidean distance (ED) on the original space, preventing false dismissals when searching the data [18, 73]. The closer the distance measured in symbolic space is to the original space, the greater the pruning power demonstrated by the method. Following prior works, we quantify this pruning power by the tightness of lower bound (TLB) and adopt this as a proxy for the performance in indexing tasks. TLB is the ratio of the measured dissimilarity between the symbolic space and the Euclidean space, as shown in Equation. 5:

$$TLB = \frac{Symbolic\ Distance(\tilde{Q}, \tilde{C})}{Euclidean\ Distance(Q, C)}. \tag{5}$$

A symbolic method that preserves this lower bounding property should retain $TLB <= 1$. We are going to demonstrate the lower bounding property of SPARTAN in two steps: (i) demonstrate the distance between the numeric approximation in the latent space lower-bounds the Euclidean distance on raw data; and (ii) prove that final symbolic representation lower-bounds the distance between the numeric approximation. By transitivity, this establishes that the distance between SPARTAN representations also lower-bounds the Euclidean distance. The first step relies on the fact that PCA projection is an orthogonal transformation that naturally preserves Euclidean distances [32], meaning for time series Q, C, the distance between all PCs (numeric representation) is identical to that in the original space $d(Q, C) = d(Q\mathcal{W}, C\mathcal{W})$. Subsequently, we could derive the lower bounding property between numeric approximation and Euclidean distance (on raw time series).

Lemma 1. *Given two time series Q and C, the numeric approximation $\hat{Q} = [\hat{Q}_1, \hat{Q}_2, \cdots, \hat{Q}_\omega]$ and $\hat{C} = [\hat{C}_1, \hat{C}_2, \cdots, \hat{C}_\omega]$ from the top ω PCs lower bounds the Euclidean distance between Q and C.*

Proof. We want to prove that $d(\hat{Q},\hat{C}) = d(Q\mathcal{W}_\omega, C\mathcal{W}_\omega) \leq d(Q,C)$. The right-hand side can be derived as:

$$\begin{aligned} d(Q,C) &= d(Q\mathcal{W}, C\mathcal{W}) \\ &= \sqrt{\sum_{i=1}^{\omega}(Q\mathcal{W}_i - C\mathcal{W}_i)^2 + \sum_{j=\omega+1}^{m}(Q\mathcal{W}_j - C\mathcal{W}_j)^2} \\ &\geq \sqrt{\sum_{i=1}^{\omega}(Q\mathcal{W}_i - C\mathcal{W}_i)^2} \\ &= d(Q\mathcal{W}_\omega, C\mathcal{W}_\omega) \end{aligned}$$ □

Therefore, we show that approximation by the first ω PCA components guarantees a lower bound of the Euclidean distance. Next, a slight modification of the *MINDIST* function [42, 43] is proposed for the proof of SPARTAN representations. Formally, we obtain the symbolic distance for two SPARTAN representations $\tilde{Q}, \tilde{C}$:

$$\text{PCA-MINDIST}(\tilde{Q},\tilde{C}) = \sqrt{\sum_{j=1}^{\omega}(dist(Ind(\tilde{Q}_j), Ind(\tilde{C}_j), j)^2}, \quad (6)$$

$$dist(q,c,j) = \begin{cases} 0, & if|q-c| \leq 1 \\ \beta_{j,max(q,c)-1} - \beta_{j,min(q,c)}, & otherwise \end{cases} \quad (7)$$

, where $\beta_{j,i}$ is the ith breakpoint from the jth symbol of a SPARTAN word and $Ind(\tilde{Q}_j)$ denotes index of the output symbol $\tilde{Q}_j$ from the lookup table. Next, we show that PCA-MINDIST$(\tilde{Q},\tilde{C}) \leq d(\hat{Q},\hat{C})$.

Lemma 2. *Given two time series Q and C, the symbolic representaion $\tilde{Q} = [\tilde{Q}_1, \tilde{Q}_2, \cdots, \tilde{Q}_\omega]$ and $\tilde{C} = [\tilde{C}_1, \tilde{C}_2, \cdots, \tilde{C}_\omega]$ lower bounds the distance between the approximation $\hat{Q}$ and $\hat{C}$ under PCA-MINDIST.*

Proof. Recall the discretization process we construct the breakpoints $\beta_{j,0}, \beta_{j,1}, \ldots \beta_{j,\alpha}$ for jth symbol ($1 \leq j \leq \omega$), where $\beta_{j,0} = -\infty$ and $\beta_{j,\alpha} = +\infty$. By our definition, the jth PC for time series Q, is assigned with the symbol ϕ_i iff $\beta_{j,i-1} \leq Q\mathcal{W}_j < \beta_{j,i}$. Denoted by q_j and c_j, the indexes of the jth symbol of $\tilde{Q}$ and $\tilde{C}$, it is sufficient to show the inequality holds for each symbol: $\forall 1 \leq j \leq \omega, d(Q\mathcal{W}_j, C\mathcal{W}_j) \geq dist(q_j, c_j, j)$.

Case 1: $|q_j - c_j| \leq 1$. In this case, when two symbols are the same or adjacent to each other, $dist(q_j, c_j, j) = 0 \leq d(Q\mathcal{W}_j, C\mathcal{W}_j)$.

Case 2: $|q_j - c_j| > 1$. Assume $Q\mathcal{W}_j > C\mathcal{W}_j$ and $q_j > c_j$; a similar proof holds for the reverse case. By definition, we obtain $dist(q_j, c_j, j) = \beta_{j,q_j-1} - \beta_{j,c_j}$ and we know there exist two sets of breakpoints such that $\beta_{j,q_j-1} \leq Q\mathcal{W}_j < \beta_{j,q_j}$ and $\beta_{j,c_j-1} \leq C\mathcal{W}_j < \beta_{j,c_j}$. By aggregating the observations, we can derive that $Q\mathcal{W}_j \geq \beta_{j,q_j-1}$ and $C\mathcal{W}_j < \beta_{j,c_j}$, which implies $Q\mathcal{W}_j - \beta_{j,q_j-1} \geq C\mathcal{W}_j - \beta_{j,c_j}$. Rearranging the terms we get $d(Q\mathcal{W}_j, C\mathcal{W}_j) = Q\mathcal{W}_j - C\mathcal{W}_j \geq \beta_{j,q_j-1} - \beta_{j,c_j} = dist(q_j, c_j, j)$. □

Therefore we retain the lower bounding property for the SPARTAN representation. In Section 5.3, we conduct a comprehensive evaluation of TLB and demonstrate the superior performance of SPARTAN compared to the current state-of-the-art methods.

4 EXPERIMENTAL SETTINGS

In this section, we review the settings for the evaluation of (i) the representation ability on time series downstream tasks, e.g., classification, clustering, and anomaly detection; (ii) the pruning power, measured by the tightness of lower bound (TLB); (iii) the scalability and runtime analysis on the large-scale datasets.

Datasets: We utilize one of the largest collections of labeled univariate time-series, the UCR time-series archive [13], for classification, clustering tasks and TLB analysis. UCR consists of 128 datasets, including both synthetic and real-world data across diverse domains. Each dataset contains between 40 and 24000 samples, with sequence lengths ranging from 15 to 3,000. We maintain the standard train-test splits of the datasets. In the case of datasets with missing values or unequal lengths, we apply a preprocessing step to forward-fill missing values and do resampling for unequal-length series. We perform z-score normalization for all datasets. For anomaly detection, we utilize the TSB-UAD archive [59], one of the largest collections of univariate time-series anomaly data, which includes 18 anomaly datasets of about 2000 univariate time series spanning different domains with high variability of anomaly types. To validate the generalization capabilities of different methods across various data domains, we perform comprehensive statistical tests to validate our findings, as detailed in the following paragraphs.

Platform and Implementation: Experiments are conducted using a cluster equipped with AMD EPYC 7713 64-Core Processors using the Rocky Linux 8.5 (Green Obsidian) operating system. We implement and run SPARTAN and all baseline methods in Python 3.8 for a fair and consistent comparison. The main dependencies are listed as follows: numpy 1.24.4, pandas 2.0.3, scikit-learn 1.3.2, scipy 1.10.1, tsfresh 0.20.2, tslearn 0.6.3. To ensure the reproducibility of our results and findings, we make the code available [1].

