STEB: In Search of the Best Evaluation Approach for Synthetic Time Series

Michael Stenger
University of Wuerzburg
97074 Wuerzburg, Germany
michael.stenger@uni-wuerzburg.de

Robert Leppich
University of Wuerzburg
97074 Wuerzburg, Germany
robert.leppich@uni-wuerzburg.de

André Bauer
Illinois Institute of Technology
Chicago, US
abauer7@iit.edu

Samuel Kounev
University of Wuerzburg
97074 Wuerzburg, Germany
samuel.kounev@uni-wuerzburg.de

Abstract

The growing need for synthetic time series, due to data augmentation or privacy regulations, has led to numerous generative models, frameworks, and evaluation measures alike. Objectively comparing these measures on a large scale remains an open challenge. We propose the **S**ynthetic **T**ime series **E**valuation **B**enchmark (STEB)—the first benchmark framework that enables comprehensive and interpretable automated comparisons of synthetic time series evaluation measures. Using 10 diverse datasets, randomness injection, and 13 configurable data transformations, STEB computes indicators for measure reliability and score consistency. It tracks running time, test errors, and features sequential and parallel modes of operation. In our experiments, we determine a ranking of 41 measures from literature and confirm that the choice of upstream time series embedding heavily impacts the final score.

1 Introduction

Time series (TS) data is central to many domains and applications such as forecasting medical data [22] or human activity recognition [17]. Yet too often there is a lack of availability, quantity, and quality of data, for instance, due to privacy concerns. Data synthesis can be a solution, but it remains challenging in practice [14, 8]. A key step towards high-quality synthetic data is a comprehensive, reliable evaluation strategy. The fundamental problem is the lack of ground truth data similar to unsupervised clustering. To handle this complex, indirect assessment, the common approach is to use different measures to quantify various quality aspects [60]. However, with dozens of measures having been proposed in recent years, the selection of the "ideal" set of measures is particularly challenging [52]. For instance, seminal works [60, 59, 49] and recently proposed frameworks [45, 41] use different combinations with little overlap. Furthermore, the aspects of synthetic data quality considered typically differ [51]. As a consequence, comparability of generative performance is hindered, the state-of-the-art unknown. Previous works studied small groups of measures in a tailored analysis or developed frameworks to generate and evaluate synthetic data. However, to the best of our knowledge, there is no detailed, objective, and broad study on the effectiveness of these measures applied to TS. Critically, this raises serious concerns regarding the reliability of past evaluation results for TS generative models. Hence, we propose **S**ynthetic **T**ime series **E**valuation **B**enchmark (STEB), the first benchmark framework to conduct large-scale and multi-faceted analysis of synthetic TS

Preprint. Under review.

evaluation measures. With STEB, we aim to narrow down the clutter of measures to a standardized set, drastically increasing the comparability of generative performance.

Scope. We focus on uni-/multivariate, real-valued TS. An evaluation measure is a function $m : \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \rightarrow \mathbb{R}$ where $\mathcal{P}$ is the super-set operator and $\mathcal{X}$ the data space, e.g., time series. For a real dataset $D_r \subset \mathcal{X}$ and synthetic dataset $D_s \subset \mathcal{X}$, we call $s = m(D_r, D_s)$ its score. In this context, "real" data is the initial data to learn. Note that m assesses the generated data, not generators. Hence, we do not consider generator-dependent measures such as duality gap [50]. Similarly, we limit this study to quantitative measures with clear s and exclude, for instance, visualizations such as the popular t-SNE plot [60].

Contributions. With this work, we contribute to synthetic data research in three ways:

1. We design and implement STEB, a novel benchmark framework for comprehensive, interpretable, and automated analysis of quantitative synthetic TS measures.
2. We analyze 41 measures with respect to reliability, consistency, and running time. More specifically, we rank the measures in four aspects of quality and in running time.
3. We investigate the impact of the upstream TS embedding on the final score.

This paper is structured as follows. Section 2 puts our work in the context of related work. Section 3 introduces STEB and Section 4 the experiments on TS synthesis measures. Section 5 presents and discusses the results. Section 6 concludes the paper.

2 Related Work

With the increasing interest in synthetic data generation, different studies on evaluation measures were conducted. One noteworthy work examines three of the then most commonly used evaluation measures for image generation [53]. Their combined theoretical and empirical approach is tailored towards each measure. Key findings are that the behavior of different measures is often largely independent of each other and synthetic data utility is application specific. Lucic et al. [34] presented an empirical study on GAN models and evaluation measures for image synthesis. Their focus is on the Fréchet inception distance (FID), analyzing bias, variance, robustness to mode dropping, different image embeddings, and FID value range. In terms of experimental design, the closest work we know of is by Huang et al. [19]. They manipulate image data in different ways to study the behavior and efficiency of five evaluation measures w.r.t. overfitting, mode collapse/dropping, discriminability, and robustness. Recently, Ismail-Fawaz et al. [20] conducted a comparison of eight measures for the evaluation of human motion generation and proposed a ninth. They analyzed each measure qualitatively, followed by a quantitative comparison using a conditional generator on one real dataset to controllably create human motions.

STEB differs from the above works in many ways: (i) While these work are all (very) focused in their analysis, we compare a wide range of measures of diverse designs and purposes; (ii) STEB is designed for TS, whereas previous works target images and human motion; (iii) STEB is model agnostic with regard to the data generator and evaluation measure, unlike Lucic et al. [34] and Huang et al. [19], who focus on non-conditional GANs, or Theis et al. [53], who employ analysis specific to the measures; (iv) We utilize different hand-crafted data manipulations, while Ismail-Fawaz et al. [20] use a neural model to create the test synthetic data; (v) Most of the related work does not capture recent developments. STEB incorporates established and recent measures and can be extended to support further analysis techniques and future measures, continuously tracking the state-of-the-art; (vi) STEB is more fine-grained in its analysis, as it differentiates four aspects of synthesis quality, and it is more comprehensive in terms of experimental parameters.

There are three related synthetic data benchmarks. Synthcity is a framework for benchmarking tabular, image, and TS data generators [45]. It incorporates multiple generators, evaluation measures, and datasets in an automated test pipeline. Similarly, TSGBench [2] and Time Series Generative Modeling (TSGM) [41] are frameworks for time series synthesis. While TSGBench specializes on benchmarking generators, TSGM is presented as a more general solution including application cases. These benchmarks mainly differ in the choice of integrated measures, generators, and datasets. They evaluate the respective generative models and accept the implemented measures as given. STEB, however, assesses the measures themselves in order to determine which to best include in tools such as Synthcity, TSGBench, or TSGM. The diversity of evaluation suites highlights the need for a systematic analysis tool for measures.

3 STEB: Synthetic Time Series Evaluation Benchmark

In this section, we introduce the approach and present the design of STEB, the first benchmark framework for analyzing evaluation measures for synthetic TS.

3.1 Controlled Distribution Modulation

The evaluation of synthetic data D_s is commonly centered around given data generators with unknown performance on a set of real data D_r. The absence of ground truth complicates analysis and comparison, leading to the development of complex evaluation measures. To this end, we change the perspective and select benchmark scenarios with an expected outcome to evaluate the performance of the measures themselves. Inspired by Lucic et al. [34], we construct such scenarios by replacing the "regular" TS generator G with a pseudo-generation method: *Transformation* T. Formally, $T : \mathbb{R}^{n \times l \times d} \times [0, 1] \to \mathbb{R}^{n \times l \times d}$ is a function taking a dataset of n real-valued TS of length l and dimension d, along with a scale factor κ specifying the transformation intensity. Its output is a "transformed" dataset D_T^κ. However, finding T such that the score $s = m(D_r, D_T^\kappa)$ can be assessed directly and absolutely is challenging for the non-trivial case $\kappa > 0$. D_T^κ must be complex enough to be a realistic test case, but determining the expected s must be known for non-trivial input data. Additionally, different measures produce scores in varying ranges and optimization directions.

To address this issue, we combine transformations with a second concept, the *modulation* of T via intensity κ. Intuitively, we modify D_r more and more, creating a series of ever more different synthetic data, and assess the score of each step relative to the others. More formally: D_r is a sample drawn from an underlying distribution P in $\mathbb{R}^{l \times d}$, and synthesizing D_s equates to generating new samples from P. A 1D simplification of P and the modulation process is depicted in Figure 1. κ allows us to create a sample D_T^κ of a shifted and distorted distribution P_T in the data space, ranging from P itself to a completely different distribution. Assuming m measures some aspect of similarity of the underlying distributions of two datasets, we expect s to get worse with increasing κ. This is the expected outcome we can test empirically. More specifically, we test if $s_0 > s_1 > s_2 > \dots$ (assumption: higher is better) for $s_i = m(D_r, D_T^{\kappa_i})$ along the "modulation path" $\kappa_0 < \kappa_1 < \kappa_2 < \dots$. Using this condition, we compute a reliability indicator for m under different test cases, varying T, D_r, and the random seed. The average value across all tests serves as approximation for the measure's reliability.

