Accelerating Similarity Search for Elastic Measures: A Study and New Generalization of Lower Bounding Distances

John Paparrizos*
The Ohio State University
paparrizos.1@osu.edu

Kaize Wu*
University of Chicago
kaizewu@uchicago.edu

Aaron Elmore
University of Chicago
aelmore@uchicago.edu

Christos Faloutsos
Carnegie Mellon University
christos@cs.cmu.edu

Michael J. Franklin
University of Chicago
mjfranklin@uchicago.edu

ABSTRACT

Similarity search is a core analytical task, and its performance critically depends on the choice of distance measure. For time-series querying, elastic measures achieve state-of-the-art accuracy but are computationally expensive. Thus, fast lower bounding (LB) measures prune unnecessary comparisons with elastic distances to accelerate similarity search. Despite decades of attention, there has never been a study to assess the progress in this area. In addition, the research has disproportionately focused on one popular elastic measure, while other accurate measures have received little or no attention. Therefore, there is merit in developing a framework to accumulate knowledge from previously developed LBs and eliminate the notoriously challenging task of designing separate LBs for each elastic measure. In this paper, we perform the first comprehensive study of 11 LBs spanning 5 elastic measures using 128 datasets. We identify four properties that constitute the effectiveness of LBs and propose the Generalized Lower Bounding (GLB) framework to satisfy all desirable properties. GLB creates cache-friendly summaries, adaptively exploits summaries of both query and target time series, and captures boundary distances in an unsupervised manner. GLB outperforms *all* LBs in speedup (e.g., up to 13.5× faster against the strongest LB in terms of pruning power), establishes new state-of-the-art results for the 5 elastic measures, and provides the first LBs for 2 elastic measures with no known LBs. Overall, GLB enables the effective development of LBs to facilitate fast similarity search.

PVLDB Reference Format:

John Paparrizos, Kaize Wu, Aaron Elmore, Christos Faloutsos, and Michael J. Franklin. Accelerating Similarity Search for Elastic Measures: A Study and New Generalization of Lower Bounding Distances. PVLDB, 16(8): 2019 - 2032, 2023. doi:10.14778/3594512.3594530

PVLDB Artifact Availability:

The source code, data, and/or other artifacts have been made available at https://www.timeseries.org/glb.

1 INTRODUCTION

The ubiquity and unprecedented growth of time-varying measurements across scientific and industrial settings are responsible for the increasing popularity of time-series analysis [35, 42, 43, 52, 59, 61, 64, 77, 79, 80]. The backbone of most time-series analytical tasks requires the detection of similar pairs of time series [27, 78]. Specifically, identifying similarities plays a central role in querying [24, 45, 50, 51, 55, 65, 68, 69, 87], indexing [17, 18, 21, 32, 48, 72, 91, 96, 103], motif discovery [7, 23, 57, 67, 99, 100], anomaly detection [9–15, 60, 71, 76, 84, 98, 99], classification [4, 29, 37, 40, 58, 73, 81], and clustering [1, 5, 6, 28, 44, 47, 56, 74, 75]. Consequently, *similarity search*, the process of retrieving the nearest neighbors to a query from a database under a certain distance measure, becomes one of the most fundamental building blocks in time-series analysis.

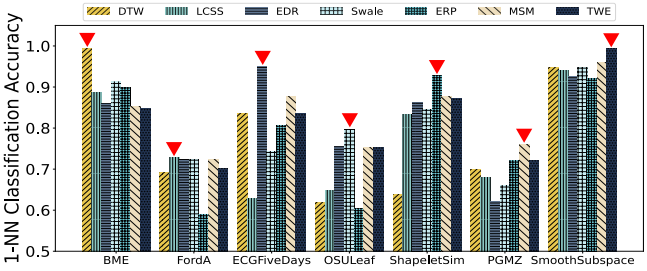


Figure 1: Classification accuracy of 7 elastic measures on sampled UCR datasets [26]. The triangle indicates the measure with the best accuracy: different winner in each dataset.

Unfortunately, the rising volumes of time series and their high dimensionality introduce severe challenges for similarity search [2, 25]. Specifically, the computational and storage costs of retrieving the nearest neighbors become prohibitively high even when the Euclidean distance (ED) is used [30, 31, 93, 94]. However, there is strong evidence that ED might not always be suitable for comparing time series [4, 27, 78, 95]. We often need to handle several distortions for effective time-series comparison, such as misalignments, stretching of observations, and fluctuations. To handle these distortions, dozens of distance measures have been proposed in the time-series literature [3, 8, 19, 20, 22, 27, 32, 33, 66, 73, 74, 83, 88, 92].

A recent study has evaluated over 70 time-series distance measures [78] and reaffirmed that *elastic* measures, which create a non-linear mapping between time series to align or stretch their points, achieve state-of-the-art performance in terms of accuracy. Importantly, the same study has shown that the nearest neighbor (classification) accuracy of ED may not always converge to the high accuracy of elastic measures with increasing dataset sizes, which contradicts the previous belief [86]. Therefore, supporting

*Authors contributed equally.

Proceedings of the VLDB Endowment, Vol. 16, No. 8 ISSN 2150-8097.
doi:10.14778/3594512.3594530

similarity search under different elastic measures is necessary for enabling effective large-scale time-series analytics.

Unfortunately, elastic distance measures scale quadratically (i.e., $O(L^2)$) to the length L of the time series. Compared to ED, which has linear complexity (i.e., $O(L)$), elastic distance measures incur an additional runtime overhead, often between one to three orders of magnitude [4, 78, 90]. This cost would prevent applications from using elastic measures in large-scale settings and favor sacrificing the high accuracy by relying on faster but less accurate measures. To alleviate this issue, the idea of *lower bounding* was developed to filter out unpromising candidates before carrying out the expensive elastic distance measure computation [32, 48, 49]. In simple terms, a lower bound (LB) is a fast distance measure that approximates an expensive elastic distance measure and is computed over some summaries of the time series instead of the actual time series.

Numerous LBs have been developed for elastic distance measures [19, 48, 49, 54, 85, 89, 90], with the goal to improve their pruning power (i.e., *tightness* of LB). However, a tighter LB does not always translate into higher speedup due to the potentially high computational cost necessary to calculate the LB. Despite over two decades of attention, there has never been, to the best of our knowledge, a comprehensive study to assess the progress in this area. In addition, the research effort on LBs has been disproportionally concentrated on Dynamic Time Warping (DTW) [82, 83], which is the oldest elastic measure with at least eight established LBs [48, 49, 54, 85, 89, 97, 101]. In contrast, other useful elastic distance measures, such as EDR [20] or SWALE [66], do not have any LBs reported in the literature, to the best of our knowledge. Importantly, as shown in Figure 1, no single elastic measure always wins on every dataset (as confirmed by a recent study [78]). Considering that all elastic measures are useful in practical time-series tasks, there is merit in developing LBs for elastic distance measures that have received relatively little attention.

Unfortunately, developing LBs is a challenging task. It took about two decades to improve the tightness of DTW LBs to over 80% [54, 85, 89, 90]. We believe it is unsustainable to expect a similar research effort for each elastic measure. Instead, a generalized framework that accumulates the knowledge from previously developed LBs would eliminate the need for designing separate LBs for each elastic measure. Based on our review and evaluation of existing LBs for DTW, we identified four critical properties that constitute the effectiveness of LBs: ($\mathcal{P}1$) *Query Dependence*, which indicates the LB extracts summaries only from the query time series; ($\mathcal{P}2$) *Data Dependence*, which indicates the LB extracts summaries from both the query and the target data time series; ($\mathcal{P}3$) *Boundary Dependence*, which indicates the LB explicitly captures distances of the first and last elements of time series; and ($\mathcal{P}4$) *Reusability*, which indicates the ability to cache and reuse results in future calculations.

This paper performs the first comprehensive study of LBs for elastic measures. Specifically, we evaluate 11 state-of-the-art LBs spanning 5 elastic measures using 128 datasets. In addition, we present the Generalized Lower Bounding (GLB) framework, the first framework to create LBs to satisfy all four previously mentioned desirable properties and adapt to all elastic measures (see Table 1 for comparison with baselines). LBs created within the GLB framework are query- and data-dependent and extract summaries (i.e., envelopes, see Section 2) from both the query and target data time series. In the GLB framework, LBs compute distances between all envelopes and the target time series, and adaptively select the envelopes that maximize the LB tightness for each pairwise comparison. To satisfy the boundary dependence, LBs under the GLB framework avoid the need to tune parameters and focus only on the leading and trailing time-series points, whereas existing LBs rely on supervised solutions. GLB's ability to reuse the extracted envelopes also avoids the significant overhead introduced by existing solutions focusing on complex and expensive transformations to capture characteristics from the target time series. Finally, GLB is adaptable to all elastic measures due to its abstraction of the different cost functions used internally in elastic distances.

Table 1: Analysis of LBs on four properties. GLB matches all properties, while competitors miss one or more.

Elastic Measure	Lower Bounds	Query Dependent	Data Dependent	Boundary Dependent	Reusable	Year & Reference
DTW	LB_Yi	✔	-	-	✔	1998 [101]
	LB_Kim	✔	-	✔	✔	2001 [49]
	LB_Keogh	✔	-	-	✔	2002 [46, 48]
	LB_Improved	✔	✔	-	-	2009 [54]
	LB_New	✔	-	✔	-	2018 [85]
	LB_Enhanced	✔	-	✔	-	2019 [89]
	LB_Petitjean	✔	✔	✔	-	2021 [97]
	LB_Webb	✔	✔	✔	-	2021 [97]
LCSS	LB_LCSS	✔	-	-	✔	2002 [62, 91, 92]
ERP	LB_Keogh-ERP	✔	-	-	✔	2004 [19]
	LB_Kim-ERP	✔	-	-	✔	2004 [19]
	LB_ERP	✔	-	✔	✔	2004 [19]
MSM	LB_MSM	✔	-	-	✔	2020 [90]
TWED	LB_TWED	✔	-	-	✔	2020 [90]
EDR	N/A	-	-	-	-	-
SWALE	N/A	-	-	-	-	-
Proposed Generalized Lower Bound						
All Elastic Measures	**GLB**	✔	✔	✔	✔	2023 (this work)

We compare GLB against the 11 state-of-the-art LBs, and we make all source codes available to ensure reproducibility.[1] Compared to the strongest LBs for DTW, GLB_DTW achieves state-of-the-art pruning power and outperforms *all* LBs in terms of speedup. Specifically, GLB_DTW is up to 13.5× faster (6.8× faster on average) when compared against the strongest LB in terms of pruning power and wins in at least 115 out of 128 (90%) datasets. For elastic measures other than DTW, GLB establishes new state-of-the-art results in both pruning power and speedup, outperforming all baselines significantly. Importantly, for elastic measures without known LBs, GLB achieves performance comparable to the results for the other elastic measures, which demonstrates the generalizability of GLB.

We present our contributions as follows:

- We provide the first thorough study of elastic measures and LBs, summarizing two decades of progress (Section 2).
- We propose the GLB framework for developing LBs for elastic measures while satisfying four properties: query, data, and boundary dependence, and reusability (Section 3.1).
- We formally define GLB by abstracting costs and combining elements that constitute the effective LBs (Sections 3.2-3.3).
- We prove the correctness of GLB (Section 3.4).
- We present new LBs for 7 elastic distance measures, including two elastic measures without known LBs (Section 4).
- We conduct the most extensive evaluation of LBs until now to demonstrate the robustness of GLB (Sections 5 and 6).

Finally, we summarize the implications of our work (Section 7).

[1] www.timeseries.org/glb

2 PRELIMINARIES AND RELATED WORK

In this section, we first review the development of elastic distance measures starting with DTW, the earliest and most popular elastic measure. We provide a generalized formula to showcase the recursive (dynamic programming) computation in all elastic measures, which highlights the different cost functions adopted by each elastic measure (Section 2.1). Then, we present LBs to accelerate similarity search for elastic measures and our problem of focus (Section 2.2).

2.1 Elastic Time Series Distance Measures

We now review the development of elastic distance measures.

Dynamic Time Warping (DTW) [82, 83]: We consider a time-series $\mathbf{x} = [x_1, x_2, ..., x_L]$, an ordered sequence of L datapoints. To measure the similarity between two time series, the most common distance measure is ED, which offers a one-to-one (linear) alignment between two time series, as shown in Figure 2(a). ED defines the distance between time series $\mathbf{x} = [x_1, x_2, ..., x_{L_\mathbf{x}}]$ and $\mathbf{y} = [y_1, y_2, ..., y_{L_\mathbf{y}}]$, with $L_\mathbf{x} = L_\mathbf{y} = L$, as $ED(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{L}(x_i - y_i)^2}$. The disadvantage of ED is its inability to recognize the similarity between time series that have comparable shapes but are stretched or differ in phase or length. The need to capture shape similarity despite these distortions inspired the development of *elastic measures*, which create a non-linear mapping between time series by comparing one-to-many points in order to align or stretch points. For example, DTW enables local alignments by permitting one-to-many points matching, as shown in Figure 2(b), where the peaks and troughs of two time series are aligned correspondingly. To find the local alignments, DTW first computes a distance matrix, D, using the following recursive computation:

$$D(i,j) = \begin{cases} (x_i - y_j)^2 & \text{if } i, j = 1 \\ D(i-1,j) + (x_i - y_j)^2 & \text{if } i \neq 1 \text{ and } j = 1 \\ D(i,j-1) + (x_i - y_j)^2 & \text{if } i = 1 \text{ and } j \neq 1 \\ min \begin{cases} D(i-1,j-1) + (x_i - y_j)^2 \\ D(i-1,j) + (x_i - y_j)^2 \\ D(i,j-1) + (x_i - y_j)^2 \end{cases} & \text{if } i, j \neq 1 \end{cases} \quad (1)$$

DTW determines the alignment path, $W = \{w_1, w_2, ..., w_p\}$, which starts from the bottom-left corner and ends at the top-right corner in the matrix (gray path in Figure 2) where the distance of alignments add up to the cell in the top-right corner, or $D(L_\mathbf{x}, L_\mathbf{y}) = \sum_{i=1}^{p}(\mathbf{x}_{W_k{}^1} - \mathbf{y}_{W_k{}^2})^2$. The warping path follows two properties [89]:

- **Boundary Constraints**: $w_1 = D(1,1)$ and $w_p = D(L_\mathbf{x}, L_\mathbf{y})$, meaning the optimal warping path has to start on the bottom-left corner of D and to end on the upper-right corner of D.
- **Continuity and Monotonicity**: if $w_k = D(i,j)$ for $k \in [2, p-1]$, $w_{k+1} \in \{D(i+1,j), D(i,j+1), D(i+1,j+1)\}$, meaning that the warping path, starting from bottom-left, only moves vertically upwards, horizontally towards the right, or diagonally towards top-right continuously until arriving at the top-right corner, as shown on the right side of Figure 2(b). DTW uses the same distance function in each matrix cell regardless of whether the optimal path arrives at that cell horizontally, vertically, or diagonally (Formula 1).