Baselines: We compare SPARTAN against the following state-of-the-art symbolic representation methods for time-series analysis: (i) **SAX**, a simple yet efficient symbolic representation by applying PAA and Gaussian assumption for discretization [42]; (ii) **ESAX**, an extension of SAX that incorporates two additional extreme points for approximation [46]; (iii) **1d-SAX**, an extension of SAX which includes the approximate slope value [49]; (iv) **TFSAX**, a SAX variant method with trend feature [79]; (v) **SAX-DR**, a SAX variant that includes a direction representation [39], (vi) **SAX_VFD**, an extension of SAX method which samples time-series-related features for representation [76]; (vii) **SFA**, a symbolic method based on Discrete Fourier Transform for approximation and MCB for discretization [65]. Recent methods ABBA [17] and fABBA [12] reduce reconstruction error using adaptive polygonal chain approximation, yet they generate variable-sized symbolic representations for each time series, making them less suitable for standardized benchmarking. We exclude them from this evaluation. In this study, we also include Euclidean distance on raw time series as a valuable baseline for comparison with the current best symbolic solutions, which will be illustrated in Section 5.8.

Representation Type: We report results on two types of symbolic representations, named (i) *Single Pattern* (SP), which transforms each time series into a single word, e.g., "AACBADCC;" (ii) *Bag-Of-Patterns* (BOP), which applies a sliding window and symbolizes the data in a subsequence manner, i.e., instead of a single word, BOP produces a list of symbolic words such as "[AABC, ACBA, $\cdots$, BCAD]", from which a histogram can be generated for subsequent analysis [44, 45, 64]. This is equivalent to the "bag of words" concept in the Information Retrieval tasks [71]. While the single pattern works in a whole time-series manner, BOP uncovers the distribution of more fine-grained (local) features by searching for repetitive subsequences. By considering both types of representation in our

evaluation study, we gain a better understanding of the model's representation ability at both global and local levels.

Symbolic Distance Measure: To capture dissimilarities between pairs of symbolic representations, methods often introduce their own symbolic distance measures. However, these existing measures present several key challenges, which complicate a fair evaluation: (i) they are often tailored to specialized feature designs and predefined lookup tables; and (ii) not all methods provide the same level of detail in their distance measures, with some lacking a dedicated symbolic measure. We observe that currently no single distance measure can universally accommodate all symbolic representations.

Given the absence of a unified testbed, we propose a generic symbolic distance that is straightforward and ensures fairness, namely the *Symbolic L_1 Distance*. We define this distance as follows: given two time series Q, C and their symbolic representation $\widetilde{Q}, \widetilde{C}$, the symbolic distance is calculated by the sum of ordinal differences between symbols: $\mathcal{D}(\widetilde{Q}, \widetilde{C}) = \sum_{i=1}^{\omega} |Ind(\widetilde{Q}_i) - Ind(\widetilde{C}_i)|$, where $Ind(\widetilde{Q}_i) \in \{1, 2, \ldots, \alpha\}$ denotes the index of each symbol with alphabet size α. Symblic L1 solely relies on the assumption that the alphabet of methods can be meaningfully ordered – if $|Ind(\widetilde{Q}_i) - Ind(\widetilde{Q}_j)| < |Ind(\widetilde{Q}_i) - Ind(\widetilde{Q}_k)|$, then $\mathcal{D}(\widetilde{Q}_i, \widetilde{Q}_j) < \mathcal{D}(\widetilde{Q}_i, \widetilde{Q}_k)$ holds. This design is compatible with all baselines, and can accommodate potential new methods in the future. Since it requires no prior knowledge of given methods, this distance is unlikely to unfairly favor one representation over another. In Section 5.7, we demonstrate Symbolic L_1 sets a solid foundation for comparing the representation power.

To measure the dissimilarity between BOP histograms, we test four widely-used distance measures including Euclidean distance (ED), Cosine similarity, KL-Divergence [38] and BOSS distance [64]. The results indicate no significant difference between these measures. Therefore, we adopt ED as the default for this context.

Statistical Test: To statistically validate the significance of performance improvement between methods, we follow [9, 15, 56, 58] and utilize the Friedman test [19] followed by the post-hoc Nemenyi test [53] with 95% confidence level, which is a well-recognized strategy for comparison of multiple algorithms across multiple datasets.

Evaluation Framework: We compare our approach on four analytical tasks: (i) for classification, we adopt the one-nearest-neighbor (1NN) classifier, evaluated by accuracy; (ii) for clustering, we utilize partition around medoids (PAM) for SP and KMeans for BOP respectively, evaluated by Rand Index (RI) [62];To account for randomness, we report the average RI results over 10 runs. (iii) for indexing, we report the tightness of lower bound (TLB) as a proxy; (iv) for anomaly detection, we apply k-nearest neighbors (KNN) detector on BOP representations and evaluate performance on two well-established evaluation metrics, VUS-PR and VUS-ROC [55], to validate their ability to distinguish anomalies. Finally, to evaluate the time efficiency, we compute the cumulative CPU runtime, and perform scaling experiments on the synthetic dataset.

5 EXPERIMENTAL RESULTS

In this section, we evaluate the performance of SPARTAN against baselines on downstream tasks as a proxy to assess their quality: (i) classification (Section 5.1 and 5.2); (ii) tightness of lower bound (Section 5.3); (iii) clustering (Section 5.4); (iv) anomaly detection (Section 5.5); (v) accuracy-to-runtime analysis (Section 5.6). We also

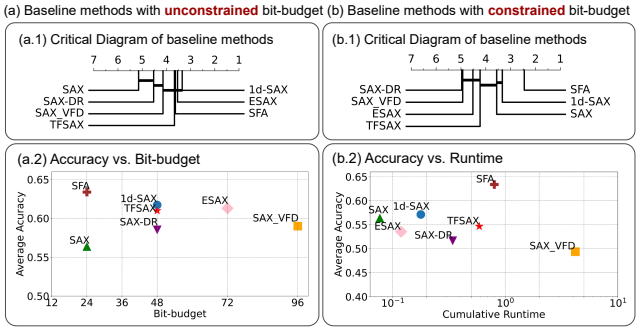

Figure 4: Evaluation of 7 baseline methods on 1NN classification accuracy, bit-budget, and runtime with (a) unconstrained bit-budget and (b) constrained bit-budget. The solid line in the CD plots connects all methods that do not perform statistically differently according to the Nemenyi test.

provide additional experiment results on symbolic distance measures (Section 5.7) and comparison with true distance (Section 5.8). We will summarize the findings in Section 5.9.

5.1 Evaluation of Baseline Methods

As mentioned in Section 3.2, for a fair comparison, it is essential to establish consistent rules and evaluate each method under the same parameter settings. In many previous studies, the number of segments ω_s is typically considered the standard constraint for comparison, while allowing flexibility in the word length ω. For example, SAX utilizes only the mean value for each segment ($\omega = \omega_s$), whereas ESAX, a representative variant of SAX, incorporates two additional features for each segment and improves performance with longer words ($\omega = 3\omega_s$) [46]. It is claimed that the difference between storage costs is considered marginal compared with the original time series length m, making it reasonable to constrain only the number of segments ω_s [46]. However, we argue that this experiment setting may lead to ambiguity in the evaluation: *it's hard to determine whether the performance improvement stems from the enhanced representation quality, or from sacrificing a larger encoding budget, e.g., a longer word of symbols.*

To address this concern and establish a reliable benchmark, we explore two experimental settings to evaluate the representation ability with different constraints (under the same alphabet size α):

(1) comparison with the same amount of segments ω_s;
(2) comparison with the same amount of symbols ω.

In the first experiment, we follow the practice in previous studies [39, 46] and allow flexibility in the word length ω for additional features (denoted as "unconstrained bit-budget"). For the second experiment, a simple yet crucial criterion can be derived for fair evaluation: *under the same encoding budget, given word length ω and alphabet size α, how does performance vary across different methods?* Ideally, a robust variant method would be able to demonstrate its representation power by outperforming SAX under both settings.