Example. Let

$$m_{\text{iMAE}} : D \times D' \mapsto \left(10^{-3} + \frac{1}{nld} \sum_{i,j,k} |D_{i,j,k} - D'_{i,j,k}| \right)^{-1} \tag{1}$$

be the inverse mean absolute error on ordered D, D'. Further, let $T : D, \kappa \mapsto \{x + \kappa \mid x \in D\}$ be a transformation increasing every scalar in every TS in D by the scale factor. We compute $D_T^0 = T(D, 0)$, $D_T^{0.5} = T(D, 0.5)$, $D_T^1 = T(D, 1)$ and $s_0 = m_{\text{iMAE}}(D, D_T^0)$, $s_1 = m_{\text{iMAE}}(D, D_T^{0.5})$, $s_2 = m_{\text{iMAE}}(D, D_T^1)$. As m_{iMAE} measures the inverse average distance between the scalars of two datasets and T increases these values with increasing κ, we find that $s_0 > s_1 > s_2$. Hence, m_{iMAE} behaves as expected and we would assign a high reliability indicator. Note that the regular MAE would satisfy none of these inequalities, resulting in bad performance.

3.2 Transformations

In the following, we list all 13 transformations implemented in STEB and used in our experiments, chosen based on the diversity of data changes they force the measures to detect, their ability to allow gradual transformation with κ, a sensible running time, and the interpretability of induced changes.

Gaussian noise adds a matrix of random values sampled from a Gaussian distribution with 0 mean and $\frac{\kappa}{2}$ variance to the TS in D_r. To standardize the amount of noise for each dataset, we scale D_r to $[0, 1]$ before applying the noise and rescale it afterwards.

Label corruption corrupts the labeling of classified datasets by randomly swapping the labels of $\frac{\kappa}{10}$ of instances. Only applicable to measures sensitive to labels.

Misalignment rotates the different channels of each TS in D_r randomly by p positions with probability κ, $1 \leq p \leq \kappa(l - 1)$, and narrows the gap between the formerly first and last values. Only applicable to multivariate TS.

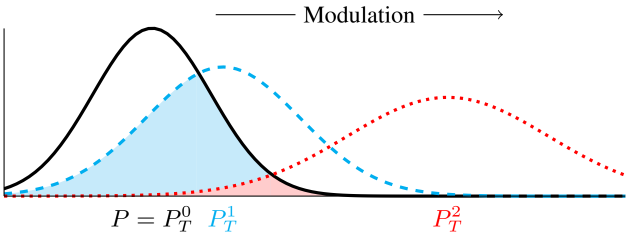

Figure 1: Depiction of the modulation concept. By modulating parameter κ, we can influence the degree to which a transformation T impacts dataset D_r (resp. its underlying distribution P) to create the pseudo-synthetic dataset D_T with distribution P_T. For $\kappa = 0$, we get $P_T^0 = P$ (black), for $\kappa = 0.3$, it might be P_T^1 (blue, dashed), and for $\kappa = 0.9$, it is P_T^2 (red, dotted).

Mode collapse simulates a mode collapse by sampling each class of D_r down by κ, replacing the dropped instances in each mode with noisy duplicates of the remaining time series. Only applicable to labeled data.

Mode dropping simulates mode dropping by replacing all TS of κ classes in D_r with TS from the remaining classes proportional to their size. Only applicable to labeled data.

Moving average transforms D_r by applying a moving average to each channel of each TS. The filter used for averaging is $a \cdot l \cdot \kappa + 1$ wide where $a = \frac{1}{3}$ if $l \geq 30$ else $a = 1$ and centered on the modified value.

Rare event drop probes the sensitivity to rare events, in this case, the smallest class in D_r. D_T is created by swapping out κ of its instances with ones of other classes taken from another substitute dataset D_{rs}. This set provides additional real TS from the same distribution exclusively accessed by transformations. Only applicable to labeled data.

Reverse substitution probes the sensitivity to leaking real TS into the synthetic set. It starts with D_{rs} and gradually adds up to ten D_r instances to D_T with increasing κ.

Salt & pepper adds noise to the data by replacing random values in D_r by 0 and 1 with probability $\frac{\kappa}{2}$.

Segment leaking builds the output D_T by using D_{rs} as a basis and replacing 30κ random segments from TS in D_{rs} with segments from D_r. A segment is one channel of a TS subsequence and between $\frac{l}{4}$ and $\frac{l}{2}$ long.

STL decomposition transforms D_r by decomposing every channel c of every TS into season s, trend t, and residual r using LOESS [10] followed by its reconstruction via linear combination $c_{\text{new}} = (\kappa u_1 + 1)s + (\kappa u_2 + 1)t + (\kappa u_3 + 1)r)$ with $u_1, u_2, u_3 \sim \mathcal{U}_{[-1,1]}$.

Substitution replaces a fraction κ of the TS in D_r with instances from D_{rs} at random.

Wavelet transform decomposes each TS in D_r with discrete wavelet transform [27], rescales its scale component by κ, and inverses the transformation again. This gradually increases/removes the temporal structure of each channel while leaving the residuals intact.

Note that none of the presented transformations is expected to mimic synthetic time series produced by typical generative models, which are practically intractable. Instead, they are intended to check individual, tractable aspects of relevant measure behavior. Collectively, the transformations cover a wide and diverse range of measure behavior across hundreds of tests. This is comparable to classification or forecasting, where the evaluation with fixed settings and dataset selection serves as approximation of general model performance.

3.3 Benchmark Design

The purpose of our benchmark is to enable the comprehensive, interpretable, and automated comparison of TS evaluation measures. Its main design goals are extensible components, fast and resource-efficient execution, and flexibility in operation. To achieve this, STEB is based on the distribution modulation and transformation concepts, while being centered around the measures under test and being supported by several auxiliary components. The high-level architecture and data-/information-flow is depicted in Figure 2. Each component is described in the next section.

STEB works with two organizational units, *experiments* and *tests*. An experiment represents one run of the benchmark; it is initiated by the user and specified through a configuration (e.g., see Listing 1). A test is one pass through STEB from the *Transformation* component to *Storage*; it is characterized by a set of parameters including the input dataset, the measure to test, and the transformation to apply. An experiment includes multiple, often thousands of tests. It starts with an initialization via the *Configuration & Management* component, followed by the *Preprocessing* of all required real data. Afterwards, the included tests are gradually processed, starting with the transformation to create D_T. If required by the measure, the test datasets are scaled to $[0, 1]$ or embedded. Finally, the data flow arrives at the *Measure* component, where a score for D_{train}, D_{T}, and $D_{\text{held-out}}$ is computed. The transformation, scaling/embedding, and measure steps are repeated for each κ. Upon completion, the different scores are collected and stored, before a new test is selected by the *Management* component. If a test incurs an exception, it will be recorded as failure, its reason logged, and the experiment resumed with the option to repeat the test at a later time. The *Evaluation* component is called once all tests are processed. Measures, embedders, transformations, and datasets are integrated via common interfaces, making these components extensible.

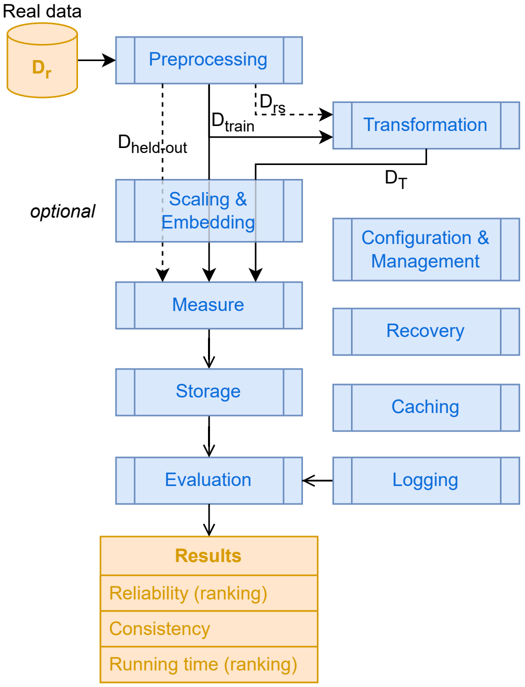

Figure 2: Architectural design of STEB. Input (top left) and output (bottom) are highlighted in orange, STEB components in blue. Datasets are referenced as D and the flow of data is indicated by arrows with filled head. The open-headed arrows mark information flow such as scores, rankings, measurements, and error messages. Dashed arrows denote conditional data flow, where $D_{\text{held\_out}}$ depends on the measure and D_{rs} on the transformation.

3.4 Components

Core STEB components are described below, supporting components for storage, caching, recovery, and logging are detailed in Appendix C.