In contrast to DTW, alternative elastic measures that we review next have three different distance functions, each corresponding to *diagonal* movements, *horizontal* movements, and *vertical* movements, respectively (Figure 2(b) depicts these three costs). As we discuss in Section 3, having different distance functions for diagonal and horizontal/vertical movements presents a unique challenge for proposing effective lower bounds for particular elastic measures. GLB provides a principled solution to tackle this critical problem. After computing distance matrix D and optimal warping path W, the distance between the time series $\mathbf{x}$ and $\mathbf{y}$ of lengths $L_\mathbf{x}$ and $L_\mathbf{y}$ is defined as $DTW(\mathbf{x}, \mathbf{y}) = \sqrt{D(L_\mathbf{x}, L_\mathbf{y})} = \sqrt{\sum_{i=1}^{p} w_i}$.

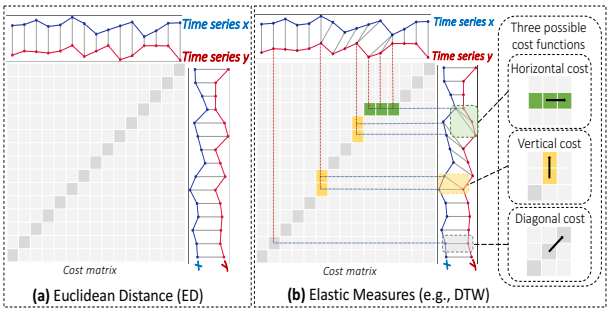


Figure 2: Cost matrix for (a) ED and (b) elastic measures.

DTW is commonly used together with a locality constraint to limit the range of warping allowed and thereby avoiding unreasonably far-reaching alignments; this approach is referred to as Constrained DTW (cDTW) [36]. cDTW is faster than DTW and often results in higher classification accuracy [78]. The most widely adopted locality constraint is the Sakoe-Chiba band [36], which we refer to as the warping window. The warping window, w, is the maximum possible deviation of the alignment path from the diagonal of D, and cells further away are not computed.

Generalization of Elastic Measures: Since the development of DTW, numerous alternative elastic measures have been proposed to address certain limitations. These measures have employed different distance functions for diagonal and vertical/horizontal movements and introduced additional parameters to tackle weaknesses of DTW.

Despite the differences in parameters and distance functions, subsequent elastic measures and DTW share a common goal of finding the warping path through dynamic programming to compute D, so elastic measures can be generalized as follows:

$$D(i,j) = \begin{cases} initial\_distance(x_i, y_j) & \text{if } i, j = 1 \\ D(i-1,j) + dist^V(x_i, y_j) & \text{if } i \neq 1 \text{ and } j = 1 \\ D(i,j-1) + dist^H(x_i, y_j) & \text{if } i = 1 \text{ and } j \neq 1 \\ min \begin{cases} D(i-1,j-1) + dist^D(x_i, y_j) \\ D(i-1,j) + dist^V(x_i, y_j) \\ D(i,j-1) + dist^H(x_i, y_j) \end{cases} & \text{if } i, j \neq 1 \end{cases} \quad (2)$$

where $dist^D(x_i, y_j)$, $dist^V(x_i, y_j)$, and $dist^H(x_i, y_j)$ are the distance functions for diagonal, vertical, and horizontal movements, respectively, and $initial\_distance(x_i, y_j)$ is the distance function for initial alignment in $D(1,1)$. After computing $D(L_\mathbf{x}, L_\mathbf{y})$, each elastic measure adopts a function, $transD(L_\mathbf{x}, L_\mathbf{y})$, that transforms distance in $D(L_\mathbf{x}, L_\mathbf{y})$ to calculate distance between $\mathbf{x}$ and $\mathbf{y}$ based on $D(L_\mathbf{x}, L_\mathbf{y})$ (e.g., for DTW, $trans(D(L_\mathbf{x}, L_\mathbf{y})) = \sqrt{D(L_\mathbf{x}, L_\mathbf{y})}$; for MSM, $trans(D(L_\mathbf{x}, L_\mathbf{y})) = D(L_\mathbf{x}, L_\mathbf{y})$).

Threshold-based Elastic Measures: To mitigate the disproportionate impact the outlier data points have in DTW distance, a group of elastic measures use a threshold parameter ϵ to decide whether two elements should match or not. Adopting a threshold restricts the relationship between two elements to be either a match or a mismatch, regardless of how close or distant they are.

Table 2: Summary of distances ($dist^D(x_i, y_j)$, $dist^H(x_i, y_j)$, and $dist^V(x_i, y_j)$), transformation functions ($trans(D(L_{\mathbf{x}}, L_{\mathbf{y}}))$, and LBs for threshold-based elastic distances.

LCSS	$dist^D(x_i, y_j)$	$\begin{cases}1 \text{ if } \lvert x_i - y_j \rvert \leq \epsilon \\ 0 \text{ otherwise}\end{cases}$
	$dist^V(x_i, y_j)$	0
	$dist^H(x_i, y_j)$	0
	$trans(D(L_{\mathbf{x}}, L_{\mathbf{y}}))$	1 - $\frac{D(L_{\mathbf{x}}, L_{\mathbf{y}})}{min(L_{\mathbf{x}}, L_{\mathbf{y}})}$
	LB	$\text{UE}(\mathbf{y})_i = max(y_{i-w} : y_{i+w})$ $\text{LE}(\mathbf{y})_i = min(y_{i-w} : y_{i+w})$ $LB\_LCSS(\mathbf{x}, \mathbf{y}) = 1 - \frac{1}{L_{\mathbf{x}}} \sum_{i=1}^{L_{\mathbf{x}}} \begin{cases}1 \text{ if } LE(\mathbf{y})_i \leq x_i \leq UE(\mathbf{y})_i \\ 0 \text{ otherwise}\end{cases}$
EDR	$dist^D(x_i, y_j)$	$\begin{cases}0 \text{ if } \lvert x_i - y_j \rvert \leq \epsilon \\ 1 \text{ otherwise}\end{cases}$
	$dist^V(x_i, y_j)$	1
	$dist^H(x_i, y_j)$	1
	$trans(D(L_{\mathbf{x}}, L_{\mathbf{y}}))$	$D(L_{\mathbf{x}}, L_{\mathbf{y}})$
	LB	No Existing Lower Bound
SWALE	$dist^D(x_i, y_j)$	$\begin{cases}r \text{ if } \lvert x_i - y_j \rvert \leq \epsilon \\ p \text{ otherwise}\end{cases}$
	$dist^V(x_i, y_j)$	p
	$dist^H(x_i, y_j)$	p
	$trans(D(L_{\mathbf{x}}, L_{\mathbf{y}}))$	$D(L_{\mathbf{x}}, L_{\mathbf{y}})$
	LB	No Existing Lower Bound

Such quantization makes threshold-based approaches more robust against outliers and noisy time series than other elastic measures. A match between a new pair of elements indicates a diagonal movement in the accumulated distance matrix D, whereas a mismatch is associated with either a horizontal or vertical movement. Table 2 summarizes distance functions, transformation functions, and LBs for three popular threshold-based elastic distance measures:

- **Longest Common Subsequence (LCSS)** [92] was originally used for pattern matching in text strings and was adapted to measure similarity between time series. LCSS increases similarity by 1 when two elements match and 0 in cases of mismatch. The LCSS distance is the LCSS similarity normalized by the length of the shorter time series.
- **Edit Distance on Real Sequences (EDR)** [20] is an adaption of the edit distance for strings. EDR computes the distance by penalizing mismatches instead of rewarding matches.
- **Sequence Weighted Alignment (SWALE)** [27] parameterizes EDR using a parameter r for a match and a punishment parameter p for mismatch, instead of fixed 1 and 0.

Metric Elastic Measures: The aforementioned elastic measures, including DTW and threshold-based measures, are all non-metric, and thus do not satisfy the triangle inequality [88]. Being a metric enables elastic measures to use generic indexing methods [38, 39, 102] and clustering methods [16, 34, 41] designed for metrics. In addition, in nearest neighbor search, triangle inequality can be applied to efficiently prune comparisons [19]. As summarized in Table 3, there are three popular metric elastic measures:

- **Edit Distance with Real Penalty (ERP)** [19] is similar to DTW as ERP uses the squared difference ($(x_i - y_j)^2$) between two elements as the distance function for diagonal movement. Different from DTW, ERP introduces an additional gap value parameter, g, to compute the distance for horizontal and vertical movements. A drawback of ERP is the inability to handle vertically shifted time series.
- **Move-Split-Merge (MSM)** [88] combines advantages of several elastic measures. MSM is a translation-invariant metric. A constant distance c is associated with *split* (replicating the previous element) and *merge* (merging two identical elements into a single one) operations, which correspond to the horizontal and vertical movements in the distance matrix. The *move* operation is a diagonal movement in the distance matrix, where the distance is $|x_i - y_j|$.
- **Time Warp Edit Distance (TWED)** [63] penalizes the difference in timestamps in addition to the difference in numerical values. TWED penalizes timestamp difference with parameter v in all three types of movements. TWED also employs an additional stiffness parameter λ in horizontal and vertical movements to control warping.

Table 3: Summary of distances ($dist^D(x_i, y_j)$, $dist^H(x_i, y_j)$, and $dist^V(x_i, y_j)$), transformation functions ($trans(D(L_{\mathbf{x}}, L_{\mathbf{y}}))$), and LBs for metric elastic distances.

ERP	$dist^D(x_i, y_j)$	$(x_i - y_j)^2$
	$dist^V(x_i, y_j)$	$(x_i - g)^2$
	$dist^H(x_i, y_j)$	$(y_j - g)^2$
	$trans(D(L_{\mathbf{x}}, L_{\mathbf{y}}))$	$D(L_{\mathbf{x}}, L_{\mathbf{y}})$
	LB	$LB\_Kim - ERP(\mathbf{x}, \mathbf{y}) = \max\begin{cases}\lvert x_1 - y_1 \rvert \\ \lvert x_{L_{\mathbf{x}}} - y_{L_{\mathbf{y}}} \rvert \\ \lvert max(\mathbf{x}) - max(\mathbf{y}) \rvert \\ \lvert min(\mathbf{x}) - min(\mathbf{y}) \rvert\end{cases}$
		$LB\_ERP(\mathbf{x}, \mathbf{y}) = \lvert \sum \mathbf{y} - \sum \mathbf{x} \rvert$
		$UE_i = max(g, max(c_{i-w} : c_{i+w}))$ $LE_i = min(g, min(c_{i-w} : c_{i+w}))$ $LB\_Keogh - ERP(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1} \begin{cases}(y_i - UE_i)^2 \text{ if } y_i > UE_i \\ (y_i - LE_i)^2 \text{ if } y_i < LE_i \\ 0 \text{ otherwise}\end{cases}}$
MSM	$dist^D(x_i, y_j)$	$\lvert x_i - y_i \rvert$
	$dist^V(x_i, y_j)$	$\begin{cases}c \text{ if } x_{i-1} \leq x_i \leq y_j \text{ or } x_{i-1} \geq x_i \geq y_j \\ c + min\begin{cases}\lvert x_i - x_{i-1} \rvert \\ \lvert x_i - y_j \rvert\end{cases} \text{ otherwise}\end{cases}$
	$dist^H(x_i, y_j)$	$\begin{cases}c \text{ if } y_{j-1} \leq y_j \leq x_i \text{ or } y_{j-1} \geq y_j \geq x_i \\ c + min\begin{cases}\lvert y_j - y_{j-1} \rvert \\ \lvert y_j - x_i \rvert\end{cases} \text{ otherwise}\end{cases}$
	$trans(D(L_{\mathbf{x}}, L_{\mathbf{y}}))$	$D(L_{\mathbf{x}}, L_{\mathbf{y}}))$
	LB	$LB\_MSM(\mathbf{x}, \mathbf{y}) = \lvert x_1 - y_1 \rvert + \sum_{i=2}^{L_Y} \begin{cases}min(\lvert y_i - max(\mathbf{x}) \rvert, c) \text{ if } y_{i-1} \geq y_i > max(\mathbf{x}) \\ min(\lvert y_i - min(\mathbf{x}) \rvert, c) \text{ if } y_{i-1} \leq y_i < min(\mathbf{x}) \\ 0 \text{ otherwise}\end{cases}$
TWED	$dist^D(x_i, y_j)$	$(x_i - y_j)^2 + (x_{i-1} - y_{j-1})^2 + v(\lvert t_{x_i} - t_{x_{i-1}} \rvert + \lvert t_{y_j} - t_{y_{j-1}} \rvert)$
	$dist^V(x_i, y_j)$	$(x_i - x_{i-1})^2 + v(\lvert t_{x_i} - t_{x_{i-1}} \rvert) + \lambda$
	$dist^H(x_i, y_j)$	$(y_j - y_{j-1})^2 + v(\lvert t_{y_j} - t_{y_{j-1}} \rvert) + \lambda$
	$trans(D(L_{\mathbf{x}}, L_{\mathbf{y}}))$	$D(L_{\mathbf{x}}, L_{\mathbf{y}})$
	LB	$LB\_TWED(\mathbf{x}, \mathbf{y}) = \min\begin{cases}(x_1 - y_1)^2 \\ x_1^2 + v + \lambda \\ y_1^2 + v + \lambda\end{cases} + \sum_{i=2}^{L_Y} \begin{cases}min(v, (y_i - max(max(\mathbf{x}), y_{i-1}))^2 \\ \text{if } y_i > max(max(\mathbf{x}), y_{i-1}) \\ min(v, (y_i - min(min(\mathbf{x}), y_{i-1}))^2 \\ \text{if } y_i < min(min(\mathbf{x}), Y_{i-1}) \\ 0 \text{ otherwise}\end{cases}$

2.2 Lower Bounds for Elastic Measures

As described in the previous section, all elastic measures compute a distance matrix containing L^2 cells where L is the length of two-time series, resulting in $O(L^2)$ complexity (we ignore the constrained variants as in the worst case they also have similarly high complexity). This runtime complexity is rather time-consuming for similarity search, considering that even for moderate-size databases, millions of such comparisons have to be performed. As a result,