We perform a comprehensive evaluation study on **1-NN classification**, a parameter-free downstream task, to evaluate the symbolic representation power of each method. A generic distance measure, Symbolic L_1, is adopted to help facilitate a fair comparison on the SP representation. In Section 5.7, we demonstrate the effectiveness of Symbolic L_1 compared to existing distance measures in the ablation

study. We present a global comparison using the Critical Diagram (CD), which provides the average rank across all datasets (position on the horizontal line) with statistical tests (solid line connects all methods with no significant performance differences). Figure 4(a) indicates that, all variant methods can outperform SAX with a higher ranking by sacrificing more bit-budgets, which aligns with our expectations. However, when it comes to the second experiment setting (Figure 4(b)), the results provide a clearer perspective: most SAX variants that introduce more features could not surpass the basic SAX under the same bit-budget constraint. This is a surprising finding, still, it can be reasonably explained. In this scenario, the performance of a symbolic method is determined by its intrinsic representation power rather than the number of selected features. Experimental results show that, despite the elaborate design, many combinations of features do not show significant improvement over SAX under the same bit-budget, which debunks the long-standing misconceptions. This highlights the necessity of this simple yet crucial standard to ensure a fair comparison.

Following the new standard, we observe that 1d-SAX is the only variant that outperforms SAX, though statistically not significant. Among all baselines, SFA, is the only method that strongly outperforms SAX under the same budget, showing its robust representation quality. In the following sections, we compare SPARTAN with the top methods and validate the findings across various downstream tasks. As no SAX variants outperform SAX significantly, we use SAX as the representative baseline along with SFA.

5.2 Evaluation on Classification

In this section, we compare our proposed SPARTAN method with the top two methods, SAX and SFA, on the classification task. As discussed above, the constraint on the bit-budget serves as a simple yet effective criterion for fair comparison. We will maintain this standard in the following downstream tasks. We start from the single pattern (SP) representation with the parameter settings of $\alpha = 4, \omega = 8$ (will demonstrate the robustness across varying parameters in the subsequent analysis). As shown in the critical diagrams (CDs) (Figure 5(a.1)), we observe that SPARTAN strongly outperforms the top two baselines on the SP representation, exhibiting statistically significant differences from both SAX and SFA. Specifically, pairwise comparisons (Figure 5(b.1)) reveal a clear trend of performance improvement over about 2/3 UCR datasets (we highlight the upper left triangle region in green, where SPARTAN outperforms its rival). The superior performance demonstrates the effectiveness and accuracy of SPARTAN representation.

To better understand the performance of SPARTAN under different representations, we also incorporate the BOP in our analysis. Without losing generality, we compare methods under $\alpha = 4, \omega = 4, w = 5\%m$ for BOP. In this representation setting, we take advantage of the BOP strategy and compute the Euclidean distance between histograms of word occurrences generated by sliding windows, which focuses more on local features. Numerosity reduction technique [64] is adopted to avoid overweighting stable sections of the time series. Figure 5(a.2) and (b.2) show that SPARTAN consistently outperforms SAX and SFA, with a clear performance gap. We attribute this consistent improvement to the data-adaptive nature of SPARTAN: the non-uniform subspaces found by SPARTAN prioritize more informative dimensions in the latent space. In contrast,

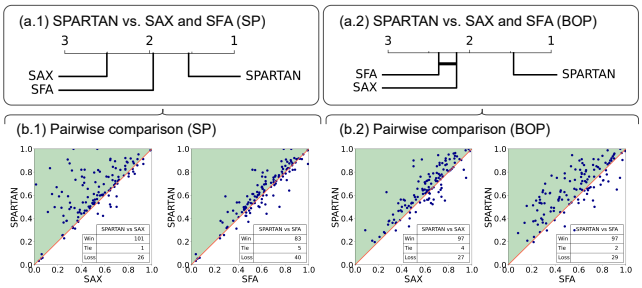

Figure 5: 1NN Classification evaluation of SPARTAN, SAX and SFA, including (a) critical diagrams (CD) with statistical test; (b) pair-wise comparisons on all 128 UCR datasets.

mean values or the first few low-frequency components from DFT may fail to effectively model the underlying distribution and may struggle to handle complex scenarios. Notably, all representations are constructed in an unsupervised manner across downstream tasks, as our goal is to establish a unified benchmark for fair evaluation across all symbolic methods. In real-world applications, this limitation can be mitigated through expert guidance or supervised techniques. For instance, SFA+ANOVA [67] has shown significant success in dictionary-based classifiers with elaborate design. Additionally, automated machine learning (AutoML) solutions [27, 34] present a promising direction for the adaptive solution (e.g., selecting important frequencies) in diverse domains.

Having demonstrated strong classification performance under standard settings, we further evaluate the robustness of our proposed method across different parameter settings. As shown in Figure 12(a), we observe that SPARTAN maintains superior performance over surveyed parameters and displays a scaling ability under increasing alphabet sizes and word lengths. This can be attributed to its non-uniform policy, which optimally distributes the encoding budget to the most informative dimensions. As a result, SPARTAN effectively mitigates the impact of noise and captures the important patterns compared to baseline approaches.

To assess the effectiveness of each component, we perform an ablation study on SPARTAN, including alphabet allocation strategy, regularization parameter λ, and the binning strategy as follows.

Alphabet Allocation: We compare the proposed Dynamic Alphabet Allocation (DAA) against three alternative strategies: (i) `SPARTAN_woDAA`, which removes the DAA module from SPARTAN and apply a uniform policy; (ii) `SPARTAN_naiveDAA`, which allocates the alphabet proportionally to explained variance (permits allocation of 0 bits); and (iii) `SPARTAN_naiveC(onstrainedDAA)`, which still allocates the alphabet proportionally but with at least 1 bit for each dimension. As depicted in Figure 6(a), SPARTAN (with DAA) consistently outperforms other simple strategies in both SP and BOP representations. Notably, naive strategies (e.g., `naiveDAA`, `naiveC`) underperform `SPARTAN_woDAA`, which applies no dynamic allocation. Specifically, `naiveDAA` often results in trivial solutions by concentrating resources on the first few dimensions while leaving the others with 0s. Although `naiveC` improves performance by explicitly introducing constraints (Figure 6(a.2)), it still produces highly imbalanced allocations and performs worse than `SPARTAN_woDAA`. These findings highlight the necessity of a well-designed algorithm for effective alphabet allocations.

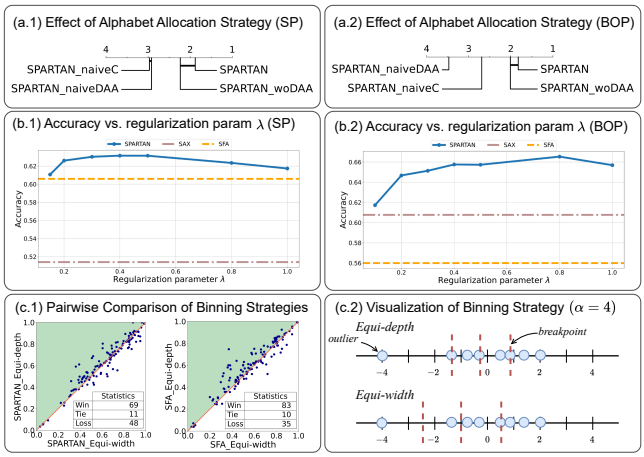

Figure 6: Ablation study of SPARTAN on 1NN Classification task. (a) Critical diagrams of SPARTAN with different alphabet allocation stategies; (b) Classification accuracy with varying regularization parameter λ; and (c) Comparison of two binning strategies, Equi-depth and Equi-width.

Regularization: We also evaluate the performance of SPARTAN with respect to the regularization parameter λ. Experiment results indicate that our method is robust to a wide range of parameter selections in both SP and BOP representation types. It is observed that SP tends to perform better with smaller λ values (light regularization), while BOP benefits from larger λ values (heavy regularization). We recommend $\lambda = 0.5$ in general case. It is noteworthy that, with sufficient time and computing resources, the parameter could also be fine-tuned with supervision to achieve optimal performance.