Data Preprocessing. STEB implements a diverse group of ten datasets from different domains. A detailed list with description, source, and characteristics can be found in Appendix B.1. We implemented an automated preprocessing pipeline to prepare each dataset in the specific format required by each measure and transformation. It retrieves the specified data from its online source, removes outliers, interpolates missing values, equalizes TS lengths, extracts class labels, and calculates dataset statistics. Depending on the input requirements of transformation and measure, the real data is split into up to three, usually two, equally sized subsets D_{train}, D_{rs}, and $D_{\text{held-out}}$. D_{train} represents the data available to a potential generator, D_{rs} is the substitute data, and $D_{\text{held-out}}$ simulates generator test data. The latter two are optional.

Scaling & Embedding. Many measures require further preprocessing of the input data. One group works well only on scaled values, another (overlapping) group operates on real-valued vectors and is not directly applicable to TS data (especially if they are multivariate). The scaling to range $[0, 1]$ is straightforward. To facilitate vector representations, this component has three embedders implemented. By default, the *TS2Vec* [61] representation model is employed. Alternatively, it features a non-DL-based embedder called *Catch22* [33] and the trivial concatenation of feature channels referred to as *Concat* $e_{\text{concat}} : \mathbb{R}^{l \times d} \to \mathbb{R}^{l \cdot d}$ as baseline model.

Measure. STEB includes 44 measures collected over the past years from different backgrounds with diverse evaluation goals and varying levels of complexity. Two are qualitative, visualization-based measures, while the others are quantitative measures. They usually produce a score $s \in \mathbb{R}_0^+$, but there are outliers such as C2ST [31] with $s \in \{\texttt{true}, \texttt{false}\}$. Moreover, the measures differ in terms of input data. Three measures only consider synthetic data, here provided as D_T, while most (33) also consider the real data provided to the generator, D_{train}. The remaining eight require a third, real dataset unknown to the generator, $D_{\text{held-out}}$. Due to the space limitations, we prepared an alphabetical list of the 44 measures with descriptions and sources in Appendix A. For implementation details, we refer to the STEB code and documentation.

Evaluation. Once all tests in an experiment are concluded, whether successful or not, the evaluation can start. To this end, the scores produced for the tested measures and the recorded running times (if available) are analyzed by this component based on three criteria: (i) the *reliability* of a measure to truthfully and accurately reflect the quality of a given synthetic dataset in its score; (ii) the *consistency* of the measure's scores with respect to changing parameters such as random seed or dataset; (iii) the measure's speed in computing a score for a given dataset. The evaluation results are aggregated in different tabular formats and diagrams, formatted, and statistically analyzed (see Section 5 and Appendix F). While recording running time is straightforward, the quantitative evaluation of reliability and consistency requires an understanding of quality. Following previous works [55, 1, 57], we break down quality into different aspects. Specifically, we consider four key aspects, which hereafter will be referred to as *categories* in the evaluation:

Fidelity refers to the similarity of individual synthetic data instances to real ones, ensuring they exhibit realistic properties such as patterns, trends, and volatility in time series.
Generalization is a generator's ability to create data beyond the training data itself or noisy versions thereof.
Privacy aims to reduce or even eliminate the risk of disclosure of sensitive information in the data.
Representativeness is the plausibility of D_s to be a sample of P, which can be thought of as the closeness of the synthetic dataset to the real one in the feature space. This implies an amount of diversity proportional to the real data and also extends to the utility of D_s for downstream tasks.

3.5 Determining Reliability and Consistency

To quantify reliability, we compute an indicator $r_{\text{rel}} \in [0, 1]$ for each category and measure m. We define the expected category-dependent behavior of m on transformation T with four options: *Improve* means we expect the score to get better with increasing κ; *Worsen* represents the expectation of a worsening score; *Constant* expects the score to remain largely unaffected by a changing κ; and *N/A* means that T is not applicable in this category. These expected behaviors can be confidently assigned a priori. The result is listed in Table 3. If the behavior is defined, we can compute a task-specific indicator $r_{\text{rel}}(t)$ using m's scores $s_0, \ldots, s_{k-1}$, where k is the number of modulation steps.
Improve. Assuming $s_i \in \mathbb{R}$ and improvement means to increase s_i, we define r_{rel} as the fraction of score pairs (s_i, s_j), $i < j$ with $s_i < s_j$. If instead s_i is a boolean, the performance is determined based on a point system, where points are assigned based on the positions of `true` and `false` along the modulation path and then normalized to $[0, 1]$.
Worsen. If $s_i \in \mathbb{R}$, we symmetrically use the fraction of pairs with $s_i > s_j$, whereas if s_i is a boolean, we swap the points assigned for `true` and `false`.
Constant. For real-valued s_i, we expect all scores to be close to their median. We use the median instead of the mean for robustness. For boolean scores, $r_{\text{rel}}(t)$ is the normalized number of unequal consecutive values. m's reliability r_{rel} is the average value across all t. More details and an example are provided in Appendix D.1.

We define consistency r_{con} of m via the pairwise statistical difference between groups of r_{rel} indicators, where each group has either the same random seed or underlying dataset. The idea is that the behavior of m should depend on the relationship of synthetic and real data and not be impacted significantly by randomness or the real dataset alone. We use r_{rel} as a proxy. Please find additional information in Appendix D.2.

The running times of measures and embedders are recorded separately. For each test, STEB tracks the time required for the measure to compute the score on a fully prepared dataset, that is, after transformation and embedding. If an embedding is required, the amounts of time needed for training (if applicable) and inference are tracked.

4 Experimental Design and Execution

In this section, we outline two different experiments conducted with STEB. For the modulation, we use eleven equally spaced steps, that is, $\kappa = 0.0, 0.1, \ldots, 1.0$, in both experiments. Similarly, we use all 10 datasets in each experiment. In term of hardware, we used a server with an AMD EPYC 7763, 256GB RAM, and an NVIDIA A100 running CUDA 12.4 capped at 40GB per worker. All running times were recorded on this system. To speed up the computation, we utilize up to five workers running in parallel. Still, we impose a strict time limit of 120 minutes per test to avoid excessive running times.

4.1 Main Experiment: Ranking Measures

We analyze 41 of STEB's 44 implemented measures. Of the remaining three measures, two are qualitative and thus difficult to assess objectively, while one computes instance-level scores unsuited to be condensed to a dataset-level score. Each measure is evaluated in all four categories. The goal is a comparison regarding reliability, consistency, and running time to test which measures are best suited to evaluate a specific category and within what time frame. Importantly, the goal is to provide running time estimates rather than precise measurements. We use TS2Vec as embedding and ten random seeds.

4.2 Side Experiment: TS Embedding Models

Embedder-dependent measures operate on embedded vector data instead of the original time series, adding an extra source of variation to the final score. This issue was previously raised in the review of [19]. To this end, we examine the role of the embedding model in the synthesis evaluation process using STEB, providing empirical results for the 24 implemented embedder-dependent measures. We pairwise compare TS2Vec, Catch22, and Concat on a test-by-test basis. More specifically, we pair the tests of different embedders on the otherwise matching parameters dataset, transformation, and random seed, to compute two metrics between the two matched score sets: the mean absolute percentage error (MAPE) and the Pearson correlation coefficient (PCC). Tests for which no matching is possible due to a failed test on one side are ignored.

5 Results

Given our experiment setup, the performed experiments resulted in an extensive output. Each of the 63,601 successful tests produces at least eleven scores along with multiple running time measurements. We evaluate and present them in different ways, some of which are placed in Appendix F due to space constraints. Not all of the 68,666 tests are successful and count when evaluating the performance of the implemented measures. The success rate heavily depends on the resource and time demands of the measure, the size of the dataset, and the transformation applied. Most measures have a success rate of around 98%. The details are listed in Table 4. Reasons for failure include system or graphics memory overflow, measure-specific exceptions, and exceeding running time. A detailed list can be found in Table 5. General guidelines on how to use the results for measure selection are given in Appendix G, which can still be used when new measures change current rankings.

5.1 Main Experiment

The main experiment included 41,432 tests, of which 39,063 were successful. Unfortunately, for the measure *DOMIAS*, all but one test failed, mostly due to excessive GPU memory demands. Still, this observed high resource demand is already a valuable insight for potential users. For the remaining 40 measures, we calculated r_{rel}, r_{con}, and the average running time. We list the reliability indicators in Table 1 including the standard deviation observed between tests in alphabetical order. A ranking by category is shown in Table 6.

The rankings for fidelity and representativeness are overall similar with α-*Precision* as the top position in both categories. *ACS* and *autocorrelation* perform surprisingly well in the generalization and privacy categories and not as well regarding fidelity and representativeness, which they are actually intended for. Generalization measures such as C_T are further down in the ranking. More intuitive is the third position of the "novelty" measure *authenticity* in the privacy category. Across categories,

Table 1: Reliability indicator overview for experiment *Main* (left) and embedding comparison for the *Embedders* experiment. LEFT: For each measure, we list r_{rel} as the mean across different tests and the standard deviation in all four categories (Mean ± StD). In the upper block (all but DOMIAS), we have highlighted the best (second best) indicator in bold (by underlining). DOMIAS is listed separately as its results are based on one test only and are thus not statistically significant (see Table 4). RIGHT: For each of the three embedder pairs, we report the mean absolute percentage error (MAPE) and the Pearson correlation coefficient (PCC). MAPE is capped at 10 (1000 %). For DOMAIS, no tests could be matched. All values are real numbers.