Algorithm 1: LB-Accelerated Nearest Neighbor Search

Require: **y**, a query series
Require: **X**, a set of data time series
Require: L, labels of data time series L
Ensure: $label\_y$, label of query time series
1: **for x** in **X do**
2: $lb\_list[i]$ = LowerBound(**x**, **y**)
3: **end for**
4: $best\_so\_far = +\infty$
5: $ordering$ = sort(lb_list)
6: **X** = **X**[$ordering$]
7: **for x** in **X do**
8: **if** lb_list[x] < best_so_far **then**
9: actual_dist = ElasticMeasure(**x**, **y**)
10: **if** actual_dist < best_so_far **then**
11: $best\_so\_far = actual\_dist$
12: $label\_y = L[\mathbf{x}]$
13: **end if**
14: **end if**
15: **end for**

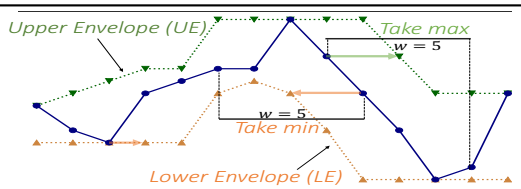


Figure 3: Construction of upper and lower envelopes: the solid line represents the original time series, and two dashed lines represent the upper and lower envelopes, respectively.

similarity search under elastic measures becomes orders of magnitude slower than when ED is used [78]. The most popular approach to accelerate elastic measures is through lower bounding. LB approximates the elastic measure distance without computing the full distance matrix. Specifically, LB uses the approximated value to filter out unpromising candidates in tasks such as nearest neighbor search. Algorithm 1 illustrates how LB is applied to accelerate the 1-NN classification of time series y. First, the list of time series is reordered based on their LB distance to the query time series y (lines 1-6). Then, going through the list, the actual elastic measure distance is only computed for promising time series whose LB distance with query time series y is less than the actual distance between y and the current nearest neighbor (lines 7-9). Finally, if the actual distance is less than the $best\_so\_far$ distance, the $best\_so\_far$ distance and the nearest neighbor are updated (lines 10-12).

Research efforts in lower bounds for elastic measures have concentrated on developing tighter lower bounds of DTW due to its popularity. LB_Kim [49] is a cheap DTW LB with $O(1)$ complexity, which takes the maximum among $\{|X_1 - Y_1|, |X_L - Y_L|, |X_{max} - Y_{max}|, |X_{min} - Y_{min}|\}$. Although LB_Kim is fast to compute, its looseness makes it ineffective in filtering out less obvious candidates. LB_Yi exploits the fact all points in **x** that are either larger than $max(\mathbf{y})$ or smaller than $min(\mathbf{y})$ necessarily contribute to the final DTW distance. Subsequent to LB_Yi, LB_Keogh [48] obtains a much higher tightness than LB_Yi and LB_Kim by utilizing envelopes and a warping window w (Table 4, Equation 4). As shown in Figure 3, LB_Keogh first constructs the upper and lower envelopes of the query time series and computes the distance between these query envelopes and the target (candidate) time series.

LB_Improved [54] computes the ordinary LB_Keogh as well as the LB_Keogh between the query time series and the projection of target time series on the query envelopes (Table 4, Equations 6 and 7). This makes LB_Improved tighter than LB_Keogh but also adds

Table 4: Key existing LBs for DTW.

DTW	$O(1)$ Complexity	$LB\_Kim(\mathbf{x},\mathbf{y}) = \max\begin{cases}\lvert x_1 - y_1\rvert \\ \lvert x_{L_\mathbf{x}} - y_{L_\mathbf{x}}\rvert \\ \lvert max(\mathbf{x}) - max(\mathbf{y})\rvert \\ \lvert min(\mathbf{x}) - min(\mathbf{y})\rvert\end{cases}$ (3)
	Based on envelopes	$\mathrm{UE}(\mathbf{y})_i = max(y_{i-w} : y_{i+w})$ $\mathrm{LE}(\mathbf{y})_i = min(y_{i-w} : y_{i+w})$ $LB\_Keogh(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{L_\mathbf{x}} \begin{cases}(x_i - UE(\mathbf{y})_i)^2 \text{ if } x_i > UE(\mathbf{y})_i \\ (x_i - LE(\mathbf{y})_i)^2 \text{ if } x_i < LE(\mathbf{y})_i \\ 0 \text{ otherwise}\end{cases}}$ (4)
		$LB\_Yi(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{L} \begin{cases}(x_i - max(\mathbf{y}))^2 \text{ if } x_i > max(\mathbf{y}) \\ (x_i - min(\mathbf{y}))^2 \text{ if } x_i < min(\mathbf{y})\end{cases}$ (5)
		$H(\mathbf{x},\mathbf{y})_i = \begin{cases}UE(\mathbf{y})_i \text{ if } x_i \geq UE(\mathbf{y})_i \\ LE(\mathbf{y})_i \text{ if } x_i \leq LE(\mathbf{y})_i \\ x_i \text{ otherwise}\end{cases}$ (6) $LB\_Improved(\mathbf{x},\mathbf{y}) = LB\_Keogh(\mathbf{x},\mathbf{y}) + LB\_Keogh(\mathbf{y}, H(\mathbf{x},\mathbf{y}))$ (7)
	Others	$\delta(x,\mathbf{y}) = min(x - y)^2 \text{ for } y \in \mathbf{y}$(8) $LB\_New(\mathbf{x},\mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_{L_\mathbf{x}} - y_\mathbf{y})^2 + \sum_{i=2}^{L_x - 1} \delta(x_i, \mathbf{y})}$(9)

overhead. In addition, the projection of candidate queries on query envelopes cannot be computed in a pre-processing step. LB_New [85] (Table 4, Equations 8 and 9) also obtains higher pruning power than LB_Keogh. However, instead of constructing envelopes and considering only elements outside of envelopes, LB_New pairs each candidate element, x_i, with the closest query element, y_j, where $y_j \in Y_i = (y_{max(1,j-w)} : y_{min(j+w,L_y)})$. Differently from the previous two approaches, LB_Enhanced [89] constructs alternating bands around the upper-left corner (i.e., Left Bands) and bottom-right corner (i.e., Right Bands), where each band captures one cell in the warping path. The number of bands is decided by a parameter V, which requires a tuning process. LB_Keogh is applied for the remaining portion of the time series not covered by the two bands. LB_Petitjean [97] improves the projection strategy in LB_Improved and incorporates the bands from LB_Enhanced, which translates into increased pruning power. LB_Webb [97] approximates LB_Petitjean without directly computing envelopes or projections, resulting in improved performance over LB_Petitjean. Due to space limitations, we omit the definitions of LB_Enhanced, LB_Petitjean, and LB_Webb in Table 4, which require introducing new notation and concepts not shared with the key existing LBs.

In addition to DTW lower bounds, a few lower bounds have been developed for other elastic measures as well. For example, in efforts to develop ERP LBs, LB_Keogh and LB_Kim were adapted to form the LB_Keogh-ERP and LB_Kim-ERP, respectively. In particular, LB_Keogh-ERP adjusted the original LB_Keogh envelopes by adding another parameter g (Table 3). LB_Kim-ERP also adds a parameter g to LB_Kim to incorporate possible gaps in the alignment. In addition to adaptation from DTW, authors of ERP also developed LB_ERP specifically for ERP. However, LB_ERP is only applicable for $g = 0$, which limits the usability. In addition to adaptation to ERP, LB_Keogh was also modified to LB_LCSS by replacing DTW's Euclidean Distance with reward parameter of 1 and penalty parameter of 0 in cases of match and mismatch. Recently, [90] defined LBs for TWED and MSM based on their distance functions to capture their initial boundary distance and query characteristics (Table 3).

After reviewing the literature, we observe that (i) multiple LBs focus on improving the tightness of LB_Keogh, which does not always translate into higher speed up; and (ii) different distances for different movements make the development of LBs challenging.

3 THE GLB FRAMEWORK

Until now, we have thoroughly reviewed the state-of-the-art elastic measures and their corresponding LB distances. It becomes evident there is merit in developing a framework to ease the process of deriving LBs for all elastic measures. By studying the development of LBs in the past two decades, we identified four critical properties (i.e., $\mathcal{P}1-4$, Section 1) necessary for the effectiveness of LBs. We aim to develop a generalized framework to satisfy all four properties. Next, we introduce GLB, our innovative framework to facilitate the creation of efficient LBs for elastic measures that can be mapped into the generalization of elastic measures described in Section 2.1 (Equation 2). The GLB framework attains its generalizability by introducing a novel abstraction of the various distance functions defined between elements of time series across elastic measures.

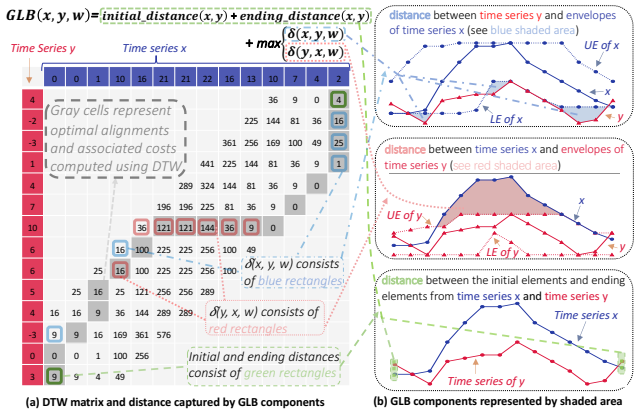
3.1 Main Ideas

GLB aims to satisfy four critical properties (i.e., $\mathcal{P}1-4$): query dependence, data dependence, boundary dependence, and reusability.

Query and Data Dependence: To consider characteristics of both the query time series and the target time series while being cache-friendly, GLB constructs envelopes for each query time series **y** and data time series **x**. Since **y** and **x** have different numerical values and, therefore, different envelopes, the distance between the data time series and the query envelope is different from the distance between the query time series and the data envelope. GLB adaptively selects between the two envelopes to maximize the LB tightness for each pairwise comparison.

Boundary Dependence: One deficiency we observed in the construction of LB_Enhanced is the need for a time-consuming parameter tuning process to decide the appropriate value for its tightness parameter V, which significantly reduces the speedup in the overall process. To build on top of LB_Enhanced and take advantage of the boundary distances, GLB includes only the distances for aligning leading elements, $initial\_distance(x_1, y_1)$, and distances for aligning ending elements, $ending\_distance(x_{L_\mathbf{x}}, y_{L_\mathbf{y}})$. In particular, $initial\_distance(x_1, y_1)$ depends on how each elastic measure computes the distance for the first cell $D(1,1)$ of the accumulated distance matrix. Such distance function varies across each elastic measure; for instance, in DTW and ERP, $initial\_distance(x_1, y_1) = (x_1 - y_1)^2$, while $initial\_distance(x_1, y_1) = |x_1 - y_1|$ in MSM. By abstracting the internal costs of elastic measures, GLB adapts the LB computation to different elastic measures, as we will see later.

Reusability: As introduced in [54], calculating a data-dependent envelope in addition to the query envelope in LB_Keogh allows a tighter lower bound. [54] demonstrated this idea by developing LB_Improved, which computes the query envelope as well as the envelope of a projection time series (see Table 4, Eq. 6). However, we observed that computing the projection time series relies on both the query and data time series, making envelopes of projection time series not reusable. In fact, according to our evaluation across 128 datasets (see Table 7), despite LB_Improved's impressive pruning power of 84.71% (avoided 84.71% of calls to the DTW distance) over LB_Keogh's 70.19%, LB_Improved is only able to reduce runtime by 63%, whereas LB_Keogh could reduce runtime by 76%. The inconsistency between LB_Improved's strong pruning power and less impressive speedup suggests that reusability is a key factor. As a result, reusability is a core design principle of GLB.



(a) DTW matrix and distance captured by GLB components (b) GLB components represented by shaded area

Figure 4: GLB_DTW for two time series with warping window=3. In the distance matrix: the red column represents query time series y; the blue row represents data time series x; gray cells represent the DTW alignment path; blue rectangles are distances between query time series y and envelope of data time series x; red rectangles are distances between data time series x and envelope of query time series y; green rectangles represent the boundary distances.

3.2 Adaptable Distance Functions

In addition to the four aforementioned properties, GLB aims to provide a generalized framework to ease the process of developing new LBs. From our extensive evaluation (see Sections 5 and 6), we observe that it took about two decades to improve the tightness of DTW's LB to over 80% today. Unfortunately, the tightness of the state-of-the-art LBs for other elastic measures, such as for MSM and TWE, are still below 20%. We believe it is unsustainable to expect similar research efforts dedicated to each elastic measure individually. Contrary to the isolated development of different LBs for different elastic measures, we observe that all elastic measures share the same dynamic programming construct, and their only differences lie in the distance functions used to capture movements along the accumulated distance matrix. Thus, for GLB, we aim to leverage this shared structure (see Section 2.1, Equation 2, on how elastic measures can be generalized into the same framework). An important advantage of such a generalized LB is the ability to transfer the research effort devoted to the development of LBs for one elastic measure to the others.

To develop a generalized framework, we need to account for the difference between DTW and other elastic measures. Specifically, when moving horizontally and vertically through the accumulated distance matrix, DTW uses the same distance function as moving diagonally; by contrast, other elastic measures use different distance functions for horizontal, vertical, and diagonal movements. For instance, in ERP, the distance for diagonal movement is $dist^D(x_i - y_j) = (x_i - y_j)^2$, but the distance for horizontal movement is $dist^H(x_i - y_j) = (x_i - g)^2$. Since we are unsure whether each element is involved in a diagonal or horizontal or vertical movement in the optimal alignment, GLB takes the minimum of the three alternatives. Similarly, when calculating the distance of the last cell, $D(L_\mathbf{x}, L_\mathbf{y})$, in the accumulated distance matrix, we need to take the minimum among three alternative movements to $D(L_\mathbf{x}, L_\mathbf{y})$.