Binning Strategy: In this study, we test two unsupervised binning strategies [65]: (i) Equi-depth, which ensures an equal number of samples in each interval; and (ii) Equi-width, which assigns equal widths to all intervals. Experimental results indicate that Equi-depth generally outperforms Equi-width for both SPARTAN and SFA, likely due to Equi-width's sensitivity to distortion from outliers (we provide an illustrative example in Figure 6(c.2)). However, in out-of-distribution scenarios, Equi-depth may overfit on the training set, while Equi-width can provide more reliable binning, explaining cases where Equi-width performs better. Notably, SPARTAN is flexible and supports alternative binning strategies. For subsequent experiments, we use Equi-depth as the default.

5.3 Evaluation on Tightness of Lower Bound

One of the advantages of SAX is that, the proposed *MINDIST* measure has been proven to lower bound the Euclidean distance (ED) in the original space. This property, highly desirable in the indexing and similarity search, ensures no false rejections of potentially similar data objects [18]. As mentioned in Section 3.3, we quantify this property using the TLB [42, 43], which serves as a proxy for the indexing task. The value of TLB closer to 1 indicates better preservation of information after dimension reduction. Thus, it shows that the smaller the gap between the symbolic and true distance, the greater the pruning power demonstrated by the methods.

However, the majority of methods either lack proof of the lower bounding property or have exhibited violations (TLB>1) in our experiments. We compare SPARTAN with SAX and SFA, the only two

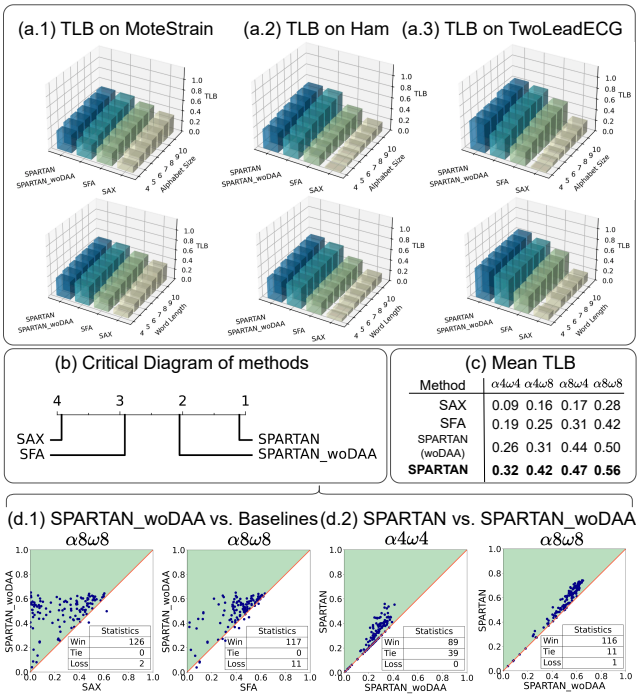

Figure 7: Evaluation study on TLB, including (a) three representative datasets with varying parameters; (b) critical diagram of SPARTAN and baseline methods; (c) mean TLB value across all 128 UCR datasets under varying parameters; (d) pairwise comparison of SPARTAN_woDAA vs. baselines, and SPARTAN vs. SPARTAN_woDAA.

baseline methods that are proven to preserve this lower bounding property. To simulate real-world applications where query series are unknown during training and to assess robustness to out-of-distribution scenarios, we adopt the train/test split from the UCR archive [13]. Specifically, all methods are trained exclusively on the training data, with TLB value computed between pair-wise symbolic representations over the test split across 128 datasets. Figure 7(a) visualizes three representative datasets across varying parameters. We observe that SPARTAN consistently outperforms SFA and SAX on these three datasets, across varying alphabet sizes and word lengths. A clearer distinction can be seen across all 128 datasets, where SPARTAN consistently takes the lead, with an approximately 15% TLB improvement over the next best baseline on average (Figure 7(b-c)). It is also shown in Figure 7 (b-d) that SPARTAN not only shows a tighter lower bound, but also exhibits robustness across varying datasets and parameter settings. Furthermore, we also observe that, SPARTAN can strongly outperform its competitors without the DAA strategy (`SPARTAN_woDAA`), owing to the optimality of our approximation method (Figure 7(d.1)). By dynamically allocating the budget (DAA), SPARTAN consistently gains a tighter lower bound (Figure 7(d.2)) across 2/3 datasets. In summary, this TLB performance ensures that, even with limited budget resources, SPARTAN consistently maintains a high-quality representation with strong pruning power.

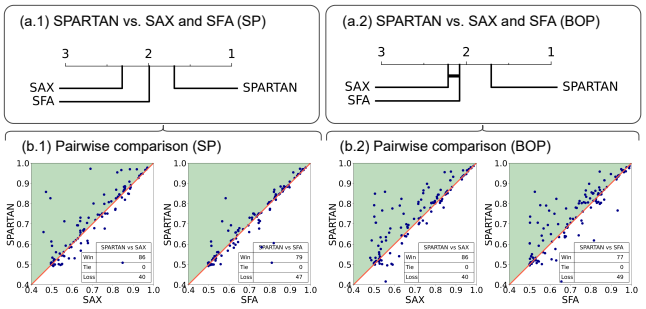

Figure 8: Evaluation study on the clustering downstream task, including (a) CD plots of SPARTAN and top two baselines; (b) pairwise comparison on UCR datasets.

5.4 Evaluation on Clustering

To better understand the representation ability in the unsupervised task, we extend our analysis to the clustering problem, a challenging downstream task that enables us to fairly assess the representation quality in the absence of labeled data [43]. Following prior works, we adopt the UCR dataset [13] and utilize the same class label for clustering evaluation. [1] We acknowledge the unclear cut issue of train/test in the clustering domain as discussed in [28], since clustering is often used more as an exploratory tool rather than a predictive model in nature. While further exploration is warranted in this area, we adhere to the pipeline outlined in prior studies [58], and report results on merged UCR train/test split for all tested methods. Following the 1NN Classification task, we also report the evaluation result on both SP and BOP representations. To properly perform clustering on SP representation, we adopt Symbolic L_1 + PAM, a widely used partitional clustering method, which searches for cluster centers within actual data samples. For BOP, each time series can be represented by a histogram of the occurrence of symbolic words, followed by the conventional KMeans clustering. All methods are evaluated by Rand Index [62].

We adopt the same parameter setting as classification and visualize the results in Figure 8. We observe that SPARTAN can strongly surpass SAX and SFA on both representation types (see Figure 8a and b). Specifically, both SFA and SPARTAN show a significant improvement in comparison to SAX with SP representation, while SPARTAN takes the lead. For BOP, where local patterns are captured through sliding windows, SPARTAN is the only method that demonstrates a significant improvement over the other two, e.g., wins about 2/3 UCR datasets (Figure 8(b.2)), while SAX and SFA show no statistical difference. These trends also align with classification findings, which demonstrate the robustness of SPARTAN.

To understand how SPARTAN performs under different settings, we compare three methods by adopting varying parameters. Without losing generality, we perform various experiments on the SP representation. Figure 12(b) displays the average Rand Index for each method across varying alphabet sizes and word lengths. We observe that SPARTAN consistently outperforms SFA and SAX under varying budget settings. Notably, all three methods improve under a larger budget, and the gap between each method is getting closer. We attribute this to the fact that, under the unsupervised setting,

[1] We exclude "Crop" and "ElectricDevices" due to the limit of computing resources.