Fidelity	Generalization	Privacy	Representativeness	Measure	TS2Vec-Concat		Catch22-Concat		Catch22-TS2Vec	
					MAPE	PCC	MAPE	PCC	MAPE	PCC
.416 ± .413	**.726** ± **.320**	.676 ± .313	.347 ± .369	ACS						
.783 ± **.305**	.358 ± .381	.261 ± .311	**.745** ± **.314**	α-Precision	9.005	0.672	1.872	0.664	2.838	0.370
.460 ± .385	.244 ± .247	.324 ± .244	.539 ± .347	ApEn						
.052 ± .124	.530 ± .442	.666 ± .391	.098 ± .189	Authenticity	> 10	0.939	> 10	0.896	> 10	0.815
.083 ± .165	.624 ± .417	**.773** ± **.306**	.141 ± .225	Autocorrelation						
.608 ± .435	.253 ± .397	.210 ± .349	.599 ± .425	β-Recall	> 10	0.944	> 10	0.891	> 10	0.807
.660 ± .193	.566 ± .203	.475 ± .102	.603 ± .170	C2ST						
.404 ± .271	.387 ± .258	.320 ± .197	.396 ± .259	CAS						
.595 ± .455	.189 ± .348	.246 ± .366	.678 ± .406	Context-FID	> 10	0.003	> 10	0.001	> 10	0.915
.769 ± .324	.263 ± .390	.165 ± .311	.716 ± .360	Coverage	> 10	0.781	> 10	0.782	> 10	0.623
.158 ± .314	.432 ± .459	.447 ± .439	.115 ± .216	$\mathrm{C_T}$	1.826	0.807	1.682	0.555	2.682	0.751
.731 ± .368	.324 ± .415	.257 ± .353	.696 ± .362	Density	> 10	0.168	> 10	−0.048	> 10	−0.089
.641 ± .278	.504 ± .269	.438 ± .200	.590 ± .257	Detection_GMM	> 10	0.487	> 10	0.338	> 10	0.180
.659 ± .366	.244 ± .361	.209 ± .328	.673 ± .345	Detection_linear	0.101	0.884	0.416	0.521	0.431	0.499
.739 ± .246	.418 ± .300	.333 ± .212	.703 ± .236	Detection_MLP	> 10	0.290	0.337	0.517	0.471	0.243
.530 ± .424	.204 ± .326	.220 ± .331	.600 ± .401	Detection_XGB	0.100	0.968	0.093	0.981	0.113	0.975
.326 ± .274	.286 ± .229	.348 ± .204	.379 ± .252	Discr. score						
.594 ± .438	.188 ± .325	.222 ± .334	.656 ± .405	Distr. metric						
.576 ± .432	.215 ± .341	.266 ± .355	.650 ± .392	FBCA	> 10	0.577	> 10	0.324	> 10	0.587
.433 ± .350	.615 ± .314	.571 ± .303	.395 ± .316	ICD						
.556 ± .423	.286 ± .409	.222 ± .354	.504 ± .417	Impr. precision	> 10	0.530	> 10	0.441	> 10	0.563
.715 ± .339	.290 ± .386	.190 ± .308	.691 ± .344	Improved recall	> 10	0.736	> 10	0.816	> 10	0.584
.697 ± .301	.336 ± .296	.310 ± .273	.683 ± .298	INND						
.616 ± .421	.233 ± .346	.240 ± .328	.649 ± .393	JSD	> 10	0.609	> 10	0.771	> 10	0.378
.602 ± .408	.251 ± .337	.256 ± .319	.638 ± .380	KLD	> 10	0.396	> 10	0.570	> 10	0.281
.382 ± .458	.684 ± .418	.524 ± .436	.231 ± .389	Max-RTS	> 10	0.398	> 10	0.706	> 10	0.461
.600 ± .318	.394 ± .320	.356 ± .299	.600 ± .311	MTop-Div	> 10	−0.006	1.012	0.529	> 10	−0.020
.472 ± .384	.291 ± .344	.197 ± .255	.414 ± .369	NDB	> 10	0.559	> 10	0.614	> 10	0.444
.295 ± .279	.098 ± .160	.075 ± .124	.313 ± .277	NDB-over/under	> 10	0.715	> 10	0.739	> 10	0.597
.713 ± .332	.333 ± .320	.283 ± .259	.689 ± .322	ONND						
.542 ± .258	.384 ± .207	.409 ± .186	.570 ± .229	Predictive score						
.601 ± .388	.300 ± .334	.305 ± .307	.656 ± .348	RTS	8.306	0.373	> 10	0.839	> 10	0.412
.532 ± .440	.158 ± .265	.163 ± .263	.584 ± .428	Sig-MMD						
.307 ± .348	.216 ± .252	.258 ± .256	.352 ± .346	Spatial corr.						
.630 ± .415	.616 ± .383	.506 ± .362	.578 ± .398	STS	> 10	0.256	1.837	0.547	1.979	0.209
.462 ± .382	.243 ± .241	.314 ± .229	.542 ± .342	Temporal corr.						
.551 ± .416	.324 ± .358	.330 ± .339	.586 ± .397	TRTS						
.429 ± .309	.332 ± .228	.415 ± .184	.501 ± .271	TSTR						
.773 ± .274	.448 ± .330	.337 ± .225	.713 ± .278	WCS						
.617 ± .404	.237 ± .330	.233 ± .302	.647 ± .376	WD	> 10	0.503	> 10	0.874	> 10	0.457
1. ± .000	.000 ± .000	.000 ± .000	1. ± .000	DOMIAS	–					

the best measures exhibit a reliability indicator of approximately $r_{\text{rel}} = 0.75$, which leaves room for improvement for future measures. However, r_{rel} has a high standard deviation, meaning that closely positioned measures cannot be ordered definitively. As expected, measures performing well in the categories fidelity and representativeness perform poorly in the categories generalization and privacy. This suggests that at least two measures should be used but not necessarily one for each category. Comparing the embedder-dependent and direct measures, there is no clear indication to suggest that directly applied measures are better or worse than embedder-dependent ones.

Considering the consistency results in Table 7, we observe pronounced differences in the impact of the dataset and the random seed. While the measures are overwhelmingly indifferent to randomness, the reliability of a measure depends more (and sometimes heavily) on the dataset. Particularly poor in this regard are *ACS*, *ICD*, and *density*. *JSD*, *KLD*, and *FBCA* stand out as rather consistent measures. Most measures demonstrate mediocre consistency, varying between categories, but without clearly favoring one specifically.

We report the average running times (per dataset) for measures in Table 8 and for embedders in Table 11. Note that the running times for measures do not include any preparation steps such as scaling or embedding. For embedders, these comprise training (where applicable), one inference for the real data, and one for the synthetic data. The fastest to compute measures are *temporal* and *spatial correlation* ($\approx 0s$), which benefit from their radical subsampling approach. The measures on subsequent positions benefit from the separately recorded embedding. With the exception of *autocorrelation* computed for PTB diagnostic ECG ($\approx 660s$), the execution times are negligible up to rank 27, staying below one minute. Especially the more complex, often deep-learning based measures run significantly longer, up to 8 minutes. Naturally, Concat is the fastest embedding, followed by Catch22 typically taking under 2 minutes and TS2Vec taking up to an hour. Still, the tables only reflect part of the picture. Tests are often stopped due to excessive running times, which underestimates the actual value for some measures.

5.2 Embedders Experiment

The results of this experiment are shown in Table 1 on the right. Measured in MAPE, the effects of changing the embedding are remarkable. The smallest MAPE is 0.093, while in 59 out of 61 cases it is over 100%, and in 51 cases it is even over 1000%. PCC is mostly positive and in 42 cases above 0.5. Hence, the scores are often correlated but very different in value. Surprisingly, there is no notable difference between the TS2Vec-to-Catch22 pair (middle columns) and the comparisons with the naive Concat embedding (left and right columns). This may be due to the already big differences between TS2Vec and Catch22. Generally, there appears to be no rule or schema, neither between measures nor between embedder pairs. However, we see that the chosen embedding has an enormous effect on the score of a measure in this experiment.

5.3 Discussion of Key Findings

In our reliability ranking, the measures α*-precision*, *ACS*, *autocorrelation*, and again α*-precision* take a close first place in the categories fidelity, generalization, privacy, and representativeness, respectively. However, relatively high standard deviations make a precision ranking impossible. Some positions like α*-precision*'s position in fidelity are very intuitive, others like *autocorrelation*'s first place in privacy are very surprising, suggesting previously unknown properties of these measures. The side experiment on embedding models shows the influence of the chosen embedder on the measure's score, implying that generators should always be evaluated using the same embedder and motivating dedicated analysis. This motivates further research towards a suitable standardized model.