3.3 Mathematical Formulation

Having introduced the main ideas behind GLB, we now provide a formal definition. GLB includes the pairwise distances between the first and last pairs of elements of two time series and then the maximum of query and data characteristics captured by query and data envelopes. A warping window can also be applied to GLB in cases when the warping windows are used for computing distances in elastic measures. GLB is formally defined as:

$$\text{GLB}(\mathbf{x}, \mathbf{y}, w) = \text{initial\_distance}(\mathbf{x}, \mathbf{y}) + \text{ending\_distance}(\mathbf{x}, \mathbf{y}) + \max\begin{cases}\delta(\mathbf{x}, \mathbf{y}, w)\\ \delta(\mathbf{y}, \mathbf{x}, w)\end{cases}$$

initial_distance and *ending_distance* are incorporated in the GLB framework so that GLB improves the tightness of LBs developed within the GLB framework (See Boundary Dependence in Section 3.1). *initial_distance* is the distance between the first elements from two time series as defined by the elastic measure. For instance, in the case of DTW, the distance between the first element of time series $\mathbf{x}$ and $\mathbf{y}$ is $(x_1 - y_1)^2$, so $initial\_distance(x_1, y_1) = (x_1 - y_1)^2$ for DTW. *ending_distance* represents the distance function for alignment of the last elements of two time series. *ending_distance* takes the minimum of the three movements:

$$ending\_distance(\mathbf{x}, \mathbf{y}) = min\begin{cases}dist^D(x_{L_\mathbf{x}}, y_{L_\mathbf{y}})\\ dist^V(x_{L_{\mathbf{x}-1}}, y_{L_\mathbf{y}})\\ dist^H(x_{L_\mathbf{x}}, y_{L_{\mathbf{y}-1}})\end{cases} \tag{10}$$

where $dist^D(x_i, y_i)$, $dist^H(x_i, y_i)$, $dist^V(x_i, y_i)$ represent distance functions for diagonal, horizontal, and vertical movements. For instance, in the case of ERP with parameter $g = 0$, there are three possible distances from the alignment between the ending element of time series $\mathbf{x}$ and $\mathbf{y}$: $\{(x_{L_\mathbf{x}} - y_{L_\mathbf{y}})^2, (x_{L_\mathbf{x}} - g)^2, (y_{L_\mathbf{y}} - g)^2\}$. Then the *ending_distance* in GLB_ERP is $min(\{(x_{L_\mathbf{x}} - y_{L_\mathbf{y}})^2, (x_{L_\mathbf{x}} - g)^2, (y_{L_\mathbf{y}} - g)^2\})$. $\delta(\mathbf{x}, \mathbf{y}, w)$ and $\delta(\mathbf{y}, \mathbf{x}, w)$ illustrate how GLB computes query and data envelopes and capture the distance between the envelopes and the other time series. In the calculation of such distance, the abstracted distance functions $\delta(\mathbf{x}, \mathbf{y}, w)$ and $\delta(\mathbf{y}, \mathbf{x}, w)$ were utilized to allow for the use of any elastic measure's custom distance functions. This is a departure from previous LBs in the literature, which required a different LB for each elastic measure. Instead, by specifying the distance functions ($dist^D(x_i, y_j)$, $dist^V(x_i, x_{i-1})$, $dist^H(y_i, y_{i-1})$), each elastic measure can calculate its own LB within the GLB framework. This approach offers greater flexibility and adaptability in computing the distance between time series. $\delta(\mathbf{x}, \mathbf{y}, w)$ and $\delta(\mathbf{y}, \mathbf{x}, w)$ are defined as:

$$\delta(\mathbf{x}, \mathbf{y}, w) = \sum_{j=2}^{L_\mathbf{y}-1}\begin{cases}min(dist^D(y_j, UE(\mathbf{x})_j), dist^H(y_j, y_{j-1})) \text{ for } y_j \geq UE(\mathbf{x})_j\\ min(dist^D(y_j, LE(\mathbf{y})_j), dist^H(y_j, y_{j-1})) \text{ for } y_j \leq LE(\mathbf{x})_j\\ 0 \text{ otherwise}\end{cases}$$

where $UE(\mathbf{x})_j = max(x_{max(1,j-w)} : x_{min(L_\mathbf{x}, j+w)})$
and $LE(\mathbf{x})_j = min(x_{max(1,j-w)} : x_{min(L_\mathbf{x}, j+w)})$, and w is the window.

$$\delta(\mathbf{y}, \mathbf{x}, w) = \sum_{i=2}^{L_\mathbf{x}-1}\begin{cases}min(dist^D(x_i, UE(\mathbf{y})_i), dist^V(x_i, x_{i-1})) \text{ for } x_i \geq UE(\mathbf{y})_i\\ min(dist^D(x_i, LE(\mathbf{y})_i), dist^V(x_i, x_{i-1})) \text{ for } x_i \leq LE(\mathbf{y})_i\\ 0 \text{ otherwise}\end{cases}$$

where $UE(\mathbf{y})_i = max(y_{max(1,i-w)} : y_{min(L_\mathbf{y}, i+w)})$
and $LE(\mathbf{y})_i = min(y_{max(1,i-w)} : y_{min(L_\mathbf{y}, i+w)})$, and window is w.

An implementation of GLB applied to accelerate the nearest neighbor search is provided in Algorithm 2. Firstly, envelopes of query and data time series are computed and stored in a cache-friendly approach (lines 1-11, 21-22) and serve as inputs to compute GLB. The GLB function computes and returns the sum of boundary distances and delta functions (lines 13-19). Finally, the GLB distances are used to accelerate the 1-NN Search (lines 21-41).

3.4 Proof of Lower Bounding Property

Having formally introduced GLB, in this section, we provide proof of the LB property of GLB. Due to this property, GLB ensures its correctness and no false positives exist (i.e., ignored comparisons with time series are correctly avoided), which ensures the utility of elastic measures in practice. For example, when GLB is used in classification tasks, the classification accuracy remains unaffected when LBs are applied. The LB property is formally defined as follows:

For any two time series $\boldsymbol{x}$ and $\boldsymbol{y}$ of length $L_\boldsymbol{x}$ and $L_\boldsymbol{y}$ respectively, an optimal alignment path $W = w_1, w_2, ..., w_p$ where $w_k = (i, j)$ indicates x_i is aligned to y_j, following inequality holds: $GLB\_EE(\boldsymbol{x}, \boldsymbol{y}) \leq EE(\boldsymbol{x}, \boldsymbol{y}))$ where EE represents an aforementioned elastic measure.

Proof Firstly, we observe the actual distance between two time series, $EE(\mathbf{x}, \mathbf{y})$, can be generally defined as:

$$\text{EE}(\mathbf{x}, \mathbf{y}) = \text{initial\_distance}(x_1, y_1) + \sum_{k=2}^{P-1} dist(X_{W_k{}^1}, Y_{W_k{}^2}) + min\begin{cases}dist^D(x_{L_x}, y_{L_y})\\ dist^V(x_{L_\mathbf{x}-1}, y_{L_\mathbf{y}})\\ dist^H(x_{L_\mathbf{x}}, y_{L_{\mathbf{x}-1}})\end{cases}$$

Since the first and last components of $EE(\mathbf{x}, \mathbf{y})$ are exactly the same as the first and second term of GLB, proving $GLB\_EE(\mathbf{x}, \mathbf{y}) \leq EE(\mathbf{x}, \mathbf{y})$ is the same as proving:

$$max\begin{cases}\delta(\mathbf{x}, \mathbf{y}, w)\\ \delta(\mathbf{y}, \mathbf{x}, w)\end{cases} \leq \sum_{k=2}^{P-1} dist(\mathbf{x}_{W_k{}^1}, \mathbf{y}_{W_k{}^2}) \tag{11}$$

where $\delta(\mathbf{x}, \mathbf{y}, w)$ and $\delta(\mathbf{y}, \mathbf{x}, w)$ are defined in Section 3.3. To prove this statement, we first show $\delta(\mathbf{x}, \mathbf{y}, w) \leq \sum_{k=2}^{P-1} dist(\mathbf{x}_{W_k{}^1}, \mathbf{y}_{W_k{}^2})$ and then $\delta(\mathbf{y}, \mathbf{x}, w) \leq \sum_{k=2}^{P-1} dist(\mathbf{x}_{W_k{}^1}, \mathbf{y}_{W_k{}^2})$. Suppose the optimal alignment for x_i is y_j*, following must hold: if $x_i > max(\mathbf{y})$, then $dist^D(x_i, y_j*) \geq dist^D(x_i, max(\mathbf{y}))$; if $x_i < min(\mathbf{y})$, then $dist^D(x_i, y_j*) \geq dist^D(x_i, min(\mathbf{y}))$. Thus, regardless of whether the alignment distance for particular x is produced by distance function $dist^D$ or $dist^V$ or $dist^H$ (depending on whether the optimal alignment path arrives at that cell diagonally or horizontally/vertically), we have:

$$\text{dist}(\mathbf{x}_{W_k{}^1}, \mathbf{y}_{W_k{}^2}) \geq min(dist^D(y_j, UE(\mathbf{x})_j), dist^H(y_j, y_{j-1})) \text{ for } y_j \geq UE(\mathbf{x})_j$$
$$\text{dist}(\mathbf{x}_{W_k{}^1}, \mathbf{y}_{W_k{}^2}) \geq min(dist^D(y_j, LE(\mathbf{x})_j), dist^H(y_j, y_{j-1})) \text{ for } y_j \leq LE(\mathbf{x})_j$$

which indicates that $\delta(\mathbf{x}, \mathbf{y}, w) \leq \sum_{k=2}^{P-1} dist(\mathbf{x}_{W_k{}^1}, \mathbf{y}_{W_k{}^2})$. The same reasoning applies to $\delta(\mathbf{y}, \mathbf{x}, w) \leq \sum_{k=2}^{P-1} dist(\mathbf{x}_{W_k{}^1}, \mathbf{y}_{W_k{}^2})$, we know:

$$\max\begin{cases}\delta(\mathbf{x}, \mathbf{y}, w)\\ \delta(\mathbf{y}, \mathbf{x}, w)\end{cases} \leq \sum_{k=2}^{P-1} dist(\mathbf{x}_{W_k{}^1}, \mathbf{y}_{W_k{}^2}).$$

4 NEW LOWER BOUNDS BASED ON GLB

In this section, we present the new LBs based on GLB. First, we present the design and formal definition of GLB_DTW, the GLB variant for DTW (Section 4.1); then, we showcase the formal definition of GLB variants for other elastic measures (Section 4.2).

4.1 GLB for Dynamic Time Warping

Improvements of GLB_DTW over LB_Keogh, are two-fold: (1) GLB_DTW incorporates the distances for aligning the first and

Algorithm 2: Nearest Neighbor Search with GLB

Require: $\boldsymbol{X}$ is a $n \times L_{\mathbf{x}}$ matrix of n data time series
Require: L is an $n \times 1$ vector with labels of data time series
Require: $\mathbf{Y}$ is an $m \times L_{\mathbf{x}}$ matrix of m query time series
Require: w is the warping window
Ensure: $query\_label$ is an $m \times 1$ vector containing labels of the m time series in $\mathbf{Y}$

function make_envelopes($\mathbf{X}$, w)
  UE = []
  LE = []
  for $i = 1$ **to** Number_of_Rows($\boldsymbol{X}$) **do**
    for $j = 1$ **to** Number_of_Columns($\boldsymbol{X}$) **do**
      $UE[i, j]$= max($\boldsymbol{X}[i, j - w] : \boldsymbol{X}[i, j + w]$)
      $LE[i, j]$= min($\boldsymbol{X}[i]_{j-w} : \boldsymbol{X}[i]_{j+w}$)
    end for
  end for
  return UE, LE
end function

function GLB($\mathbf{x}$, $\mathbf{y}$, **yue**, **xle**, **yue**, **yle**)
  $boundary\_distance$ = initial_distance($\mathbf{x}$, $\mathbf{y}$) + ending_distance($\mathbf{x}$, $\mathbf{y}$)
  $query\_distance$ = delta(x, yue, yle)
  $data\_distance$ = delta(y, xue, xle)
  $GLB\_dist = boundary\_distance$ + max($query\_distance, data\_distance$)
  return GLB_dist
end function

XUE, XLE = make_envelopes(X, window)
YUE, YLE = make_envelopes(Y, window)
$query\_class$ = []
for $i = 1$**to** m **do**
  $best\_so\_far = \infty$
  lb_list = []
  for $j = 1$**to** n **do**
    $lb\_list_j = GLB(X_i, Y_j, XUE_i, XLE_i, YUE_j, YLE_j)$
  end for
  $ordering = sort(lb\_list)$
  $X = X[ordering]$
  for $j = 1$**to** n **do**
    if $lb\_dist < best\_so\_far$ **then**
      $actual\_dist = ElasticMeasure(X_j, Y_i)$
      if $actual\_dist < best\_so\_far$ **then**
        $best\_so\_far = actual\_dist$
        $query\_class_i = L_j$
      end if
    end if
  end for
end for

last elements from two sequences (the first two terms of Equation 14), whereas LB_Keogh will only include these distances if the first and/or last element of the candidate series is outside of envelopes of query series; (2) whereas LB_Keogh only computes envelopes for query series, GLB_DTW computes envelopes for both data series (Equation 12) and query series (Equation 13), finds the distances arising from their respective distance to the other time series and then takes the maximum between the two.

$$\delta_{DTW}(\mathbf{x}, \mathbf{y}, w) = \sum_{j=2}^{L_{\mathbf{y}}-1} \begin{cases} (y_j - UE(\mathbf{x})_j)^2 \text{ for } y_j \geq UE(\mathbf{x})_j \\ (y_j - LE(\mathbf{x})_j)^2 \text{ for } y_j \leq LE(\mathbf{x})_j \\ 0 \text{ otherwise} \end{cases} \tag{12}$$

$$\delta_{DTW}(\mathbf{y}, \mathbf{x}, w) = \sum_{i=2}^{L_{\mathbf{x}}-1} \begin{cases} (x_i - UE(\mathbf{y})_i)^2 \text{ for } x_i \geq UE(\mathbf{y})_i \\ (x_i - LE(\mathbf{y})_i)^2 \text{ for } x_i \leq LE(\mathbf{y})_i \\ 0 \text{ otherwise} \end{cases} \tag{13}$$

Figure 4 shows the LB distance captured by LB_Keogh (Figure 4(b-1)) and GLB_DTW. We observe that by computing envelopes for the data time series and by considering initial and ending boundary distances, GLB_DTW improves the tightness of the LB (i.e., includes more distances than LB_Keogh). Formally defined in Equation 14, GLB_DTW consists of distances captured by a boundary distance function, query envelope, and data envelope:

$$GLB\_DTW = \sqrt{(x_i - y_i)^2 + (x_{L_{\mathbf{x}}} - y_{L_{\mathbf{y}}})^2 + max \begin{cases} \delta_{DTW}(\mathbf{y}, \mathbf{x}, w) \\ \delta_{DTW}(\mathbf{x}, \mathbf{y}, w) \end{cases}} \tag{14}$$

Table 5: Summary of GLB variants for ERP, MSM, TWED, LCSS, EDR, and SWALE.