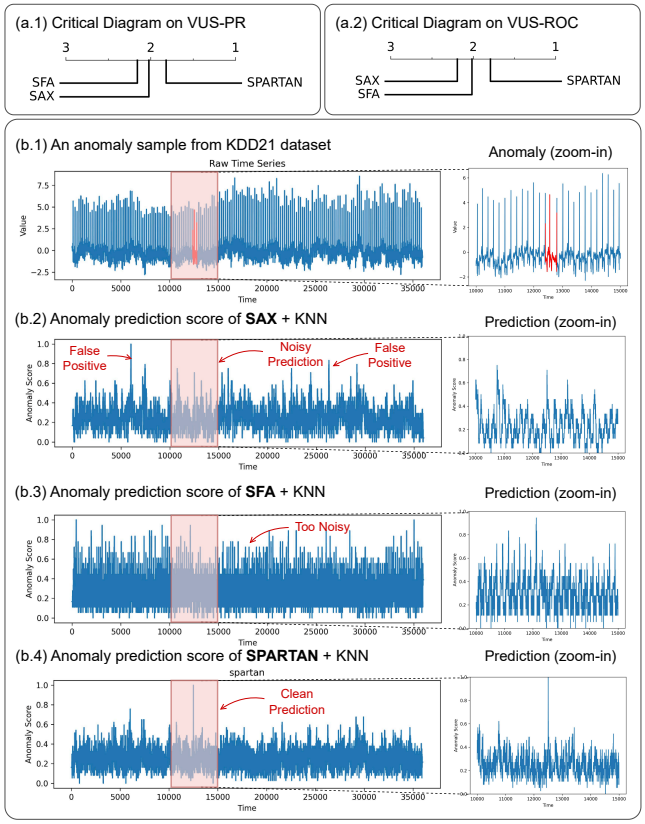

Figure 9: Anomaly detection evaluation on symbolic methods + KNN detector, including (a) CD plots evaluated by VUS-PR/VUS-ROC; (b) visualization of an anomaly example from KDD21 dataset and the prediction results of each method. The red bounding box and lines highlight the anomalies.

the performance of each method converges more quickly than in the supervised classification task. Overall, SPARTAN consistently maintains a high-quality representation across different tasks.

5.5 Evaluation on Anomaly Detection

Time-series anomaly detection (TSAD), a process of identifying abnormal time points or subsequences from the queried time series, has received increasing attention in recent years. Compared with 1NN classification and clustering, which assess the general representation ability of symbolic methods, anomaly detection downstream task challenges the methods to properly capture both normal and abnormal patterns. Enlightened by this, we add this anomaly detection evaluation study to validate our previous findings. Specifically, we employ BOP representation with the K-Nearest-Neighbor (K-NN) detector [60, 61], a general anomaly detector independent of symbolic methods, with Symbolic L_1 as default. We utilize the TSB-UAD collection [59], which includes about 2000 time series spanning different domains. All methods are evaluated by VUS-PR and VUS-ROC as suggested by [55]. Following previous studies, we conduct Friedman-Nemenyi test to validate the statistical difference.

Considering the larger search scope and the granularity of anomaly patterns in comparison to previous tasks, we utilize alphabet size $\alpha = 16$ and word length $\omega = 16$. We adopt sliding window

length $w = 100$ with top $K = 50$ nearest neighbors [59] for all methods [59]. Figure 9(a) shows the critical diagram under VUS-PR and VUS-ROC. We observe that SPARTAN consistently outperforms SFA and SAX with statistical differences under this downstream task, indicating a strong generalization ability across diverse anomaly types. To further analyze representation performance in the context of anomaly detection, we closely examine individual cases. Notably, SFA, despite being the state-of-the-art among baseline methods, exhibits performance degradation in numerous test cases (e.g., the representative example shown in Figure 9(b)), which is also supported by the VUS-PR results in Figure 9(a.1). This degradation can be attributed to SFA's focus on lower frequencies with the most energies, which may lead to the omission of anomalies that primarily occur in higher frequencies. Addressing this limitation through proper frequency selection – whether guided by expert knowledge or automated solutions – presents a promising avenue for enhancing performance in this unsupervised context. As for SAX, it highly relies on the quality of PAA (i.e., mean value of the segments), making it more sensitive to large value fluctuations but prone to generating false positives. In contrast, SPARTAN captures more essential information by prioritizing the informativeness of dimensions, which enables a more robust representation that can effectively capture both normal and abnormal patterns.

5.6 Accuracy-to-Runtime Analysis

To understand the robustness and scalability of SPARTAN, we start by analyzing the runtime of different components during the training and inference. Figure 10(a) visualize the runtime analysis for both training and testing on two large UCR datasets [13] containing thousands of samples. The results indicate that while the dynamic alphabet allocation (DAA), breakpoint creation (Binning), and mapping symbols (mapping) only contribute to a small portion, the majority of the computation cost is dominated by approximation. Though PCA is well-known for its effectiveness in dimensionality reduction, its high computational complexity may raise concerns. To address this, we introduce two optimized versions of SPARTAN: **SPARTAN-R** and **SPARTAN-S**, which alleviate the computational cost through randomized solutions and sampling strategies.

As shown in Table 1, compared with other approximation methods, FFT ($O(nm\log(m))$ and PAA ($O(nm)$), this high complexity of standard PCA ($O(nm^2)$ from singular value decomposition (SVD)) may cause concern for the efficiency of SPARTAN. We suggest that, this concern can be firstly alleviated by two factors: (i) the computational burden occurs only during training, allowing most of the cost to be handled offline beforehand. During online inference, SPARTAN merely requires a simple matrix multiplication for approximation ($O(nm\omega), \omega \ll m$) (Figure 10(a.1)); and (ii) approximate SVD solutions based on randomized solvers [25, 50] can be easily adapted to our case, since the method requires only the first few components with large explained variance, significantly reducing complexity (Figure 10(a)), making the training more comparable to inference time ($O(nm\omega), \omega \ll m$). This reduction is particularly advantageous in situations where high precision is not necessary. We denote this randomized solution as **SPARTAN-R** (Figure 10).

However, we acknowledge that the complexity's dependency on the cardinality of the symbolic representation may still raise concerns when scaled to substantially large datasets. To address this issue, we further extend our approach to **SPARTAN-S**, a sampling version with a better scaling ability on increasingly large datasets. We observe that the intensive learning process of SPARTAN can be significantly reduced by only sampling a small training subset. To better understand the robustness of SPARTAN's learning process, we evaluated the 1NN classification performance across various sampling rates on all 128 UCR datasets. Notably, we observe that SPARTAN-S can still strongly outperform the top baselines, SFA and SAX, with only 20% randomly sampled training data (Figure 10(b)), which demonstrates its robustness.

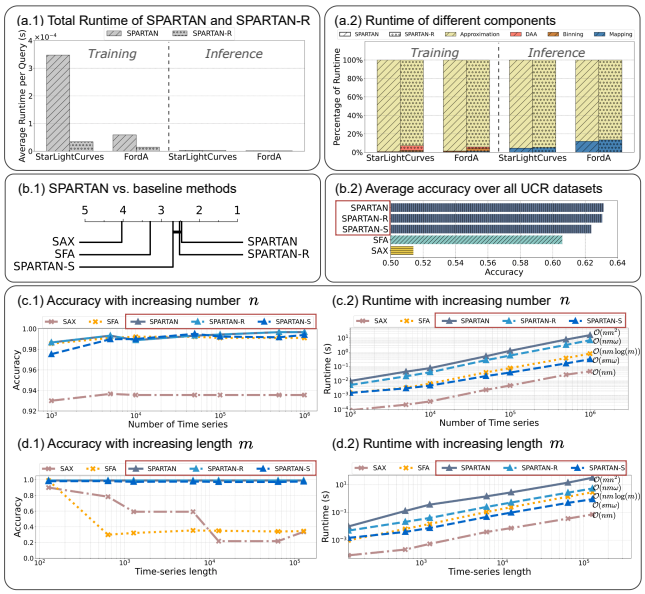

Figure 10: Accuracy-to-runtime analysis of SPARTAN against top leading methods on 1NN classification task, including (a) runtime analysis on different components of SPARTAN and SPARTAN-R; (b) comparison of SPARTAN, SPARTAN-R, and SPARTAN-S vs. SAX and SFA on all UCR datasets; (c,d) accuracy-to-runtime analysis on large-scale datasets.