5.4 Limitations

STEB is mainly limited by the choice and combination of transformations. While it incorporates many diverse transformation designs, their connection is not fully explored and other designs may be better suited to test certain categories. Naturally, this also limits the generalizability of the results with respect to all other potential generation methods. Furthermore, STEB currently tests four categories, while a broader range is necessary for complete coverage of desirable synthetic data properties. For instance, fairness [23] or splitting representativeness into diversity and utility could be investigated in future work. To accommodate new measures in the future, we designed STEB with extensibility in mind. As for the experimental design, not all potentially relevant options can be explored due to the combinatorial explosion of the number of test configurations. However, we are confident to cover a wide and deep array of parameters.

6 Conclusion

Currently, the comprehensive comparison of synthetic TS quality remains challenging due to the hodgepodge of measures, little analysis of their efficacy, and generally lacking standardization in evaluation. To tackle these obstacles, we propose a novel benchmark for evaluating the performance of quality measures for synthetic time series. Our benchmark STEB computes indicators for the reliability and consistency of the scores and tracks the running time for computing each measure. To this end, we employ an array of TS transformations along a modulation path to control data modification. We utilized STEB to compare and rank 41 quantitative measures and found that the choice of TS embedding has significant impact on the measure's score. As STEB will be open-sourced after acceptance, we plan to improve and extend this benchmark with the community. This includes the handling of variable length TS, the addition of other measures, and new transformations. Lastly, it would be interesting to investigate the measures' sensitivity to the sizes of real and synthetic datasets.

References

[1] Ahmed Alaa, Boris Van Breugel, Evgeny S. Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? Sample-level metrics for evaluating and auditing generative models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 290–306. PMLR, 17–23 Jul 2022.

[2] Yihao Ang, Qiang Huang, Yifan Bao, Anthony K. H. Tung, and Zhiyong Huang. Tsgbench: Time series generation benchmark. *Proc. VLDB Endow.*, 17(3):305–318, November 2023.

[3] Hiba Arnout, Johannes Kehrer, Johanna Bronner, and Thomas Runkler. Visual evaluation of generative adversarial networks for time series data. *arXiv preprint*, December 2019.

[4] Christoph Bandt and Bernd Pompe. Permutation entropy: a natural complexity measure for time series. *Physical review letters*, 88(17):174102, 2002.

[5] Serguei Barannikov, Ilya Trofimov, Grigorii Sotnikov, Ekaterina Trimbach, Alexander Korotin, Alexander Filippov, and Evgeny Burnaev. Manifold topology divergence: a framework for comparing data manifolds. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 7294–7305. Curran Associates, Inc., 2021.

[6] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *AAAI-94 Workshop on Knowledge Discovery in Databases*, volume 10, pages 359–370, Menlo Park, California, 1994. The AAAI Press.

[7] R. Bousseljot, D. Kreiseler, and A. Schnabel. Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet, January 1995.

[8] Eoin Brophy, Zhengwei Wang, Qi She, and Tomás Ward. Generative adversarial networks in time series: A systematic literature review. *ACM Comput. Surv.*, 55(10), February 2023.

[9] Luis M. Candanedo, Véronique Feldheim, and Dominique Deramaix. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and Buildings*, 140:81–97, 2017.

[10] Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition. *J. off. Stat*, 6(1):3–73, 1990.

[11] Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. The ucr time series classification archive, October 2018. `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`.

[12] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[13] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint*, June 2017.

[14] Joao Fonseca and Fernando Bacao. Tabular and latent space synthetic data generation: a literature review. *Journal of Big Data*, 10(1):115, 2023.

[15] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[16] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation [Online]*, 101(23):e215–e220, 2000 (June 13).

[17] Fuqiang Gu, Mu-Huan Chung, Mark Chignell, Shahrokh Valaee, Baoding Zhou, and Xue Liu. A survey on deep learning for human activity recognition. *ACM Comput. Surv.*, 54(8), October 2021.

[18] J L Hodges Jr. The significance probability of the smirnov two-sample test. *Arkiv för matematik*, 3(5):469–486, 1958.

[19] Gao Huang, Yang Yuan, Qiantong Xu, Chuan Guo, Yu Sun, Felix Wu, and Kilian Weinberger. An empirical study on evaluation metrics of generative adversarial networks. *arXiv preprint*, August 2018.

[20] Ali Ismail-Fawaz, Maxime Devanne, Stefano Berretti, Jonathan Weber, and Germain Forestier. Establishing a unified evaluation framework for human motion generation: A comparative analysis of metrics. *arXiv preprint*, 2024.

[21] Paul Jeha, Michael Bohlke-Schneider, Pedro Mercado, Shubham Kapoor, Rajbir Singh Nirwan, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. PSA-GAN: Progressive self attention GANs for synthetic time series. In *International Conference on Learning Representations*, 2022.

[22] Shruti Kaushik, Abhinav Choudhury, Pankaj Kumar Sheron, Nataraj Dasgupta, Sayee Natarajan, Larry A Pickett, and Varun Dutt. Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in big data*, 3:4, 2020.

[23] Patrik Joslin Kenfack, Daniil Dmitrievich Arapov, Rasheed Hussain, S.M. Ahsan Kazmi, and Adil Khan. On the fairness of generative adversarial networks (gans). In *2021 International Conference "Nonlinearity, Information and Robotics" (NIR)*, pages 1–7, 2021.

[24] William H. Kruskal and W. Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952.

[25] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[26] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 95–104. Association for Computing Machinery, 2018.

[27] Gregory R. Lee, Ralf Gommers, Filip Waselewski, Kai Wohlfahrt, and Aaron O'Leary. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 2019.

[28] Mark Leznik, Arne Lochner, Stefan Wesner, and Jörg Domaschka. [sok] the great gan bake off, an extensive systematic evaluation of generative adversarial network architectures for time series synthesis. *Journal of Systems Research*, 2(1), 2022.

[29] Xiaomin Li, Vangelis Metsis, Huangyingrui Wang, and Anne Hee Hiong Ngu. Tts-gan: A transformer-based time-series generative adversarial network. In Martin Michalowski, Syed Sibte Raza Abidi, and Samina Abidi, editors, *Artificial Intelligence in Medicine*, pages 133–143, Cham, 2022. Springer International Publishing.

[30] Xiaomin Li, Anne Hee Hiong Ngu, and Vangelis Metsis. Tts-cgan: A transformer time-series conditional gan for biosignal data augmentation. *arXiv preprint*, June 2022.

[31] David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *International Conference on Learning Representations*, Toulon, France, April 2017.

[32] Hang Lou, Siran Li, and Hao Ni. Pcf-gan: generating sequential data via the characteristic function of measures on the path space. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 39755–39781. Curran Associates, Inc., 2023.

[33] Carl H Lubba, Sarab S Sethi, Philip Knaute, Simon R Schultz, Ben D Fulcher, and Nick S Jones. catch22: CAnonical Time-series CHaracteristics: Selected through highly comparative time-series analysis. *Data Mining and Knowledge Discovery*, 33(6):1821–1852, 2019.

[34] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[35] Maggie, Oren Anava, Vitaly Kuznetsov, and Will Cukierski. Web traffic time series forecasting. *Kaggle*, 2017.

[36] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60, 1947.

[37] Casey Meehan, Kamalika Chaudhuri, and Sanjoy Dasgupta. A non-parametric test to detect data-copying in generative models. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3546–3556. PMLR, August 2020.

[38] Daniela Micucci, Marco Mobilio, and Paolo Napoletano. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, 7(10), 2017.

[39] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7176–7185. PMLR, July 2020.

[40] Hao Ni, Lukasz Szpruch, Magnus Wiese, Shujian Liao, and Baoren Xiao. Conditional sig-wasserstein gans for time series generation. *arXiv preprint*, June 2020.

[41] Alexander Nikitin, Letizia Iannucci, and Samuel Kaski. Tsgm: A flexible framework for generative modeling of synthetic time series. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 129042–129061. Curran Associates, Inc., 2024.

[42] Skyler Norgaard, Ramyar Saeedi, Keyvan Sasani, and Assefaw H. Gebremedhin. Synthetic sensor data generation for health applications: A supervised deep learning approach. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1164–1167, 2018.

[43] Kun Ouyang, Reza Shokri, David S. Rosenblum, and Wenzhuo Yang. A non-parametric generative model for human trajectories. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 3812–3817. AAAI Press, 2018.

[44] Marco A. F. Pimentel, Alistair E. W. Johnson, Peter H. Charlton, Drew Birrenkott, Peter J. Watkinson, Lionel Tarassenko, and David A. Clifton. Toward a robust estimation of respiratory rate from pulse oximeters. *IEEE Transactions on Biomedical Engineering*, 64(8):1914–1923, 2017.