ERP	Boundaries	$(x_i - y_j)^2$
	UE	$\overline{UE}(\mathbf{x})_j = max(x_{max(1,j-w)} : x_{min(L_{\mathbf{x}}, j+w)}); UE(\mathbf{y})_i = max(y_{max(1,i-w)} : y_{min(L_{\mathbf{y}}, i+w)})$
	LE	$LE(\mathbf{x})_j = min(x_{max(1,j-w)} : x_{min(L_{\mathbf{x}}, j+w)}); LE(\mathbf{y})_i = min(y_{max(1,i-w)} : y_{min(L_{\mathbf{y}}, i+w)})$
	$\delta(\mathbf{y}, \mathbf{x}, w)$	$\sum_{i=2}^{L_{\mathbf{x}}-1} \begin{cases} min((x_i - UE(\mathbf{y})_i)^2, (x_i - g)^2) \text{ for } x_i \geq UE(\mathbf{y})_i \\ min((x_i - LE(\mathbf{y})_i)^2, (x_i - g)^2) \text{ for } x_i \leq LE(\mathbf{y})_i \\ 0 \text{ otherwise} \end{cases}$
	$\delta(\mathbf{x}, \mathbf{y}, w)$	$\sum_{j=2}^{L_{\mathbf{y}}-1} \begin{cases} min((y_j - UE(\mathbf{x})_j)^2, (y_j - g)^2) \text{ for } y_j \geq UE(\mathbf{x})_j \\ min((y_j - LE(\mathbf{x})_j)^2, (y_j - g)^2) \text{ for } y_j \leq LE(\mathbf{x})_j \\ 0 \text{ otherwise} \end{cases}$
MSM	Boundaries	$\lvert x_1 - y_1 \rvert + C(x_{L_{\mathbf{x}}}, x_{L_{\mathbf{x}}-1}, y_{L_{\mathbf{y}}}, y_{L_{\mathbf{y}}-1}) where$ $C(x_{L_{\mathbf{x}}}, x_{L_{\mathbf{x}}-1}, y_{L_{\mathbf{y}}}, y_{L_{\mathbf{y}}-1}) = min \begin{cases} min(\lvert x_{LX} - y_{LY} \rvert, c) \text{ if } \begin{cases} x_{L_{\mathbf{x}}-1} \geq x_{L_{\mathbf{x}}} \geq y_{L_{\mathbf{y}}} \\ x_{L_{\mathbf{x}}-1} \leq x_{L_{\mathbf{x}}} \leq y_{L_{\mathbf{y}}} \\ y_{L_{\mathbf{y}}-1} \geq y_{L_{\mathbf{y}}} \geq x_{L_{\mathbf{x}}} \\ y_{L_{\mathbf{y}}-1} \leq y_{L_{\mathbf{y}}} \leq x_{L_{\mathbf{x}}} \end{cases} \\ min \begin{cases} \lvert x_{L_{\mathbf{x}}} - y_{L_{\mathbf{y}}} \rvert \\ c + \lvert x_{L_{\mathbf{x}}} - x_{L_{\mathbf{x}}-1} \rvert \\ c + \lvert y_{L_{\mathbf{y}}} - y_{L_{\mathbf{y}}-1} \rvert \end{cases} \text{ otherwise} \end{cases}$
	UE	$\overline{UE}(\mathbf{x})_j = max(x_{max(1,j-w)} : x_{min(L_{\mathbf{x}}, j+w)}); \overline{UE}(\mathbf{y})_i = max(y_{max(1,i-w)} : y_{min(L_{\mathbf{y}}, i+w)})$
	LE	$LE(\mathbf{x})_j = min(x_{max(1,j-w)} : x_{min(L_{\mathbf{x}}, j+w)}); LE(\mathbf{y})_i = min(y_{max(1,i-w)} : y_{min(L_{\mathbf{y}}, i+w)})$
	$\delta(\mathbf{y}, \mathbf{x}, w)$	$\sum_{i=2}^{L_{\mathbf{x}}-1} \begin{cases} min(\lvert x_i - UE(\mathbf{y})_i \rvert, c) \text{ for } x_i \geq UE(\mathbf{y})_i \\ min(\lvert x_i - LE(\mathbf{y})_i \rvert, c) \text{ for } x_i \leq LE(\mathbf{y})_i \\ 0 \text{ otherwise} \end{cases}$
	$\delta(\mathbf{x}, \mathbf{y}, w)$	$\sum_{j=2}^{L_{\mathbf{y}}-1} \begin{cases} min(\lvert y_j - UE(\mathbf{x})_j \rvert, c) \text{ for } y_j \geq UE(\mathbf{x})_j \\ min(\lvert y_j - LE(\mathbf{x})_j \rvert, c) \text{ for } y_j \leq LE(\mathbf{x})_j \\ 0 \text{ otherwise} \end{cases}$
TWED	Boundaries	$(x_1 - y_1)^2 + min \begin{cases} (x_{L_{\mathbf{x}}} - y_{L_{\mathbf{y}}})^2 + (x_{L_{\mathbf{x}}-1} - y_{L_{\mathbf{y}}-1})^2 \\ (x_{L_{\mathbf{x}}} - x_{L_{\mathbf{x}}-1})^2 + \lambda \\ (y_{L_{\mathbf{y}}} - y_{L_{\mathbf{y}}-1})^2 + \lambda \end{cases}$
	UE	$\overline{UE}(\mathbf{x})_j = max(x_{max(1,j-w)} : x_{min(L_{\mathbf{x}}, j+w)}); \overline{UE}(\mathbf{y})_i = max(y_{max(1,i-w)} : y_{min(L_{\mathbf{y}}, i+w)})$
	LE	$LE(\mathbf{x})_j = min(x_{max(1,j-w)} : x_{min(L_{\mathbf{x}}, j+w)}); LE(\mathbf{y})_i = min(y_{max(1,i-w)} : y_{min(L_{\mathbf{y}}, i+w)})$
	$\delta(\mathbf{y}, \mathbf{x}, w)$	$\sum_{i=2}^{L_X-1} \begin{cases} min((x_i - UE(\mathbf{y})_i)^2 + (x_{j-1} - UE(\mathbf{y})_{i-1})^2) \\ \text{for } x_i \geq UE(\mathbf{y})_i, x_{i-1} \geq UE(\mathbf{y})_i \\ min((x_i - LE(\mathbf{y})_i)^2 + (x_{i-1} - LE(\mathbf{y})_{i-1})^2) \\ \text{for } x_i \leq LE(\mathbf{y})_i, x_{i-1} \leq LE(\mathbf{y})_i \\ 0 \text{ otherwise} \end{cases}$
	$\delta(\mathbf{x}, \mathbf{y}, w)$	$\sum_{j=2}^{L_Y-1} \begin{cases} min((y_j - UE(\mathbf{x})_j)^2 + (y_{j-1} - UE(\mathbf{x})_{j-1})^2) \\ \text{for } y_j \geq UE(\mathbf{x})_j, y_{j-1} \geq UE(\mathbf{x})_j \\ min((y_j - LE(\mathbf{x})_j)^2 + (y_{j-1} - LE(\mathbf{x})_{j-1})^2) \\ \text{for } y_j \leq LE(\mathbf{x})_j, y_{j-1} \leq LE(\mathbf{x})_j \\ 0 \text{ otherwise} \end{cases}$
LCSS	Boundaries	$M_{LCSS}(x_1, y_1, \epsilon) + M_{LCSS}(x_{L_{\mathbf{x}}}, y_{L_{\mathbf{y}}}, \epsilon) where$ $M_{LCSS}(A, B, \epsilon) = \begin{cases} 1 \text{ if } \lvert A - B \rvert \leq \epsilon \\ 0 \text{ otherwise} \end{cases}$
	UE	$\overline{UE}(\mathbf{x})_j = max(x_{max(1,j-w)} : x_{min(L_{\mathbf{x}}, j+w)}); \overline{UE}(\mathbf{y})_i = max(y_{max(1,i-w)} : y_{min(L_{\mathbf{y}}, i+w)})$
	LE	$LE(\mathbf{x})_j = min(x_{max(1,j-w)} : x_{min(L_{\mathbf{x}}, j+w)}); LE(\mathbf{y})_i = min(y_{max(1,i-w)} : y_{min(L_{\mathbf{y}}, i+w)})$
	$\delta(\mathbf{y}, \mathbf{x}, w)$	$\sum_{i=2}^{L_X-1} \begin{cases} M_{LCSS}(x_i, UE(\mathbf{y})_i) \text{ for } x_i \geq UE(\mathbf{y})_i \\ M_{LCSS}(x_i, LE(\mathbf{y})_i) \text{ for } x_i \leq LE(\mathbf{y})_i \\ 0 \text{ otherwise} \end{cases}$
	$\delta(\mathbf{x}, \mathbf{y}, w)$	$\sum_{j=2}^{L_Y-1} \begin{cases} M_{LCSS}(y_j, UE(\mathbf{x})_j) \text{ for } y_j \geq UE(\mathbf{x})_j \\ M_{LCSS}(y_j, LE(\mathbf{x})_j) \text{ for } y_j \leq LE(\mathbf{x})_j \\ 0 \text{ otherwise} \end{cases}$
GLB Variants for Elastic Measures without Existing Lower Bounds		
EDR	Boundaries	$0 + \min \begin{cases} 1 \\ M_{EDR}(x_{L_{\mathbf{x}}}, y_{L_{\mathbf{y}}}, \epsilon) \end{cases}$, where $M_{EDR}(A, B, \epsilon) = \begin{cases} 0 \text{ if } \lvert A - B \rvert \leq \epsilon \\ 1 \text{ otherwise} \end{cases}$
	UE	$\overline{UE}(\mathbf{x})_j = max(x_{max(1,j-w)} : x_{min(L_{\mathbf{x}}, j+w)}); \overline{UE}(\mathbf{y})_i = max(y_{max(1,i-w)} : y_{min(L_{\mathbf{y}}, i+w)})$
	LE	$LE(\mathbf{x})_j = min(x_{max(1,j-w)} : x_{min(L_{\mathbf{x}}, j+w)}); LE(\mathbf{y})_i = min(y_{max(1,i-w)} : y_{min(L_{\mathbf{y}}, i+w)})$
	$\delta(\mathbf{x}, \mathbf{y}, w)$	$\sum_{j=2}^{L_Y-1} \begin{cases} M_{EDR}(y_j, UE(\mathbf{x})_j, \epsilon) \text{ for } y_j \geq UE(\mathbf{x})_j \\ M_{EDR}(y_j, LE(\mathbf{x})_j, \epsilon) \text{ for } y_j \leq LE(\mathbf{x})_j \\ 0 \text{ otherwise} \end{cases}$
	$\delta(\mathbf{y}, \mathbf{x}, w)$	$\sum_{i=2}^{L_X-1} \begin{cases} M_{EDR}(x_j, UE(\mathbf{y})_i, \epsilon) \text{ for } x_j = i \geq UE(\mathbf{y})_i \\ M_{EDR}(x_i, LE(\mathbf{y})_i, \epsilon) \text{ for } x_i \leq LE(\mathbf{y})_i \\ 0 \text{ otherwise} \end{cases}$
SWALE	Boundaries	$M_{SWALE}(x_{L_{\mathbf{x}}}, y_{L_{\mathbf{y}}}, \epsilon)$, where $M_{SWALE}(A, B, \epsilon) = \begin{cases} r \text{ if } \lvert A - B \rvert \leq \epsilon \\ p \text{ otherwise} \end{cases}$
	UE	$\overline{UE}(\mathbf{x})_j = max(x_{max(1,j-w)} : x_{min(L_{\mathbf{x}}, j+w)}); \overline{UE}(\mathbf{y})_i = max(y_{max(1,i-w)} : y_{min(L_{\mathbf{y}}, i+w)})$
	LE	$LE(\mathbf{x})_j = min(x_{max(1,j-w)} : x_{min(L_{\mathbf{x}}, j+w)}); LE(\mathbf{y})_i = min(y_{max(1,i-w)} : y_{min(L_{\mathbf{y}}, i+w)})$
	$\delta(\mathbf{y}, \mathbf{x}, w)$	$\sum_{j=2}^{L_Y-1} \begin{cases} M_{SWALE}(y_j, UE(\mathbf{x})_j, \epsilon) \text{ for } y_j \geq UE(\mathbf{x})_j \\ M_{SWALE}(y_j, LE(\mathbf{x})_j, \epsilon) \text{ for } y_j \leq LE(\mathbf{x})_j \\ 0 \text{ otherwise} \end{cases}$
	$\delta(\mathbf{x}, \mathbf{y}, w)$	$\sum_{i=2}^{L_X-1} \begin{cases} M_{SWALE}(x_j, UE(\mathbf{y})_i, \epsilon) \text{ for } x_j = i \geq UE(\mathbf{y})_i \\ M_{SWALE}(x_i, LE(\mathbf{y})_i, \epsilon) \text{ for } x_i \leq LE(\mathbf{y})_i \\ 0 \text{ otherwise} \end{cases}$

4.2 GLB Variants of Other Elastic Measures

Having introduced GLB for DTW, we now focus on six alternative elastic measures. Table 5 summarizes all LBs produced by GLB.

4.2.1 Edit Distance with Real Penalty (ERP). Similar to GLB_DTW, GLB_ERP uses $(x_i - y_j)^2$ as $initial\_distance(\mathbf{x}, \mathbf{y})$. On the other hand, as shown in Table 4, while DTW has the same distance functions for diagonal and horizontal/vertical movements, ERP has $(x_i - g)^2$ or $(y_j - g)^2$ for horizontal/vertical movements, but $(x_i - y_j)^2$ for diagonal movements; having different distance functions results in two differences of GLB_ERP from GLB_DTW: (1) GLB_ERP takes the minimum of the two possible distance functions in the construction of $\delta(\mathbf{y}, \mathbf{x}, w)$ and $\delta(\mathbf{x}, \mathbf{y}, w)$, as shown in Table 5 (2) GLB_ERP takes the minimum of possible distance functions when computing the ending boundary distance.