To better understand the scalability of different methods, we conduct a case study on the synthetic CBF dataset [63], where a large-scale dataset can be flexibly constructed with an arbitrary number of samples. For a fair comparison, we keep $\alpha = 4, \omega = 8$ for all methods and set a sampling rate of 5% for SPARTAN-S (with a maximum of 1000 training samples). We evaluated all methods on 1NN classification under the same budget. Both accuracy and runtime results are reported on (i) varying number of time series and (ii) varying time-series length. As the number of samples n increases (Figure 10(c.1)), SPARTAN and SFA consistently take the lead. The close performance can be attributed to the fact that the generative patterns in CBF are often considered relatively easy to differentiate, explaining their similarly high performance in this setting. Meanwhile, the accelerated versions of SPARTAN perform similarly to the standard version but with much less computational cost (Figure 10(c.2)). However, when the dimensionality of raw time series becomes increasingly large (Figure 10(d.1)), we observe SPARTAN of all versions is the only family of methods that can consistently maintain the performance, while SFA and SAX experience

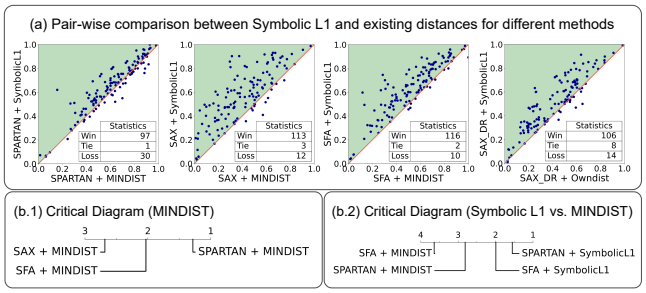

Figure 11: Ablation study on the symbolic distances (evaluated on 1NN classification task). (a) pair-wise comparison between Symbolic L_1 and existing distance (denoted as MINDIST or "Owndist"); and (b) Average rankings on SPARTAN, SFA, and SAX, over MINDIST and Symbolic L_1.

significant performance degradation at different levels. This can be attributed to the challenge in reconstructing information using mean values or low-frequency components when facing the increasingly large dimensionality, while SPARTAN methods address this issue by prioritizing the most informative symbols with a non-uniform policy at both the approximation and discretization stages. Importantly, SPARTAN-S, achieving a 2× speed-up compared to SFA on 1M samples in Figure 10(c.2) and extremely long time series with 128K time steps (Figure 10(d.2)), without compromising the representation quality. These findings highlight SPARTAN as a robust solution for balancing the representation capability and runtime efficiency on large volumes of data for practical applications.

5.7 Revisiting the Symbolic Distance Measure

As illustrated in Section 4, given the absence of a unified testbed, we proposed Symbolic L_1, a simple yet effective way that provides consistent discriminative power for measuring the dissimilarity for all methods. As illustrated in Section 4, this distance measures the sum of ordinal differences between discretized symbols, relying on a simple assumption that the indices of symbols can be meaningfully ordered, which is compatible with all existing methods we have sampled, as well as new methods that may emerge in future evaluation. Since it requires no prior knowledge of given methods, this distance is unlikely to unfairly favor one representation over another. Without losing generality, we re-conduct the experiments on 1NN classification using Symbolic L_1 and existing distances to validate its effectiveness. As we can see in Figure 11(a), Symbolic L_1 consistently improves classification accuracy over existing distances across various symbolic methods, including SPARTAN, SAX, SFA, and SAX variants. More importantly, we also observe the same trend and conclusion when comparing both Symbolic L_1 and exiting distance such as MINDIST (Figure 11(b)). These results highlight its stronger discriminative power, demonstrating its potential as a robust and general-purpose distance measure.

5.8 Comparing with Euclidean Distance

Symbolic methods provide highly interpretable representation with significantly reduced dimension from the original space, offering the advantages of low computational and storage costs. However, it is noteworthy that this process of approximation and discretization inevitably lead to information loss from the raw data. In this section, we evaluate ED on raw time series as a valuable reference point for

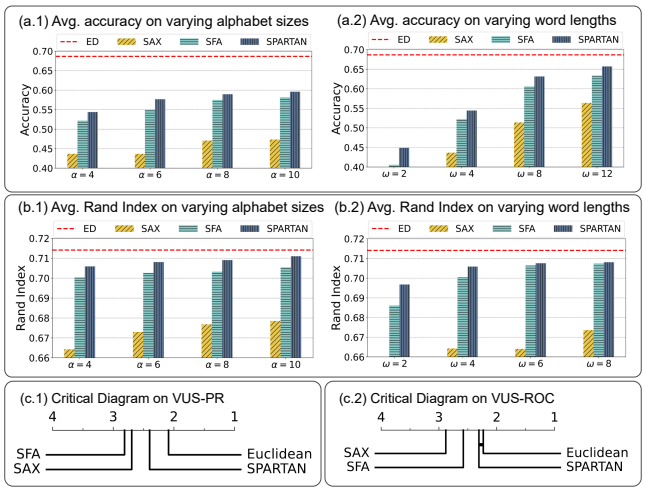

Figure 12: Comparison between symbolic representations and Euclidean distance (ED) on raw time series: (a) classification; (b) clustering; and (c) anomaly detection. In (a) and (b), ED performance is represented by the red dashed line.

comparison with symbolic methods. Specifically, we perform (i) ED + 1NN for classification on UCR datasets [13]; (ii) ED + KMedoids for clustering on UCR datasets; and (iii) ED + KNN on TSB-UAD [59].

As shown in Figure 12, there is a clear performance gap between symbolic representation and ED, attributable to information loss from dimensionality reduction. However, with a larger encoding budget – for example, with larger alphabet sizes and longer word lengths – the performance gap between ED and symbolic methods narrows, aligning with expectations. This trend is consistent across both classification (Figure 12(a)) and clustering (Figure 12(b)) tasks. For the anomaly detection task, SPARTAN shows no statistically significant difference compared to ED + KNN in terms of VUS-ROC (Figure 12(c)). Across all experiments, SPARTAN consistently outperforms its competitors and achieves performance closest to ED, demonstrating superior representation efficiency and pruning power within the constraints of a given encoding budget.

5.9 Summary of Results

In summary, our extensive experimental results suggest that (i) despite elaborate features, none of the SAX variants strongly outperform the original SAX under the same budget, while SFA outperforms all other existing methods; (ii) SPARTAN significantly outperforms SAX and SFA in both supervised (1NN classification) and unsupervised context (clustering and anomaly detection), which indicates the robust representation ability in capturing both normal and abnormal patterns; (iii) SPARTAN consistently maintains a high-quality representation with stronger pruning power in terms of TLB; and (iv) SPARTAN methods achieves 2× speed up over SFA on large-scale datasets while maintaining the top performance. Furthermore, we further demonstrate the robustness and fairness of our proposed Symbolic L_1, and evaluate ED on raw time series as a valuable reference point for comparison with symbolic methods. Overall, SPARTAN strikes a better representation quality without introducing storage or runtime overheads.

6 CONCLUSION

In this work, we present SPARTAN, a novel symbolic approximation method that intelligently allocates the encoding budget according to the significance of each dimension. To demonstrate the SPARTAN's robustness, we conducted the most comprehensive evaluation of symbolic methods to date, along with seven state-of-the-art methods. Our experimental results reveal that the no SAX variants are able to strongly outperform SAX given the same budget, while SFA outperforms all other methods. We demonstrate the superior performance of SPARTAN against SAX and SFA across all four downstream tasks, including classification, clustering, indexing (tightness of lower bound), and anomaly detection, without introducing additional storage and computation overheads. Overall, this research work paves the way for future research on producing high-quality symbolic representations of time-series data.

REFERENCES

[1] 2024. Anonymized GitHub. https://anonymous.4open.science/r/SPARTAN. Accessed: 2024-10-17.

[2] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. 1993. Efficient similarity search in sequence databases. In *Foundations of Data Organization and Algorithms*, David B. Lomet (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 69–84.

[3] Shadab Alam, Franco D Albareti, Carlos Allende Prieto, Friedrich Anders, Scott F Anderson, Timothy Anderton, Brett H Andrews, Eric Armengaud, Éric Aubourg, Stephen Bailey, et al. 2015. The eleventh and twelfth data releases of the Sloan Digital Sky Survey: final data from SDSS-III. *The Astrophysical Journal Supplement Series* 219, 1 (2015), 12.