[45] Zhaozhi Qian, Rob Davis, and Mihaela van der Schaar. Synthcity: a benchmark framework for diverse use cases of tabular synthetic data. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 3173–3188. Curran Associates, Inc., 2023.

[46] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[47] Eitan Richardson and Yair Weiss. On gans and gmms. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[48] Joshua S Richman and J Randall Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American journal of physiology-heart and circulatory physiology*, 278(6):H2039–H2049, 2000.

[49] Ali Seyfi, Jean-Francois Rajotte, and Raymond Ng. Generating multivariate time series with common source coordinated gan (cosci-gan). *Advances in Neural Information Processing Systems*, 35:32777–32788, 2022.

[50] Sahil Sidheekh, Aroof Aimen, and Narayanan C Krishnan. On characterizing gan convergence through proximal duality gap. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9660–9670. PMLR, 18–24 Jul 2021.

[51] Michael Stenger, André Bauer, Thomas Prantl, Robert Leppich, Nathaniel Hudson, Kyle Chard, Ian Foster, and Samuel Kounev. Thinking in categories: A survey on assessing the quality for time series synthesis. *J. Data and Information Quality*, 16(2), June 2024.

[52] Michael Stenger, Robert Leppich, Ian Foster, Samuel Kounev, and André Bauer. Evaluation is key: a survey on evaluation measures for synthetic time series. *Journal of Big Data*, 11(1):66, 2024.

[53] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations*, 2016.

[54] Raphael Vallat. Antropy, 2024. Version 0.1.8.

[55] Boris van Breugel, Trent Kyono, Jeroen Berrevoets, and Mihaela van der Schaar. Decaf: Generating fair synthetic data using causally-aware generative networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 22221–22233. Curran Associates, Inc., 2021.

[56] Boris van Breugel, Hao Sun, Zhaozhi Qian, and Mihaela van der Schaar. Membership inference attacks against synthetic data through overfitting detection. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 3493–3514. PMLR, 25–27 Apr 2023.

[57] Boris van Breugel and Mihaela van der Schaar. Beyond privacy: Navigating the opportunities and challenges of synthetic data. *arXiv preprint*, April 2023.

[58] Magnus Wiese, Lianjun Bai, Ben Wood, and Hans Buehler. Deep hedging: Learning to simulate equity option markets. *arXiv preprint*, November 2019.

[59] Tianlin Xu, Li Kevin Wenliang, Michael Munn, and Beatrice Acciaio. Cot-gan: Generating sequential data via causal optimal transport. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8798–8809. Curran Associates, Inc., 2020.

[60] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[61] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8980–8987, June 2022.

A Summary of Evaluation Measures

Below, we list each measure analyzed in this work in alphabetical order and briefly describe it. For many of them, a detailed definition can be found in [52]. In any case, we reference the original work introducing each measure as well as other sources used in their (re-)implementation. An asterisk indicates that the measure requires a TS embedding. In this case, we use vectors $\vec{x}, \vec{y}$ to represent TS X, Y. l is the TS length, d the number of feature channels, and δ the embedding dimension.

ACS [29]. Average cosine similarity (ACS) compares all pairs of real TS X and synthetic TS Y with respect to their cosine similarity $\frac{\vec{x}\cdot\vec{y}}{||\vec{x}||||\vec{y}||}$. The vectors $\vec{x}$ and $\vec{y}$ are calculated by aggregating seven different statistics of X and Y, respectively. If the data is labeled, the pairs are only constructed within each class. The final score is computed by averaging the similarities of all pairs.

ApEn [28]. Approximate entropy (ApEn) was proposed by [48] to determine the regularity and complexity of univariate time series. The synthetic data measure is derived by computing the squared difference between the approximate entropy of each channel of the real and synthetic time series. [28] subsample both datasets to speed up the computation due to the quadratic complexity. We chose a sample size of $n = 100$.

Authenticity* [1]. Authenticity seeks to measure the proportion of synthetic instances that are novel relative to the real dataset by comparing the distance of the nearest synthetic neighbor to that of the nearest real one. The distances are computed in a spherical embedding space where samples closer to the origin are meant to be typical instances of the real data distribution.

Autocorrelation [40]. The autocorrelation measure is the squared difference of auto-correlation matrices computed for the real and synthetic dataset, respectively. The auto-correlations are determined for each channel up to lag $\frac{l}{4}$ and averaged across the TS.

C2ST [31]. The classifier 2-sample test (C2ST) generally assesses whether two sets of data points are sampled from the same distribution. Here, this is realized through a DL-based binary classifier $c : X \rightarrow \{0, 1\}$ applied to real data as class 0 and synthetic data as class 1, combined with a hypothesis test on the class predictions. c is trained on D_{train} and the predictions taken from $D_{\text{held\_out}}$ and a portion of D_s.

$\mathbf{C_T}$* [37]. The data copying test targets generator overfitting, that is, it detects synthetic data that are merely minimal variations of real data instances. Using D_{train} and $D_{\text{held\_out}}$, the measure compares the distances between TS from D_{train} and $D_{\text{held\_out}}$ with those between TS from D_{train} and D_s using an hypothesis test. Ideally, these distances should be approximately even in both cases.

CAS [46]. The classifier accuracy score (CAS) is a measure for conditional generative models, meaning it requires labeled data. This method trains a deep classifier separately on D_{S} and D_{train}, yielding two models. Both are evaluated on $D_{\text{held\_out}}$ to see if the accuracies achieved by both models are similar.

Context-FID* [21]. Context-FID is a derivative of the "regular" Fréchet inception distance (FID) popular in image synthesis. The original implementation of Context-FID used the unsupervised representation model proposed by [15], while we separated the embedding step and distance calculation. We replaced the representation model with a newer method, TS2Vec.

Coverage* [39]. This measure counts the number of real data instances $\vec{x}$ for which there is a synthetic instance $\vec{y}$ in its neighborhood and divides it by the size of D_r. Coverage C is defined as

$$C := \frac{1}{|D_r|} \sum_{\vec{x} \in D_r} \mathbf{1}\{\exists \vec{y} \in D_s \colon \vec{y} \in B(\vec{x}, \text{dNN}_k(\vec{x}, D_r))\}. \tag{2}$$

where $\text{dNN}_k(\vec{x}, D)$ is the distance of vector $\vec{x}$ to the kth-nearest neighbor in D and $B(c, r)$ a ball in the vector space with center c and radius r.

Density* [39]. The density measure determines for each synthetic instance in how many neighborhoods of real instances it is located, adds the result up across all synthetic instance, and divides the sum by the neighborhood size times the size of the synthetic dataset. For the embedded real $\vec{x}$ and synthetic $\vec{y}$, density D is given by

$$D := \frac{1}{K\,|D_s|} \sum_{\vec{y} \in D_s} \sum_{\vec{x} \in D_r} \mathbf{1}\{\vec{y} \in B(\vec{x}, \text{dNN}_k(\vec{x}, D_r))\}, \tag{3}$$

where $\text{dNN}_k(\vec{x}, D)$ is the distance of vector $\vec{x}$ to the kth-nearest neighbor in D and $B(c, r)$ a ball in the vector space with center c and radius r.

Detection_MLP* [45]. This is a variant of Discriminative score applied to already embedded data and with a multilayer perceptron (MLP) with depth two and 100 hidden units as discriminator model. Its score is the AUCROC score of classifying real and synthetic data.

Detection_XGB* [45]. This is a variant of Discriminative score applied to already embedded data and with XGBoost classifier as discriminator model. Its score is the AUCROC score of classifying real and synthetic data.

Detection_GMM* [45]. This is a variant of discriminative score applied to already embedded data and with a Gaussian mixture model (GMM) as discriminator. Its score is the AUCROC score of classifying real and synthetic data.

Detection_linear* [45]. This is a variant of discriminative score applied to already embedded data and with a logistic regression classifier as discriminator model. Its score is the AUCROC score of classifying real and synthetic data.

Discriminative score [60]. Uses a post-hoc RNN to classify (i.e., discriminate) original data and synthetic data with the achieved accuracy as score. We re-inplemented the original architecture, but updated the training procedure and standardized it for all DL-based discriminators.

Distributional metric [58]. Distributional metric compares the real to the synthetic data distribution based on their probability mass function. To this end, a binning of the values in each channel across the real dataset on one, and the synthetic dataset on the other side, is performed. Finally, the mean absolute difference between binnings of corresponding channels of real and synthetic datasets is calculated and averaged over all channels.

DOMIAS* [56]. DOMIAS is a data privacy measure centered around a density-based membership inference attack. The attack aims to infer membership by targeting local overfitting of the generative model. This measure was proposed in three variants in the original work. We utilize the best-performing one which uses a block neural auto-regressive flow (BNAF) for density estimation.