4.2.2 Longest Common Subsequence (LCSS). As a threshold-based elastic measure, LCSS employs a matching function M_{LCSS} (See Table 2) to determine if two elements are close enough to be "matched". GLB_LCSS uses the same matching function M_{LCSS} to compute boundary distances by checking if the two initial elements and two ending elements from two time series "match" with each other (see Table 5). In constructing $\delta(\mathbf{y}, \mathbf{x}, w)$ and $\delta(\mathbf{x}, \mathbf{y}, w)$ (see Table 5), the same matching function M_{LCSS} is used to compute distances arising from the distance between one time series and envelopes of the other time series. A unique feature of LCSS is that LCSS first calculates similarity and computes distance based on similarity; accordingly, GLB_LCSS also calculates similarity ($LCSS\_Sim$) before converting similarity to distance (see Table 5).

4.2.3 Move-Split-Merge (MSM). As shown in Table 3, the alignment distance between each pair of elements x_i and y_i using MSM depends on both distance between x_i and y_i and x_{i-1} and y_{i-1}, GLB_MSM defines a new distance function C to capture the distance for ending boundary alignment (see Table 5). Since the distance associated with each movement to the next cell in the distance matrix can be either $|x_i - y_i|$ or c, GLB_MSM takes the minimum of the two to ensure the distance is less than the actual distance.

4.2.4 Time Warp Edit Distance (TWED). Since the distance function for diagonal movements includes pairwise comparison between x_i and y_j as well as x_{i-1} and y_{j-1} in TWED, GLB_TWED only captures the distance associated with cases when two consecutive elements both fall outside of envelopes. Since there are three possible distances associated with ending boundary alignment, GLB_TWED takes the minimum among the three alternatives.

4.2.5 EDR. Similar to LCSS, EDR is a threshold-based elastic measure that employs a match function M_{EDR} (see Table 5) to decide if two elements mismatch and add to distance between time series. As shown in Table 5, GLB_EDR adopts the same M_{EDR} in defining the distance when computing $\delta(\mathbf{y}, \mathbf{x}, w)$ and $\delta(\mathbf{x}, \mathbf{y}, w)$.

4.2.6 SWALE. Similar to LCSS and EDR, SWALE is a threshold-based elastic measure that employs a match function M_{EDR}, with the difference being SWALE parameterizes the penalty for a mismatch and the reward for a match with p and r respectively, instead of using 1 and 0 for match and mismatch. GLB_SWALE uses the same matching function to compute the initial and ending boundary distances, as well as in $\delta(\mathbf{x}, \mathbf{y}, w)$ and $\delta(\mathbf{y}, \mathbf{x}, w)$ to compute the distance between one time series and envelopes of the other.

5 EXPERIMENT SETTINGS

In this section, we report our experimental settings. We aim to provide the first comprehensive study of LBs for a diverse set of elastic measures. We aim to show in each case how GLB improves the current state of the art. Therefore, first, we evaluate the performance of GLB against state-of-the-art LBs for DTW, which is the most widely used elastic measure. Then, we compare GLB with state-of-the-art LBs of alternative elastic measures. Finally, for elastic measures without known LBs, we present the performance of GLB, which could serve as the baseline for further research.

Datasets: Our experiments are performed on 128 datasets from the UCR archive [26]. The UCR archive includes datasets from various domains and is the largest public collection of labeled time series datasets. Datasets are normalized and split into training and test sets. For datasets with varying lengths and missing values, we resort to pre-processed versions in [70], which used standardized resampling and interpolation methods to fix these issues.

Platforms: We ran our experiments on a server with the following configuration: Dual Intel(R) Xeon(R) Silver 4116 (12-core with 2-way SMT), 2.10 GHz, 196GB RAM. The server ran Ubuntu 18.04.3 LTS (64-bit) with Python 3.7.5, Numba 0.53.1, and GCC 8.4.0 compiler.

Implementations: We implemented all methods in Python for consistency. We employ Numba [53], which translates Python functions to optimized machine code at runtime to accelerate the computation of lower bounds and elastic measures. For reproducibility purposes, we make our source code available.[2]

Choice of Parameters: We choose parameters that demonstrated optimal performance in empirical evaluation [78] or recommended by authors who proposed these elastic measures (Table 6). For the choice of warping window, w, searching for the optimal window size would require a laborious process [90]. For consistency with previous evaluation efforts (see references in Table 1), we use a window size of 5% for all lower bounds and elastic measures. This window size has also achieved state-of-the-art 1-NN classification accuracy [74, 78] for unsupervised settings.

Baselines: We compare GLB variants against state-of-the-art LBs for elastic time series distances. For DTW, we compare GLB_DTW against the following strong LBs:

- **LB_Yi**: a simple LB that compares every element of the data with the min and max values of the query time series
- **LB_Kim**: a boundary-dependent LB with $O(1)$ complexity
- **LB_Keogh**: the LB that introduced query envelopes
- **LB_New**: a tight LB with boundary dependence
- **LB_Improved**: a LB that builds on LB_Keogh and exploits data time series characteristics to increase its tightness

We focus on key existing LBs for DTW (as identified in Section 2, Table 4) and omit comparisons with variants that combine techniques or introduce adaptations to trade off pruning power for efficiency. These combinations or adaptations are often orthogonal contributions that can be incorporated into GLB as well with similar effects. Instead, we include comparisons with state-of-the-art LBs of 4 additional elastic measures, namely, ERP, MSM, TWED, and LCSS, overall 11 LBs spanning 5 elastic measures (see Table 1 for references and Tables 2, 3, and 4 for their exact mathematical formulations). More specifically, we omitted LB_Enhanced [89] because it requires a time-consuming parameter tuning process and, therefore, it is not directly comparable. Similarly, we omitted LB_Petitjean and LB_Webb, which are variants of LB_Improved.

[2] www.timeseries.org/glb

Table 6: Parameters for all elastic measures evaluated.

Elastic Measures	Parameters
Dynamic Time Warping (DTW)	-
Edit Distance on Real Sequences (EDR)	$\epsilon = 0.1$
Longest Common Subsequence (LCSS)	$\epsilon = 0.2$
Sequence Weighted Alignment (SWALE)	$\epsilon = 0.2, p = 5, r = 1$
Edit Distance with Real Penalty (ERP)	$gap = 0$
Move-Split-Merge (MSM)	$c = 0.5$
Time Warp Edit Distance (TWE)	$\lambda = 1, \nu = 0.0001$

Tighter lower bounds, like LB_Improved and LB_New, typically involve higher costs, resulting in poor speedup relative to their impressive pruning power. Therefore, we evaluate LBs in isolation but also, as noted in [54], in a cascade, where the expensive LBs are only computed for the cases that cannot be pruned by cheaper LBs. To take advantage of such property, we evaluate the performance of LB_Improved and LB_New running together with LB_Keogh.

Metrics: We compare the effectiveness lower bounds on (1) **pruning power** and (2) **speedup**. Pruning power is the percentage of the true distance computation avoided in the 1-NN search due to adopting a lower bound. Speedup is calculated by dividing the runtime of traditional 1-NN search by the runtime of 1-NN search using a lower bound. We compute all reusable components of lower bounds as a pre-processing step to avoid repetitive calculation and include pre-processing time in our runtime measurement.

6 EXPERIMENTAL RESULTS

In this section, we report the results of our experiments. First, we evaluate GLB_DTW against existing lower bounds for DTW, which is the most widely used and studied elastic measure (Section 6.1). Then, we compare GLB variants for elastic measures other than DTW (including LCSS, ERP, MSM, and TWED) with their existing lower bounds (Section 6.2). Additionally, we showcase the performance of novel lower bounds we propose for EDR and SWALE (Section 6.3). Next, we present an exploratory breakdown analysis of GLB_DTW to understand the contributions of its three components to its performance (Section 6.4). Finally, we demonstrate trade-offs for GLB by varying relevant parameters (Sections 6.5 and 6.6).

6.1 Evaluation of GLB_DTW

To understand if GLB_DTW (Section 3) is an effective LB for DTW, we evaluate it against existing DTW LBs using their pruning power and speedup in 1-NN classification task across 128 datasets. Table 7 reports the performance of GLB_DTW against DTW LBs.

Comparison against LB_Kim, LB_Yi, LB_Keogh: As shown in Table 7, GLB_DTW's pruning power is higher than LB_Keogh by 11.7%, LB_Kim by 58.5%, and LB_Yi by 22.2% on average. Across 128 datasets, the difference in pruning power between GLB_DTW and LB_Keogh goes up to 62.3% and for LB_Kim up to 99.0%. The advantage in pruning power is consistent as out of a total of 128 datasets, GLB_DTW obtains higher pruning power than LB_Kim, LB_Keogh in 127 datasets (Figure 6, Part (a)) and higher than LB_Yi in 110 datasets. GLB's high pruning power shows the effectiveness of query/data dependencies and boundary distances in establishing a tighter lower bound. In terms of speedup, GLB_DTW outperforms LB_Kim in 127 datasets, LB_Keogh in 115 datasets (Figure 6 Part (b)),

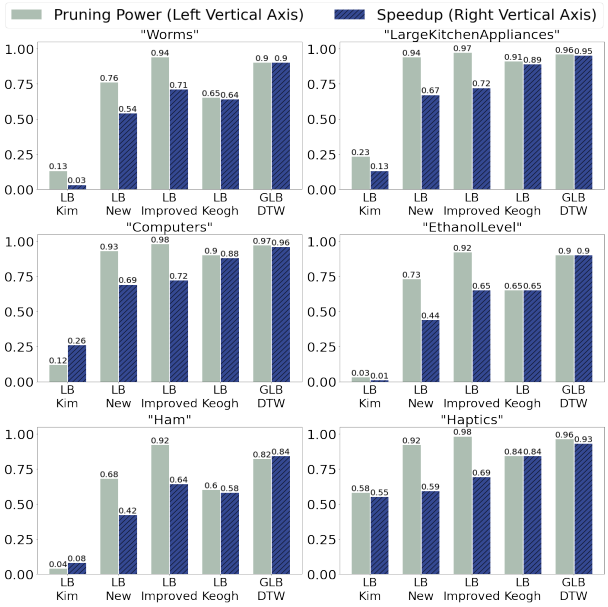


Figure 5: Average pruning power and speedup of DTW LBs.

and LB_Yi in 109 datasets. As summarized in Table 6, GLB_DTW's is faster than LB_Keogh by 1.6x, faster than LB_Kim by 8x on average, and faster than LB_Yi by 4.14x. The advantage of GLB_DTW over LB_Keogh goes up to 3.4x, for LB_Kim up to 36.6x, and for LB_Yi up to 17.44x. GLB_DTW's superior speedup results from its higher tightness over LB_Kim, LB_Keogh, and LB_Yi.

Comparison against LB_New and LB_Improved : We compare the pruning power and speedup of GLB_DTW against LB_New and LB_Improved both when they are used in isolation and running in cascade (with LB_Keogh). Firstly, we compare the pruning power and speedup of GLB_DTW, LB_New, and LB_Improved when used in isolation. Table 7 shows the GLB_DTW's pruning power is slightly better than LB_New by 0.6%, and worse than LB_Improved by 2.8% on average. Interestingly, despite the similarity in pruning power among three lower bounds, LB_Improved and LB_New achieved significantly less speedup than GLB_DTW. Across 128 datasets, GLB_DTW outperforms LB_New in 127 datasets and LB_Improved in all 128 datasets in terms of speedup. It is worth noting that the fact that GLB_DTW has less pruning power than LB_Improved is a result of purposeful design. As we mentioned in Section 3, GLB can integrate LB_Improved into GLB_DTW by replacing current envelopes with the tighter LB_Improved and thus achieve higher pruning power. Nonetheless, the inability to precompute and reuse LB_Improved results in a drastic increase in runtime. In rare situations where even a slight increase in pruning power would compensate for large overheads, GLB_DTW can integrate LB_Improved and meet the needs.

Now we focus on the comparison between GLB_DTW running in a cascade of two envelopes against the cascade of LB_Keogh + LB_New and the cascade of LB_Keogh + LB_Improved. This setting is important for cases where the precomputation of the GLB envelopes is not possible or parameters may need to change dynamically and, therefore, caching of envelopes is not useful. In

Table 7: Summary of pruning power and speedup for existing DTW lower bounds and GLB_DTW.

Evaluation of DTW Lower Bounds						
Metrics	Pruning Power			Speedup		
LBs	Average	Max	Std	Average	Max	Std
LB_Kim	23.41%	87.44%	0.1166%	2.456x	7.591x	1.0752x
LB_Yi	59.72%	83.28%	21.28%	1.321x	4.331x	0.5325x
LB_Keogh	70.19%	98.49%	27.37%	6.314x	25.071x	6.0849x
LB_New	81.27%	99.42%	22.12%	1.110x	2.696x	0.5178x
LB_Improved	84.71%	99.69%	20.02%	1.492x	3.077x	0.6814x
GLB_DTW	**81.93%**	**99.69%**	22.44%	**10.176x**	**41.725x**	10.3284x
LBs when running in a cascade						
LB_New (Cascade)	80.70%	99.29%	21.92%	4.779x	19.838x	4.8138x
LB_Improved (Cascade)	84.0%	99.61%	20.10%	7.393x	32.192x	8.1032x
GLB_DTW (Cascade)	**80.35%**	**99.62%**	23.08%	**10.381x**	**43.491x**	10.1959x

terms of speedup, GLB_DTW enjoys a significant advantage by outperforming LB_Keogh + LB_New in 123 datasets and LB_Keogh + LB_Improved in 124 datasets. On average, GLB_DTW is 1.4x faster than LB_Keogh + LB_Improved and 2.1x faster than LB_Keogh + LB_New. The evaluation of GLB_DTW, LB_New, and LB_Improved highlights the fact that high pruning power doesn't necessarily translate into high speedup. As "HandOutlines", "AllGestureWiimoteX" and "ScreenType" datasets in Figure 5 demonstrate, although LB_Keogh + LB_Improved has higher pruning power than GLB_DTW, their speedups are significantly less GLB_DTW.

6.2 Evaluation of GLB on Elastic Measures other than DTW

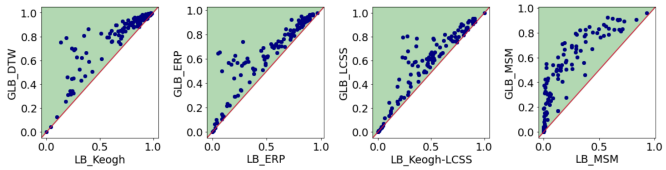
Having demonstrated the advantage of GLB_DTW over existing DTW LBs, we now turn our attention to the comparison between GLB and state-of-the-art LBs of other common elastic measures, including LCSS, ERP, MSM, and TWED, in terms of pruning power and speedup. Next, we report the performance of GLB against three ERP LBs and the LBs of LCSS, MSM, and TWED, respectively.