[4] Henrik André-Jönsson and Dushan Z. Badal. 1997. Using signature files for querying time-series data. In *Principles of Data Mining and Knowledge Discovery*, Jan Komorowski and Jan Zytkow (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 211–220.

[5] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. 2024. Chronos: Learning the Language of Time Series. *arXiv preprint arXiv:2403.07815* (2024).

[6] Alberto Apostolico, Mary Ellen Bock, and Stefano Lonardi. 2002. Monotony of surprise and large-scale quest for unusual words. In *Proceedings of the Sixth Annual International Conference on Computational Biology* (Washington, DC, USA) *(RECOMB '02)*. Association for Computing Machinery, New York, NY, USA, 22–31. https://doi.org/10.1145/565196.565200

[7] Anthony Bagnall, Aaron Bostrom, James Large, and Jason Lines. 2016. The Great Time Series Classification Bake Off: An Experimental Evaluation of Recently Proposed Algorithms. Extended Version. arXiv:1602.01711 [cs.LG]

[8] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. 2018. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075* (2018).

[9] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery* 31 (2017), 606–660.

[10] Konstantinos Bountrogiannis, George Tzagkarakis, and Panagiotis Tsakalides. 2021. Anomaly detection for symbolic time series representations of reduced dimensionality. In *2020 28th European Signal Processing Conference (EUSIPCO)*. Ieee, 2398–2402.

[11] Kin-Pong Chan and Ada Wai-Chee Fu. 1999. Efficient time series matching by wavelets. In *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*. 126–133. https://doi.org/10.1109/ICDE.1999.754915

[12] Xinye Chen and Stefan Güttel. 2023. An Efficient Aggregation Method for the Symbolic Representation of Temporal Data. *ACM Trans. Knowl. Discov. Data* 17, 1, Article 5 (feb 2023), 22 pages. https://doi.org/10.1145/3532622

[13] Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. 2018. The UCR Time Series Classification Archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

[14] C. S. Daw, C. E. A. Finney, and E. R. Tracy. 2003. A review of symbolic analysis of experimental data. *Review of Scientific Instruments* 74, 2 (02 2003), 915–930. https://doi.org/10.1063/1.1531823 arXiv:https://pubs.aip.org/aip/rsi/article-pdf/74/2/915/19150542/915_1_online.pdf

[15] Janez Demšar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* 7 (dec 2006), 1–30.

[16] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2020. The Lernaean Hydra of Data Series Similarity Search: An Experimental Evaluation of the State of the Art. arXiv:2006.11454 [cs.DB]

[17] Steven Elsworth and Stefan Güttel. 2020. ABBA: adaptive Brownian bridge-based symbolic aggregation of time series. *Data Mining and Knowledge Discovery* 34, 4 (2020), 1175–1200.

[18] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. 1994. Fast subsequence matching in time-series databases. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data* (Minneapolis, Minnesota, USA) *(SIGMOD '94)*. Association for Computing Machinery, New York, NY, USA, 419–429. https://doi.org/10.1145/191839.191925

[19] Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association* 32, 200 (1937), 675–701.

[20] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2013. Optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2946–2953.

[21] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2013. Optimized Product Quantization for Approximate Nearest Neighbor Search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[22] Pierre Geurts. 2001. Pattern Extraction for Time Series Classification. In *Principles of Data Mining and Knowledge Discovery*, Luc De Raedt and Arno Siebes (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 115–127.

[23] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. MOMENT: A Family of Open Time-series Foundation Models. In *International Conference on Machine Learning*.

[24] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. 2023. Large Language Models Are Zero-Shot Time Series Forecasters. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 19622–19635. https://proceedings.neurips.cc/paper_files/paper/2023/file/3eb7ca52e8207697361b2c0fb3926511-Paper-Conference.pdf

[25] N. Halko, P. G. Martinsson, and J. A. Tropp. 2011. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Rev.* 53, 2 (2011), 217–288. https://doi.org/10.1137/090771806 arXiv:https://doi.org/10.1137/090771806

[26] Xiaoxu He. 2023. A Survey on Time Series Forecasting. In *3D Imaging—Multidimensional Signal Processing and Deep Learning*, Srikanta Patnaik, Roumen Kountchev, Yonghang Tai, and Roumiana Kountcheva (Eds.). Springer Nature Singapore, Singapore, 13–23.

[27] Xin He, Kaiyong Zhao, and Xiaowen Chu. 2021. AutoML: A survey of the state-of-the-art. *Knowledge-based systems* 212 (2021), 106622.

[28] Christopher Holder, Matthew Middlehurst, and Anthony Bagnall. 2024. A review and evaluation of elastic distance functions for time series clustering. *Knowledge and Information Systems* 66, 2 (2024), 765–809.

[29] Yun-Wu Huang and Philip S. Yu. 1999. Adaptive query processing for time-series data. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Diego, California, USA) *(KDD '99)*. Association for Computing Machinery, New York, NY, USA, 282–286. https://doi.org/10.1145/312129.318357

[30] Félix Iglesias and Wolfgang Kastner. 2013. Analysis of similarity measures in times series clustering for the discovery of building energy patterns. *Energies* 6, 2 (2013), 579–597.

[31] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. 2024. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=Unb5CVPtae

[32] Ian T. Jolliffe. 2002. *Principal Component Analysis* (2 ed.). Springer New York. https://doi.org/10.1007/b98835

[33] Ian T. Jolliffe and Jorge Cadima. 2016. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374, 2065 (April 2016), 20150202. https://doi.org/10.1098/rsta.2015.0202 Publisher: Royal Society.

[34] Shubhra Kanti Karmaker ("Santu"), Md. Mahadi Hassan, Micah J. Smith, Lei Xu, Chengxiang Zhai, and Kalyan Veeramachaneni. 2021. AutoML to Date and Beyond: Challenges and Opportunities. *ACM Comput. Surv.* 54, 8, Article 175 (Oct. 2021), 36 pages. https://doi.org/10.1145/3470918

[35] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. 2001. Locally adaptive dimensionality reduction for indexing large time series databases. *SIGMOD Rec.* 30, 2 (may 2001), 151–162. https://doi.org/10.1145/376284.375680

[36] Eamonn Keogh and Abdullah Mueen. 2017. *Curse of Dimensionality*. Springer US, Boston, MA, 314–315. https://doi.org/10.1007/978-1-4899-7687-1_192

[37] Eamonn J. Keogh, Kaushik Chakrabarti, Michael J. Pazzani, and Sharad Mehrotra. 2001. Dimensionality Reduction for Fast Similarity Search in Large Time Series

Databases. *Knowledge and Information Systems* 3 (2001), 263–286. https://api.semanticscholar.org/CorpusID:15462747

[38] S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22, 1 (1951), 79 – 86. https://doi.org/10.1214/aoms/1177729694

[39] Tianyu Li, Fang-Yan Dong, and Kaoru Hirota. 2013. Distance Measure for Symbolic Approximation Representation with Subsequence Direction for Time Series Data Mining. *Journal of Advanced Computational Intelligence and Intelligent Informatics* 17, 2 (2013), 263–271.

[40] Xiang Li, Jinglu Wang, Xiaohao Xu, Muqiao Yang, Fan Yang, Yizhou Zhao, Rita Singh, and Bhiksha Raj. 2023. Towards Noise-Tolerant Speech-Referring Video Object Segmentation: Bridging Speech and Text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 2283–2296. https://doi.org/10.18653/v1/2023.emnlp-main.140

[41] Yuan Li, Jessica Lin, and Tim Oates. [n. d.]. *Visualizing Variable-Length Time Series Motifs*. 895–906. https://doi.org/10.1137/1.9781611972825.77 arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611972825.77

[42] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. 2003. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*. 2–11.

[43] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. 2007. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and knowledge discovery* 15 (2007), 107–144.