FBCA* [49]. Feature-based correlation analysis (FBCA) calculates the discrepancy of correlations of feature vectors extracted from the real and synthetic time series. It applies five different statistics to the two correlation matrices. These are mean absolute error, mean squared error, Frobenius norm, Kendall rank correlation coefficient, and Spearman rank correlation coefficient.

ICD [28] Intra-class distance (ICD) measures the average distance between the generated TS using the dynamic time-warping distance (DTW) [6]. Formally, ICD is defined as

$$\text{ICD} := \frac{\sum_{Y \in D_s} \sum_{Y' \in D_s} \text{DTW}(Y, Y')}{|D_s|^2}. \tag{4}$$

[28] subsample D_s to speed up the computation due to the quadratic complexity. We chose a sample size of $n = 100$, considerably more representative than the original $n = 10$.

Improved precision* [25]. This measure counts and averages the number of generated TS for which there is a real TS in its vicinity. More formally,

$$\text{IP} = \frac{1}{|D_s|} \sum_{\vec{y} \in D_s} \mathbf{1}\{\exists \vec{x} \in D_r \colon \vec{y} \in B(\vec{x}, \text{dNN}_k(\vec{x}, D_r))\}, \tag{5}$$

where $\text{dNN}_k(\vec{x}, D)$ is the distance of vector $\vec{x}$ to the kth-nearest neighbor in D and $B(c, r)$ a ball in the vector space with center c and radius r.

Improved recall* [25].

$$\text{IP} = \frac{1}{|D_r|} \sum_{\vec{x} \in D_r} \mathbf{1}\{\exists \vec{y} \in D_s \colon \vec{x} \in B(\vec{y}, \text{dNN}_k(\vec{y}, D_s))\}. \tag{6}$$

INND [3]. The incomming nearest neighbor distance (INND) calculates the average dynamic time-warping distance (DTW) of any synthetic TS to its nearest real neighbor. Formally,

$$\text{INND} = \frac{1}{|D_s|} \sum_{Y \in D_s} \min_{X \in D_r} \text{DTW}(Y, X). \tag{7}$$

[3] subsample D_R and use only one synthetic TS to monitor training. To evaluate, we chose a sample size of $n = 100$ for both datasets.

JSD* [43]. Generally speaking, the Jensen-Shannon divergence (JSD) measures the dissimilarity between two distributions. In this case, these are the distributions of scalar values in the TS in the real and synthetic datasets, discretized by binning these values.

KLD* [47]. Similar to JSD, the Kullback-Leibler divergence measures the dissimilarity between two distributions. Again, these are the distributions of scalar values in the TS in the real and synthetic datasets, discretized by binning these values.

Max-RTS* [42]. The Maximum real to synthetic similarity (Max-RTS) computes the cosine similarity between the real and synthetic TS closest to each other in the embedding space given by

$$\text{Max-RTS} = \max_{\vec{x} \in D_r, \vec{y} \in D_s} \left\{ \frac{\vec{x} \cdot \vec{y}}{||\vec{x}||_2 \cdot ||\vec{y}||_2} \right\}. \tag{8}$$

MTop-Div* [5]. Manifold topology divergence (MTop-Div) determines the discrepancy between the real and synthetic data distribution topologically. The datasets are interpreted as point clouds and topological concepts are used to assess the similarity of the two clouds.

NDB* [47]. Number os statistically different bins (NDB) assesses the degree to which the modes in the synthetic match those in the real training dataset. The modes are estimated using a K-means clustering, and matches are determined using a two-sample hypothesis test comparing pairs of real and synthetic clusters. As a baseline, the procedure is repeated with real held-out data. The score is the absolute difference between both sums of matching clusters (real-real vs real-synthetic).

NDB-over/under* [37]. This is an adaption of the NDB measure. The main difference is that here, the hypothesis tested is the equality of cluster distributions. That is, the distributions of corresponding real and synthetic clusters should be the same. The goal is to detect under-represented and over-represented data regions. The final score is two-fold: A real number for the number of under-represented clusters, and one for the number of over-represented clusters.

ONND [3]. The outgoing nearest neighbor distance (ONND) calculates the average dynamic time-warping distance (DTW) of any real TS to its nearest synthetic neighbor. Formally,

$$\text{ONND} = \frac{1}{|D_r|} \sum_{X \in D_r} \min_{Y \in D_s} \text{DTW}(X, Y). \tag{9}$$

[3] subsample D_R and D_s to speed up the computation due to the quadratic complexity. We chose a sample size of $n = 100$.

Predictive score [60]. This measure trains a simple forecasting model on the generated time series to conduct one-step-ahead predictions. Then, it evaluates its performance on real data using mean absolute error (MAE) and returns its mean.

RTS* [42] Real-to-synthetic similarity (RTS) compares the average cosine similarity within the real data to the cosine similarity between all real TS and 10 random synthetic TS. More, specifically, the score is given by

$$\text{RTS} = \left| \frac{1}{10 \cdot |D_r|} \sum_{i=1}^{10} \sum_{\vec{x} \in D_r} \frac{\vec{x} \cdot \vec{y}_i}{||\vec{x}||_2 \cdot ||\vec{y}_i||_2} - \binom{|D_r|}{2} \sum_{i \neq j} \frac{\vec{x}_i \cdot \vec{x}_j}{||\vec{x}_i||_2 \cdot ||\vec{x}_j||_2} \right|. \tag{10}$$

Sig-MMD [32]. Computes the maximum mean discrepancy (MMD) between signature features extracted from the real and synthetic dataset, respectively. We use a signature kernel with random fourier features (RFF) map and tensor random projections (TRP) from the KSig library[1].

Spatial correlation [28]. Spatial correlation calculates the squared difference of inter-channel Pearson correlations of multivariate real and synthetic TS. For each time series and on both datasets separately, the correlation coefficient is determined for each pair of channels and averaged within the dataset. To reduce computational cost, both datasets are sampled down to $n = 100$ TS.

STS* [42]. Synthetic-to-synthetic similarity measures the "typical" cosine similarity between embedded synthetic TS. Typical means that for every instance, the distances to only five others are calculated. This measure does not use real data.

Temporal correlation [28]. Computes the channel-wise correlation between observations of each TS using the frequency peaks exposed by the fast Fourier transformation (FFT). Then, calculates the mean squared difference between these correlations in the real and synthetic dataset.

TRTS [13]. "Train on real-test on synthetic" measures how well a model trained on real data performs on the generated data. The absolute difference between the performances on real data versus on synthetic data is used as score. For forecasting, the same model as in predictive score [60] is used.

[1] `https://github.com/tgcsaba/KSig`

TSTR [13]. “Train on synthetic-test on real” is a utility-based measure determining the usefulness of the synthetic data on a downstream task compared to real data. As downstream task, we also choose forecasting and utilize the model used for predictive score [60]. A score is computed by taking the absolute performance difference between the forecaster trained on synthetic data and the one trained on real data, evaluated on held-out real test data. This last piece sets the two measures apart.

Wavelet coherence score [30]. The wavelet coherence score (WCS) computes the mean pairwise wavelet coherence, which is an analysis technique for time and frequency correlation, between real and synthetic TS. To reduce computational cost, both datasets are sampled down to $n = 100$ TS.

WD* [47]. The Wasserstein-1 distance (WD) is a distance function between probability distributions. In this case, it is applied to the scalar values of the TS in the real and synthetic dataset. For each set a discrete 1D distribution is created by binning the values. The WD is applied to the real and synthetic binning.

α-Precision* [1]. This measure is based around a fraction $\alpha \in [0, 1]$ of real time series considered “typical” for this data distribution. A synthetic TS falling into the support of the typical part of the real distribution is considered realistic and faithful. A score is deduced by aggregating the deviation between expected fraction of synthetic TS in the support and the actual fraction over different values for α. The support is determined in a spherical embedding space where samples closer to the origin are meant to be typical instances of the real data distribution.

β-Recall* [1]. Analogous to α-Precision, this measure is based around a fraction $\beta \in [0, 1]$ of synthetic time series considered “typical” for this generator. For each β, the measure determines the fraction of real TS with at least one typical synthetic TS in its vicinity. A score is derived by taking the average of the divergence between the expected fraction and the actual one across the values for β. The distances are computed in a spherical embedding space where samples closer to the origin are meant to be typical instances of the real data distribution.

The following three measures are part of the benchmark, but were not used in any of the experiments:

Distribution visualization [60]. This is a qualitative measure, creating a dot-plot visualizing the distributions of the embedded real and synthetic data. The embedding is computed with t-distributed stochastic neighbor embedding (t-SNE) or principal component analysis (PCA) of a subsample of $n = 1000$ instances each.

Visual assessment [3]. This measure is based on the visual evaluation of generated time series data by plotting a (small) subsample of the synthetic dataset. Ideally, the plots are assessed by multiple domain expert.

Realism score* [25] The measure approximates instance fidelity via the position of an embedded synthetic TS in the real data manifold. The closer the generated TS is to a real TS compared to other real TS, the more realistic the generated TS is. The Euclidean distance is used for distance calculation. Realism is a sample-level variant of improved precision and therefore not adequate for our dataset-level experiments. Instead, we test improved precision.