Comparison against LB_Kim-ERP, LB_ERP, and LB_Keogh-ERP: Out of a total of 128 datasets, GLB_ERP obtains higher pruning power than LB_Kim-ERP, LB_ERP, and LB_Keogh-ERP in 127 datasets, as shown in Figure 6 Part (a). As shown in Table 8, GLB_ERP's pruning power is higher than LB_Keogh-ERP by 66.9%, LB_Kim-ERP by 53.7%, and LB_Keogh-ERP by 11.7% on average. In terms of speedup, GLB_DTW outperforms LB_Kim-ERP in 124 datasets, LB_ERP in 126 datasets, and LB_Keogh-ERP in 118 datasets, as shown in Figure 6 Part (b). As summarized in Table 8, GLB_DTW's is faster by LB_Keogh-ERP by 1.3x, faster than LB_Kim-ERP by 4.3x on average, and faster than LB_ERP by 4.8x. GLB_ERP outperforms all ERP lower bounds in speedup.

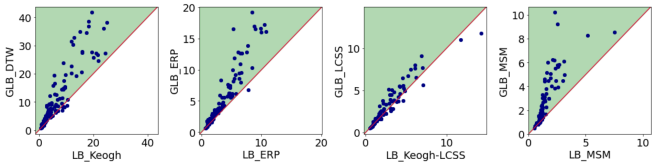
Comparing with LB_Keogh-LCSS, LB_MSM, and LB_TWED: Now we focus on LBs of LCSS, MSM, and TWED, which only have one respective LB reported in the literature. For **LCSS**, Figure 6 part (a) shows that GLB_LCSS significantly outperforms LB_Keogh-LCSS in pruning power: out of a total of 128 datasets, GLB_LCSS achieves higher pruning power than LB_Keogh-LCSS in 119 datasets. In terms of speedup, GLB_LCSS outperforms LB_Keogh-LCSS in 76 datasets, as shown in Figure 6 part (b). For **MSM**, we illustrate that GLB_MSM significantly outperforms LB_MSM. GLB_MSM achieves higher pruning power than LLB_MSM in 120 out of 128 datasets, as shown in Figure 6 Part (a). GLB_MSM pruning power is higher than LB_MSM by 23.13% on average. In terms of speedup, GLB_MSM outperforms LB_MSM in 113 datasets, as shown in Figure 6 Part (b), and GLB_MSM is faster than LB_MSM by 1.57x on average. For **TWED**, we also see that GLB_TWED significantly outperforms LB_TWED. GLB_TWED's achieves higher pruning power than LB_TWED in 125 out of 128 datasets. GLB_TWED pruning power is higher than LB_TWED by 67.32% on average. In terms of speedup, GLB_TWED outperforms LB_TWED in 119 datasets, and GLB_TWED is faster than LB_TWED by 6.56x on average. Based on GLB's consistent advantage in pruning power and speedup over existing baselines, we conclude that GLB variants have established new state-of-the-art performance in popular elastic measures, including LCSS, MSM, and TWED.

Table 8: Summary of average pruning power and speedup for existing LBs and GLB variants. The last column, "win," indicates the number of datasets where the corresponding LB is winning out of a total of 128 datasets in terms of speedup.

Evaluation of Lower Bounds of Other Elastic Measures							
Metrics	Pruning Power			Speedup			Win
LBs	Average	Max	Std	Average	Max	Std	-
Edit Distance with Real Penalty (ERP)							
LB_ERP	0.61%	36.99%	3.66%	0.967x	1.469x	0.0953x	1
LB_Kim	13.87%	76.50%	16.57%	1.134x	3.536x	0.3153x	1
LB_Keogh-ERP	55.84%	96.23%	29.15%	3.222x	10.784x	2.4601x	9
GLB_ERP	**67.55%**	**97.56%**	**26.67%**	**4.780x**	**19.191x**	**4.4416x**	**117**
Move-Split-Merge (MSM)							
LB_MSM	18.97%	92.92%	23.01%	1.358x	7.495x	0.8324x	15
GLB_MSM	**42.09%**	**95.90%**	**30.82%**	**2.302x**	**10.217x**	**1.9056x**	**113**
Time Warp Edit Distance (TWED)							
LB_TWED	2.42%	56.36%	6.8%	0.988x	2.082x	0.1182x	9
GLB_TWED	**69.75%**	**99.78%**	**31.94%**	**6.517x**	**26.285x**	**6.227x**	**119**
Longest Common Subsequence (LCSS)							
LB_Keogh-LCSS	43.72%	100%	29.48%	2.291x	14.257x	2.0396x	52
GLB_LCSS	**53.58%**	**100%**	**31.19%**	**2.541x**	**11.775x**	**2.1548x**	**76**



(a) Pruning Power Comparison



(b) Speedup Comparison

Figure 6: Comparison of GLB variants and state-of-the-art LBs for DTW, ERP, LCSS, and MSM. Part(a) and Part(b) show the pruning power and speedup over 128 datasets, respectively. The blue dots above the diagonal indicate datasets over which GLB outperforms the state of the art.

Table 9: Summary of average pruning power and speedup for GLB_EDR and GLB_SWALE

Evaluation of Novel EDR and SWALE Lower Bounds						
Metrics	Pruning Power			Speedup		
LBs	Average	Max	Std	Average	Max	Std
Edit Distance for Real Sequences (EDR)						
GLB_EDR	**52.55%**	**97.26%**	**31.99%**	**4.096x**	**28.909x**	**4.6569x**
Sequence Weighted Alignment (SWALE)						
GLB_SWALE	**85.07%**	**99.996%**	**24.66%**	**10.684x**	**46.764x**	**9.9162x**

Table 10: Summary of pruning power and speedup for existing DTW lower bounds and GLB_DTW

Lower Bounds	Pruning Power	Speedup
Query Only	69.08%	6.260x
Query + Boundary	74.40%	6.737x
Query + Data	79.89%	9.437x
Query + Data + Boundary (GLB)	**81.93%**	**10.17x**

6.3 Case Studies of EDR and SWALE LBs

Having shown the comparison between GLB and existing lower bounds for various elastic measures, we now demonstrate the performance of the novel lower bounds we propose for EDR and SWALE based on GLB. Table 9 shows the pruning power and speedup for EDR and SWALE. We observe that GLB_EDR and GLB_SWALE achieve high pruning power and speed up comparable to that of state-of-the-art lower bounds for other elastic measures.

6.4 Break Down Analysis

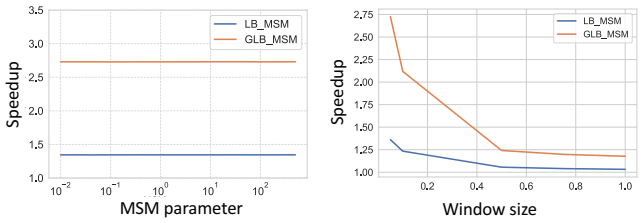
To understand the contribution of various GLB components to its superior performance, we performed a breakdown analysis of GLB_DTW to find the contributions of four different component combinations: query envelopes only, query envelopes and boundary distances, query envelopes and data envelopes, as well as GLB itself.

As shown in Table 10, adopting data envelopes is the driving force behind the improvement in pruning power and speedup. To further understand the contributions of data envelopes, we have kept a record of the percentage of instances when GLB uses data envelopes and query envelopes. We found that GLB chooses the query envelope 50.63% of the time and the data envelope 49.37% of the time. Such results indicate that using only query envelopes, as in the case of LB_Keogh, is suboptimal half of the time, and accounting for both data envelopes and query envelopes is essential.

6.5 Varying Window and other Parameters

In order to assess the effect of varying parameters and changing window sizes on the performance of elastic measure LBs, we conduct a comprehensive evaluation of their impact on the performance of MSM LBs, which produced exceptional results in recent studies [78]. In the first experiment, we held the parameter c constant at 0.5 while systematically increasing the window size from 5% to 100% of the length of the original time series. As the window size increases, the performance of the LBs deteriorates. The results, depicted in Figure 7(b), indicate a declining speedup of LBs as the window size increases. Despite this trend, it is evident that GLB_MSM consistently outperforms LB_MSM across all window sizes.

In the second experiment, we maintained a fixed 5% window size and evaluated ten different parameter values ranging from 0.01 to 500. The results, depicted in Figure 7(a), demonstrate that



(a) Varying elastic measure parameter **(b) Varying warping window size**

Figure 7: Average speedup of LB_MSM and GLB_MSM with varying parameter and warping window size.

variations in this parameter of MSM have limited influence on the performance of MSM LBs with respect to speedup. Furthermore, it is evident that GLB_MSM consistently surpasses LB_MSM.

6.6 Varying Pre-computed Envelopes

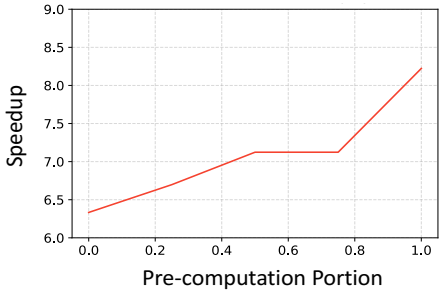


Figure 8: Trade-off of precomputation vs. speedup.

In order to evaluate the trade-off between precomputation of data envelopes and speedup of GLB, we conducted a comprehensive analysis of GLB_DTW in a cascade with varying degrees of precomputation: 0%, 25%, 50%, 75%, and 100%. As precomputation increases, speedup increases at the cost of higher storage requirements. Our results, as illustrated in Figure 8, confirm this hypothesis, showing a positive correlation between increased precomputation from 0% to 100% and increased speedup.

7 CONCLUSION

In this paper, we presented GLB, a generalized framework for deriving lower bounds for elastic measures. Motivated by the disproportionate attention of the research effort to a single elastic measure, we designed an LB to extract cache-friendly summary characteristics, adaptively exploit summaries of both query and target data time series, capture boundary distances in an unsupervised and parameter-free manner, and achieve high LB tightness with low computational cost, resulting in substantial speedup. Based on GLB, we propose new LBs for all elastic measures, including those without existing LBs in the literature. We extensively evaluate GLB against existing LBs, resulting in one of the most comprehensive experimental studies in this area. In particular, we included 11 state-of-the-art lower bounds spanning 5 elastic measures and used 128 datasets in our evaluation. Our findings show that GLB outperforms all existing baseline LBs in terms of speedup. GLB LBs for elastic measures without existing LBs obtain comparable performance to state-of-the-art LBs of other elastic measures. Overall, GLB is a generalizable framework for developing efficient LBs that facilitate accurate and fast time series similarity search.

REFERENCES

[1] Jonathan Alon, Stan Sclaroff, George Kollios, and Vladimir Pavlovic. 2003. Discovering clusters in motion time-series data. In *CVPR*. 375–381.

[2] George Amvrosiadis, Ali R Butt, Vasily Tarasov, Erez Zadok, Ming Zhao, Irfan Ahmad, Remzi H Arpaci-Dusseau, Feng Chen, Yiran Chen, Yong Chen, et al. 2018. Data Storage Research Vision 2025: Report on NSF Visioning Workshop held May 30–June 1, 2018. (2018).

[3] Johannes Aßfalg, Hans-Peter Kriegel, Peer Kröger, Peter Kunath, Alexey Pryakhin, and Matthias Renz. 2006. Similarity search on time series based on threshold queries. In *International Conference on Extending Database Technology*. Springer, 276–294.

[4] 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, 3 (2017), 606–660.

[5] Anthony J Bagnall and Gareth J Janacek. 2004. Clustering time series from ARMA models with clipped data. In *KDD*. 49–58.

[6] Mohini Bariya, Alexandra von Meier, John Paparrizos, and Michael J Franklin. 2021. k-shapestream: Probabilistic streaming clustering for electric grid events. In *2021 IEEE Madrid PowerTech*. IEEE, 1–6.

[7] Nurjahan Begum and Eamonn Keogh. 2014. Rare time series motif discovery from unbounded streams. *Proceedings of the VLDB Endowment* 8, 2 (2014), 149–160.

[8] Donald J Berndt and James Clifford. 1994. Using Dynamic Time Warping to Find Patterns in Time Series.. In *AAAI Workshop on KDD*. 359–370.

[9] Paul Boniol, Michele Linardi, Federico Roncallo, Themis Palpanas, Mohammed Meftah, and Emmanuel Remy. 2021. Unsupervised and scalable subsequence anomaly detection in large data series. *The VLDB Journal* (2021), 1–23.

[10] Paul Boniol and Themis Palpanas. 2020. Series2graph: Graph-based subsequence anomaly detection for time series. *Proceedings of the VLDB Endowment* 13, 12 (2020), 1821–1834.

[11] Paul Boniol, John Paparrizos, Yuhao Kang, Themis Palpanas, Ruey S Tsay, Aaron J Elmore, and Michael J Franklin. 2022. Theseus: navigating the labyrinth of time-series anomaly detection. *Proceedings of the VLDB Endowment* 15, 12 (2022), 3702–3705.

[12] Paul Boniol, John Paparrizos, and Themis Palpanas. 2023. New Trends in Time-Series Anomaly Detection. In *EDBT*.

[13] Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J Franklin. 2021. Sand in action: subsequence anomaly detection for streams. *Proceedings of the VLDB Endowment* 14, 12 (2021), 2867–2870.

[14] Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J Franklin. 2021. SAND: streaming subsequence anomaly detection. *Proceedings of the VLDB Endowment* 14, 10 (2021), 1717–1729.

[15] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *ACM sigmod record*, Vol. 29. ACM, 93–104.

[16] N. Brisaboa, O. Pedreira, D. Seco R. Solar, and R. Uribe. 2008. Clustering-Based Similarity Search in Metric Spaces with Sparse Spatial Centers. *Proc. 34th Int'l Conf. Current Trends in Theory and Practice of Computer Science* 4910 (2008), 186–197.

[17] Yuhan Cai and Raymond Ng. 2004. Indexing spatio-temporal trajectories with Chebyshev polynomials. In *SIGMOD*. 599–610.

[18] Alessandro Camerra, Themis Palpanas, Jin Shieh, and Eamonn Keogh. 2010. iSAX 2.0: Indexing and mining one billion time series. In *2010 IEEE International Conference on Data Mining*. IEEE, 58–67.

[19] Lei Chen and Raymond Ng. 2004. On the marriage of Lp-norms and edit distance. In *VLDB*. 792–803.