[44] Jessica Lin, Rohan Khade, and Yuan Li. 2012. Rotation-invariant similarity in time series using bag-of-patterns representation. *Journal of Intelligent Information Systems* 39 (2012), 287 – 315. https://api.semanticscholar.org/CorpusID:873260

[45] Jason Lines, Sarah Taylor, and Anthony Bagnall. 2016. HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles for Time Series Classification. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. 1041–1046. https://doi.org/10.1109/ICDM.2016.0133

[46] Battuguldur Lkhagva, Yu Suzuki, and Kyoji Kawagoe. 2006. Extended SAX: Extension of symbolic aggregate approximation for financial time series data representation. *DEWS2006 4A-i8* 7 (2006).

[47] JLEKS Lonardi and Pranav Patel. 2002. Finding motifs in time series. In *Proc. of the 2nd Workshop on Temporal Data Mining*. 53–68.

[48] Mohammad Saeid Mahdavinejad, Mohammadreza Rezvan, Mohammadamin Barekatain, Peyman Adibi, Payam Barnaghi, and Amit P. Sheth. 2018. Machine learning for internet of things data analysis: a survey. *Digital Communications and Networks* 4, 3 (2018), 161–175. https://doi.org/10.1016/j.dcan.2017.10.002

[49] Simon Malinowski, Thomas Guyet, René Quiniou, and Romain Tavenard. 2013. 1d-sax: A novel symbolic representation for time series. In *International Symposium on Intelligent Data Analysis*. Springer, 273–284.

[50] Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. 2011. A randomized algorithm for the decomposition of matrices. *Applied and Computational Harmonic Analysis* 30, 1 (2011), 47–68. https://doi.org/10.1016/j.acha.2010.02.003

[51] Matthew Middlehurst, James Large, Gavin Cawley, and Anthony Bagnall. 2021. *The Temporal Dictionary Ensemble (TDE) Classifier for Time Series Classification*. Springer International Publishing, 660–676. https://doi.org/10.1007/978-3-030-67658-2_38

[52] Matthew Middlehurst, William Vickers, and Anthony Bagnall. 2019. *Scalable Dictionary Classifiers for Time Series Classification*. Springer International Publishing, 11–19. https://doi.org/10.1007/978-3-030-33607-3_2

[53] Peter Bjorn Nemenyi. 1963. *Distribution-free multiple comparisons*. Princeton University.

[54] Thach Le Nguyen and Georgiana Ifrim. 2022. MrSQM: Fast Time Series Classification with Symbolic Representations. arXiv:2109.01036 [cs.LG]

[55] John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S Tsay, Aaron Elmore, and Michael J Franklin. 2022. Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection. *Proceedings of the VLDB Endowment* 15, 11 (2022), 2774–2787.

[56] John Paparrizos, Ikraduya Edian, Chunwei Liu, Aaron J Elmore, and Michael J Franklin. 2022. Fast adaptive similarity search through variance-aware quantization. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2969–2983.

[57] John Paparrizos and Michael J. Franklin. 2019. GRAIL: efficient time-series representation learning. *Proc. VLDB Endow.* 12, 11 (jul 2019), 1762–1777. https://doi.org/10.14778/3342263.3342648

[58] John Paparrizos and Luis Gravano. 2015. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 1855–1870.

[59] John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S Tsay, Themis Palpanas, and Michael J Franklin. 2022. Tsb-uad: an end-to-end benchmark suite for univariate time-series anomaly detection. *Proceedings of the VLDB Endowment* 15, 8 (2022), 1697–1711.

[60] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient algorithms for mining outliers from large data sets. *SIGMOD Rec.* 29, 2 (May 2000), 427–438. https://doi.org/10.1145/335191.335437

[61] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (Dallas, Texas, USA) *(SIGMOD '00)*. Association for Computing Machinery, New York, NY, USA, 427–438. https://doi.org/10.1145/342009.335437

[62] William M Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association* 66, 336 (1971), 846–850.

[63] Naoki Saito. 2000. LOCAL FEATURE EXTRACTION AND ITS APPLICATIONS USING A LIBRARY OF BASES. *Topics in Analysis and Its Applications: Selected Theses* (2000), 269.

[64] Patrick Schäfer. 2015. The BOSS is concerned with time series classification in the presence of noise. *Data Min. Knowl. Discov.* 29, 6 (nov 2015), 1505–1530. https://doi.org/10.1007/s10618-014-0377-7

[65] Patrick Schäfer and Mikael Högqvist. 2012. SFA: a symbolic fourier approximation and index for similarity search in high dimensional datasets. In *Proceedings of the 15th International Conference on Extending Database Technology* (Berlin, Germany) *(EDBT '12)*. Association for Computing Machinery, New York, NY, USA, 516–527. https://doi.org/10.1145/2247596.2247656

[66] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly detection in time series: a comprehensive evaluation. *Proc. VLDB Endow.* 15, 9 (may 2022), 1779–1797. https://doi.org/10.14778/3538598.3538602

[67] Patrick Schäfer and Ulf Leser. 2017. Fast and Accurate Time Series Classification with WEASEL. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. ACM. https://doi.org/10.1145/3132847.3132980

[68] Pavel Senin, Jessica Lin, Xing Wang, Tim Oates, Sunil Gandhi, Arnold P Boedihardjo, Crystal Chen, and Susan Frankenstein. 2015. Time series anomaly discovery with grammar-based compression.. In *Edbt*. 481–492.

[69] Pavel Senin, Jessica Lin, Xing Wang, Tim Oates, Sunil Gandhi, Arnold P Boedihardjo, Crystal Chen, and Susan Frankenstein. 2018. Grammarviz 3.0: Interactive discovery of variable-length time series patterns. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12, 1 (2018), 1–28.

[70] Pavel Senin, Jessica Lin, Xing Wang, Tim Oates, Sunil Gandhi, Arnold P Boedihardjo, Crystal Chen, Susan Frankenstein, and Manfred Lerner. 2014. Grammarviz 2.0: a tool for grammar-based pattern discovery in time series. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part III 14*. Springer, 468–472.

[71] Pavel Senin and Sergey Malinchik. 2013. SAX-VSM: Interpretable Time Series Classification Using SAX and Vector Space Model. In *2013 IEEE 13th International Conference on Data Mining*. 1175–1180. https://doi.org/10.1109/ICDM.2013.52

[72] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing* 90 (2020), 106181.

[73] Jin Shieh and Eamonn Keogh. 2008. iSAX: indexing and mining terabyte sized time series. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Las Vegas, Nevada, USA) *(KDD '08)*. Association for Computing Machinery, New York, NY, USA, 623–631. https://doi.org/10.1145/1401890.1401966

[74] Numanul Subhani, Luis Rueda, Alioune Ngom, and Conrad J Burden. 2010. Multiple gene expression profile alignment for microarray time-series data clustering. *Bioinformatics* 26, 18 (2010), 2281–2288.

[75] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*.

[76] Lijuan Yan, Xiaotao Wu, and Jiaqing Xiao. 2022. An Improved Time Series Symbolic Representation Based on Multiple Features and Vector Frequency Difference. *Journal of Computer and Communications* 10, 06 (2022), 44–62.

[77] Fan Yang, Muqiao Yang, Xiang Li, Yuxuan Wu, Zhiyuan Zhao, Bhiksha Raj, and Rita Singh. 2024. A closer look at reinforcement learning-based automatic speech recognition. *Computer Speech & Language* 87 (2024), 101641. https://doi.org/10.1016/j.csl.2024.101641

[78] Ling Yang and Shenda Hong. 2022. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *International Conference on Machine Learning*. PMLR, 25038–25054.

[79] Yufeng Yu, Yuelong Zhu, Dingsheng Wan, Qun Zhao, and Huan Liu. 2019. A novel trend symbolic aggregate approximation for time series. *arXiv preprint arXiv:1905.00421* (2019).

[80] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. 2022. TS2Vec: Towards Universal Representation of Time Series. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 8 (Jun. 2022), 8980–8987. https://doi.org/10.1609/aaai.v36i8.20881

[81] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. 2021. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 2114–2124.

[82] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. 2022. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems* 35 (2022), 3988–4003.
[83] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems* 36 (2023), 43322–43355.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009