[20] Lei Chen, M Tamer Özsu, and Vincent Oria. 2005. Robust and fast similarity search for moving object trajectories. In *SIGMOD*. 491–502.

[21] Qiuxia Chen, Lei Chen, Xiang Lian, Yunhao Liu, and Jeffrey Xu Yu. 2007. Indexable PLA for efficient similarity search. In *VLDB*. 435–446.

[22] Yueguo Chen, Mario A Nascimento, Beng Chin Ooi, and Anthony KH Tung. 2007. Spade: On shape-based pattern detection in streaming time series. In *ICDE*. 786–795.

[23] Bill Chiu, Eamonn Keogh, and Stefano Lonardi. 2003. Probabilistic discovery of time series motifs. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 493–498.

[24] Richard Cole, Dennis Shasha, and Xiaojian Zhao. 2005. Fast window correlations over uncooperative time series. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 743–749.

[25] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. 2008. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)* 40, 2 (2008), 1–60.

[26] H. A. Dau, E. Keogh, K. Kamgar, C.-C., M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. [n.d.]. *The ucr time series classification archive*. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/

[27] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. 2008. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment* 1, 2 (2008), 1542–1552.

[28] Rui Ding, Qiang Wang, Yingnong Dang, Qiang Fu, Haidong Zhang, and Dongmei Zhang. 2015. Yading: Fast clustering of large-scale time series data. *Proceedings of the VLDB Endowment* 8, 5 (2015), 473–484.

[29] Adam Dziedzic, John Paparrizos, Sanjay Krishnan, Aaron Elmore, and Michael Franklin. 2019. Band-limited training and inference for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, 1745–1754.

[30] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2018. The lernaean hydra of data series similarity search: An experimental evaluation of the state of the art. *Proceedings of the VLDB Endowment* 12, 2 (2018), 112–127.

[31] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2019. Return of the Lernaean Hydra: experimental evaluation of data series approximate similarity search. *Proceedings of the VLDB Endowment* 13, 3 (2019), 403–420.

[32] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. 1994. Fast Subsequence Matching in Time-series Databases. In *SIGMOD*. 419–429.

[33] Elias Frentzos, Kostas Gratsias, and Yannis Theodoridis. 2007. Index-based most similar trajectory search. In *ICDE*. 816–825.

[34] V. Ganti, R. Ramakrishnan, J. Gehrke, A.L. Powell, and J.C. French. 1999. Clustering Large Data Sets in Arbitrary Metric Spaces. *Proc. IEEE Int'l Conf. Data Eng. (ICDE)* (1999), 502–511.

[35] Rahul Goel, Sandeep Soni, Naman Goyal, John Paparrizos, Hanna Wallach, Fernando Diaz, and Jacob Eisenstein. 2016. The social dynamics of language change in online networks. In *Social Informatics: 8th International Conference, SocInfo 2016, Bellevue, WA, USA, November 11-14, 2016, Proceedings, Part I 8*. Springer, 41–57.

[36] Sakoe H and Chiba S. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans Acoust, Speech, Signal Process* 26, 1 (1978), 43–49.

[37] Jon Hills, Jason Lines, Edgaras Baranauskas, James Mapp, and Anthony Bagnall. 2014. Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery* 28, 4 (2014), 851–881.

[38] G.R. Hjaltason and H. Samet. 2003. Index-Driven Similarity Search in Metric Spaces. *ACM Trans. Database Systems* 28, 4 (2003), 517 – 580.

[39] G.R. Hjaltason and H. Samet. 2003. Properties of Embedding Methods for Similarity Searching in Metric Spaces. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)* 25, 5 (2003), 530–549.

[40] Bing Hu, Yanping Chen, and Eamonn Keogh. 2013. Time Series Classification under More Realistic Assumptions. In *SDM*. 578–586.

[41] P. Indyk. 1999. A Sublinear Time Approximation Scheme for Clustering in Metric Spaces. *Proc. Ann. Symp. Foundations of Computer Science (FOCS)* (1999), 154–159.

[42] Hao Jiang, Chunwei Liu, Qi Jin, John Paparrizos, and Aaron J Elmore. 2020. Pids: attribute decomposition for improved compression and query performance in columnar storage. *Proceedings of the VLDB Endowment* 13, 6 (2020), 925–938.

[43] Hao Jiang, Chunwei Liu, John Paparrizos, Andrew A Chien, Jihong Ma, and Aaron J Elmore. 2021. Good to the last bit: Data-driven encoding with codecdb. In *Proceedings of the 2021 International Conference on Management of Data*. 843–856.

[44] Konstantinos Kalpakis, Dhiral Gada, and Vasundhara Puttagunta. 2001. Distance measures for effective clustering of ARIMA time-series. In *ICDM*. 273–280.

[45] Shrikant Kashyap and Panagiotis Karras. 2011. Scalable knn search on vertically stored time series. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1334–1342.

[46] Eamonn Keogh. 2002. Exact Indexing of Dynamic Time Warping. In *Proceedings of the 28th International Conference on Very Large Data Bases*. VLDB Endowment.

[47] Eamonn Keogh and Jessica Lin. 2005. Clustering of time-series subsequences is meaningless: Implications for previous and future research. *Knowledge and Information Systems* 8, 2 (2005), 154–177.

[48] Eamonn Keogh and Chotirat Ann Ratanamahatana. 2004. Exact indexing of dynamic time warping. *Knowledge and Information Systems* 7, 3 (2004), 358–386.

[49] Sang-Wook Kim, Sanghyun Park, and Wesley W Chu. 2001. An index-based approach for similarity search supporting time warping in large sequence databases. In *Data Engineering, 2001. Proceedings. 17th International Conference on*. IEEE, 607–614.

[50] Chan Kin-pong and Fu Ada. 1999. Efficient Time Series Matching by Wavelets. In *ICDE*. 126–133.

[51] Flip Korn, H. V. Jagadish, and Christos Faloutsos. 1997. Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences. In *SIGMOD*. 289–300.

[52] Sanjay Krishnan, Aaron J Elmore, Michael Franklin, John Paparrizos, Zechao Shang, Adam Dziedzic, and Rui Liu. 2019. Artificial intelligence in resource-constrained and shared environments. *ACM SIGOPS Operating Systems Review* 53, 1 (2019), 1–6.

[53] S. K. Lam, Pitrou A., and S Seibert. 2015. Numba: A llvm-based python jit compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure*

in HPC (2015), 1–6.
[54] Daniel Lemire. 2009. Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern recognition* 42, 9 (2009), 2169–2180.
[55] Xiang Lian, Lei Chen, Jeffrey Xu Yu, Guoren Wang, and Ge Yu. 2007. Similarity match over high speed time-series streams. In *ICDE*. 1086–1095.
[56] Jessica Lin, Michail Vlachos, Eamonn Keogh, and Dimitrios Gunopulos. 2004. Iterative incremental clustering of time series. In *EDBT*. 106–122.
[57] Michele Linardi and Themis Palpanas. 2018. Scalable, variable-length similarity search in data series: The ULISSE approach. *Proceedings of the VLDB Endowment* 11, 13 (2018), 2236–2248.
[58] Jason Lines and Anthony Bagnall. 2015. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery* 29, 3 (2015), 565–592.
[59] Chunwei Liu, Hao Jiang, John Paparrizos, and Aaron J Elmore. 2021. Decomposed bounded floats for fast compression and queries. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2586–2598.
[60] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 413–422.
[61] Shinan Liu, Tarun Mangla, Ted Shaowang, Jinjin Zhao, John Paparrizos, Sanjay Krishnan, and Nick Feamster. 2023. AMIR: Active Multimodal Interaction Recognition from Video and Network Traffic in Connected Environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 1 (2023), 1–26.
[62] Vlachos M, Hadjieleftheriou M, Gunopulos D, and Keogh E. 2003. Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the 9th ACM SIGKDD international conference on knowledge discovery and data mining (SIGKDD)*. ACM, 216–225.
[63] Pierre-François Marteau. 2009. Time Warp Edit Distance with Stiffness Adjustment for Time Series Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (2009), 306 – 318.
[64] Kathy McKeown, Hal Daume III, Snigdha Chaturvedi, John Paparrizos, Kapil Thadani, Pablo Barrio, Or Biran, Suvarna Bothe, Michael Collins, Kenneth R Fleischmann, et al. 2016. Predicting the impact of scientific concepts using full-text features. *Journal of the Association for Information Science and Technology* 67, 11 (2016), 2684–2696.
[65] Vasileios Megalooikonomou, Qiang Wang, Guo Li, and Christos Faloutsos. 2005. A multiresolution symbolic representation of time series. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*. IEEE, 668–679.
[66] Michael D Morse and Jignesh M Patel. 2007. An efficient and accurate method for evaluating time series similarity. In *SIGMOD*. 569–580.
[67] Abdullah Mueen, Eamonn Keogh, Qiang Zhu, Sydney Cash, and Brandon Westover. 2009. Exact discovery of time series motifs. In *Proceedings of the 2009 SIAM international conference on data mining*. SIAM, 473–484.
[68] Panagiotis Papapetrou, Vassilis Athitsos, Michalis Potamias, George Kollios, and Dimitrios Gunopulos. 2011. Embedding-based subsequence matching in time-series databases. *TODS* 36, 3 (2011), 17.
[69] Ioannis Paparrizos. 2018. *Fast, Scalable, and Accurate Algorithms for Time-Series Analysis*. Ph.D. Dissertation. Columbia University.
[70] J. Paparrizos. 2019. *2018 ucr time-series archive: Backward compatibility, missing values, and varying lengths*. https://github.com/johnpaparrizos/UCRArchiveFixes.
[71] 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.
[72] 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.
[73] John Paparrizos and Michael J Franklin. 2019. GRAIL: efficient time-series representation learning. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1762–1777.
[74] 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*. ACM, 1855–1870.
[75] John Paparrizos and Luis Gravano. 2017. Fast and Accurate Time-Series Clustering. *ACM Transactions on Database Systems (TODS)* 42, 2 (2017), 8.
[76] 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.
[77] John Paparrizos, Chunwei Liu, Bruno Barbarioli, Johnny Hwang, Ikraduya Edian, Aaron J Elmore, Michael J Franklin, and Sanjay Krishnan. 2021. VergeDB: A Database for IoT Analytics on Edge Devices.. In *CIDR*.
[78] John Paparrizos, Chunwei Liu, Aaron J Elmore, and Michael J Franklin. 2020. Debunking four long-standing misconceptions of time-series distance measures. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1887–1905.
[79] John Paparrizos, Ryen W White, and Eric Horvitz. 2016. Detecting devastating diseases in search logs. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 559–568.
[80] John Paparrizos, Ryen W White, and Eric Horvitz. 2016. Screening for pancreatic adenocarcinoma using signals from web search logs: Feasibility study and results. *Journal of oncology practice* 12, 8 (2016), 737–744.
[81] Chotirat Ann Ratanamahatana and Eamonn Keogh. 2004. Making time-series classification more accurate using learned constraints. In *SDM*. 11–22.
[82] Hiroaki Sakoe and Seibi Chiba. 1971. A dynamic programming approach to continuous speech recognition. In *ICA*. 65–69.
[83] Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing* 26, 1 (1978), 43–49.
[84] 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.
[85] Yilin Shen, Yanping Chen, Eamonn Keogh, and Hongxia Jin. 2018. Accelerating time series searching with large uniform scaling. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 234–242.
[86] Jin Shieh and Eamonn Keogh. 2008. i SAX: indexing and mining terabyte sized time series. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 623–631.
[87] Yutao Shou, Nikos Mamoulis, and David Cheung. 2005. Fast and exact warping of time series using adaptive segmental approximations. *Machine Learning* 58, 2-3 (2005), 231–267.
[88] Alexandra Stefan, Vassilis Athitsos, and Gautam Das. 2013. The move-split-merge metric for time series. *TKDE* 25, 6 (2013), 1425–1438.
[89] Chang Wei Tan, François Petitjean, and Geoffrey I Webb. 2019. Elastic bands across the path: A new framework and method to lower bound DTW. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 522–530.
[90] Chang Wei Tan, François Petitjean, and Geoffrey I Webb. 2020. FastEE: Fast Ensembles of Elastic Distances for time series classification. *Data Mining and Knowledge Discovery* 34, 1 (2020), 231–272.
[91] Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. 2006. Indexing multidimensional time-series. *The VLDB Journal* 15, 1 (2006), 1–20.
[92] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. 2002. Discovering similar multidimensional trajectories. In *Proceedings 18th international conference on data engineering*. IEEE, 673–684.
[93] Jun Wang, Wei Liu, Sanjiv Kumar, and Shih-Fu Chang. 2015. Learning to hash for indexing big data—A survey. *Proc. IEEE* 104, 1 (2015), 34–57.
[94] Jingdong Wang, Ting Zhang, Nicu Sebe, Heng Tao Shen, et al. 2017. A survey on learning to hash. *IEEE transactions on pattern analysis and machine intelligence* 40, 4 (2017), 769–790.
[95] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh. 2013. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery* (2013), 1–35.
[96] Yang Wang, Peng Wang, Jian Pei, Wei Wang, and Sheng Huang. 2013. A data-adaptive and dynamic segmentation index for whole matching on time series. *Proceedings of the VLDB Endowment* 6, 10 (2013), 793–804.
[97] Geoffrey I Webb and François Petitjean. 2021. Tight lower bounds for dynamic time warping. *Pattern Recognition* 115 (2021), 107895.
[98] Dragomir Yankov, Eamonn Keogh, and Umaa Rebbapragada. 2008. Disk aware discord discovery: Finding unusual time series in terabyte sized datasets. *Knowledge and Information Systems* 17, 2 (2008), 241–262.
[99] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2016. Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 1317–1322.
[100] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Zachary Zimmerman, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2018. Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile. *Data Mining and Knowledge Discovery* 32, 1 (2018), 83–123.
[101] Byoung-Kee Yi, HV Jagadish, and Christos Faloutsos. 1998. Efficient retrieval of similar time sequences under time warping. In *Data Engineering, 1998. Proceedings., 14th International Conference on*. IEEE, 201–208.
[102] P. Yianilos. 1993. Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces. *Proc. ACM-SIAM Symp. Discrete Algorithms* (1993), 311–321.
[103] Kostas Zoumpatianos, Stratos Idreos, and Themis Palpanas. 2016. ADS: the adaptive data series index. *The VLDB Journal—The International Journal on Very Large Data Bases* 25, 6 (2016), 843–866.