B Details on Datasets, Embedders, and Randomness

B.1 Datasets

In both experiments, we used the following ten datasets. This selection covers multiple source domains, a wide range of dataset sizes, TS lengths, and TS dimensions, as well as labeled and unlabeled data. In addition, the values themselves are diverse with respect to three statistical characteristics. Please find a summary of key characteristics in Table 2.

Appliances energy The UCI Appliances energy prediction dataset consists of multivariate measurements recorded by sensors in a low-energy building, augmented by weather readings and two random variables [9]. Measurements were taken at 10-minute intervals for approximately 4.5 months. By

Table 2: Summary of dataset statistics. For each dataset, this table includes its domain and size, the length and dimension of the contained TS, the number of classes, the average singular value decomposition (SVD) entropy [54], the average permutation entropy [4], and the average correlation between TS features.

Dataset	Domain	Size	TS length	TS dimension	Classes	∅ SVD entropy	∅ Permutation entropy	∅ Feature correlation
Appliances energy	Smart home	19592	144	28	-	0.265	1.701	0.044
ElectricDevices	Devices	16637	96	1	7	1.392	1.343	-
Exchange rate	Finance	7559	30	8	-	0.044	2.293	0.304
Google stock	Finance	3662	24	6	-	0.271	2.311	0.624
PPG and respiration	Medical	21600	125	5	-	0.409	1.670	-0.106
PTB diagnostic ECG	Medical	57618	1000	15	11	0.271	2.302	0.098
Sine	-	10000	100	2	5	0.348	1.583	0.524
StarLightCurves	Sensor	9236	1024	1	3	0.093	1.059	-
UniMiB SHAR	Motion	11771	151	3	17	1.068	2.229	-0.004
Wikipedia web traffic	Networking	117277	550	1	-	0.497	2.522	-

sliding a window of 144 steps with stride one along the time axis, we create a set of overlapping, individual, multivariate time series.

ElectricDevices This dataset is part of the UCR Time Series Classification Archive [11], which comprises 128 labeled subsets from different domains and with different characteristics. We chose ElectricDevices as the best fit with the other sets in our selection.

Exchange rate A collection of the daily exchange rates for the eight currencies of Australia, British, Canada, Switzerland, China, Japan, New Zealand, and Singapore, respectively, ranging from 1990 to 2016 [26]. We apply the sliding window approach again with stride one.

Google stock This set contains the daily historical Google stocks data from 2004 to 2019 in one continuous, aperiodic sequence with features volume, high, low, opening, closing, and adjusted closing prices [60]. We apply the sliding window approach again with stride one.

PPG and respiration Assembled by the Beth Israel Deaconess Medical Centre(BIDMC), this dataset contains physiological signals and static features extracted from the much larger MIMIC-II matched waveform database [44, 16]. We extracted five dynamic features plus labels from the 45 patients for which they are provided: RESP, PLETH, V, AVR, and II. The sequence length of 125 corresponds to 1s of recordings (Sampling rate 125Hz).

PTB diagnostic ECG The PTB diagnostic ECG database is a collection of 549 15-lead ECGs (i.e., 15 feature channels) for 294 patients, including clinical summaries for each record [7, 16]. We extract subsequences of 1000 steps, which corresponds to 1s-long recordings (Sampling rate 1000Hz). Further, we use the eleven diagnosed conditions as class labels.

Sine This is a self-crafted dataset of time series composed of two sine waves each. The set contains multiple, imbalanced classes which differ in wave amplitude, x-shift, phase length, and phase offset between feature channels.

StarLightCurves This is the second dataset from the UCR Time Series Classification Archive [11]. The TS are labeled.

UniMiB SHAR Researchers from the University of Milano-Bicocca created this mulitvariate dataset by collecting acceleration samples acquired with an Android smartphone [38]. The three features represent X-, Y-, and Z-coordinates. Each instance is labeled with one of 17 activities, which we use as classes.

Wikipedia web traffic This set contains visitation data for over $100,000$ Wikipedia articles [35]. Each of the TS included represents the number of daily views of a different Wikipedia article, starting from July 1st, 2015 up until December 31st, 2016. The data was originally compiled for a competition with training and test data. We, however, only use the train set.

B.2 Embedders

STEB currently offers two non-trivial embedding models, which we also used for our experiments. These are:

TS2Vec is a deep-learning model for time series embedding [61]. The model features a CNN-based encoder consisting of cascading dilated convolutional blocks. Training is conducted via hierarchical contrasting loss, which is crucial to the method's success. The learned sequence representations are aggregated from representations of individual time steps created by the convolution blocks.

Catch22. The second model is the feature extractor catch22 [33]. It computes a diverse set of 24 statistical descriptors of a given univariate time series or one feature channel of a multivariate time series. For the latter, we concatenate the feature vectors of each channel to obtain an embedding for the entire time series, i.e.,

$$X \mapsto \text{catch22}(\vec{c}_0) \parallel \text{catch22}(\vec{c}_1) \parallel \ldots \parallel \text{catch22}(\vec{c}_{d-1}), \tag{11}$$

where d is the number of channels.

B.3 Randomness

In our experiments, randomness plays a role at different stages, mainly in splitting the real dataset after preprocessing, while training the TS2Vec embedder, and during the execution of measures. During one test run, we use the same random seed, specified in the test parametrization, at every step of the test to ensure reproducibility. For the *Main* experiment, the ten seeds tried are 42, 461900, 854324, 679123, 107460, 952343, 580127, 893234, 560239, and 501932. Due to time constraints, the *Embedders* experiment is limited to a subset of five seeds, namely 952343, 580127, 893234, 560239, and 501932.

C Additional Benchmark Details

Below, we describe the supporting components of STEB in more detail. Their context within STEB and the relationship between components is visualized in Figure 2.

Configuration & Management. This component guides and coordinates the experiment, starting with loading and validating the configuration, creating the parameter sets of the tests to run, and initiating the test processing. It monitors the execution and triggers recovery if necessary. Experiments can be flexibly executed in two modes of operation: sequential and parallel. In sequential mode, the tests are processed one after the other; caching and logging are done in the file system. The advantage is low computational overhead and few additional dependencies. In parallel mode, the component spawns a user-specified number of worker instances that select tests from a pool and process them in parallel. Users can also limit CPU and RAM use and choose between GPU-enabled and CPU-only workers. Technically, these workers are Docker containers, which implies additional dependencies to run STEB but speeds up the processing and facilitates (dependency) isolation of the measures to be evaluated.

Storage. In the simple sequential mode, everything is stored in a workspace in the file system. In parallel mode, the monitoring, logging, caching, handling of results, and evaluation is optimized using a MongoDB[2] database. This is also more user-friendly, as database monitoring tools support easy, visual tracking of the experiment progress, particularly failed tests.

Caching. Many artifacts such as processed datasets, trained models, or distance matrices produced by preprocessing, embedders, and measures, are duplicated across the modulation steps inside a test and throughout the different tests of an experiment. To speed up the running time, conserve valuable resources, and save energy, caching can be enabled. When it is enabled, each artifact is created once, stored away, and loaded whenever needed—provided that the test parameters match.

Recovery. Since individual tests sometimes fail, workers crash, and experiments are interrupted, this component handles the return to a valid program state. This includes seamlessly continuing experiments, automatically restarting workers, and cleaning up inconsistent caches.

[2] `https://www.mongodb.com`

Table 3: Summary of the expected measure behaviors per category in each test. We differentiate four major behaviors: The score improves ($\nearrow$), worsens ($\searrow$), remains constant (c), and not applicable (-). Transformations marked with * are preceded by a shuffle of the input dataset, even for $\kappa = 0$.

	Fidelity	Generalization	Privacy	Representativeness
Label corruption	$\searrow$	c	-	$\searrow$
Gaussian noise*	$\searrow$	$\nearrow$	$\nearrow$	$\searrow$
Misaligning channels*	$\searrow$	c	$\nearrow$	$\searrow$
Mode dropping*	c	c	$\nearrow$	$\searrow$
Mode collapse*	c	c	$\nearrow$	$\searrow$
Moving average*	$\searrow$	$\nearrow$	$\nearrow$	$\searrow$
Rare event sensitivity*	c	c	$\nearrow$	$\searrow$
Reverse substitution*	c	$\searrow$	$\searrow$	c
Salt & Pepper*	$\searrow$	$\nearrow$	$\nearrow$	$\searrow$
Segment leaking*	$\searrow$	$\searrow$	$\searrow$	$\searrow$
STL*	$\searrow$	$\nearrow$	$\nearrow$	$\searrow$
Substitution*	c	$\nearrow$	$\nearrow$	c
Wavelet transform*	$\searrow$	$\nearrow$	$\nearrow$	$\searrow$

Logging. This component logs various aspects of the experiment execution, informing the user and providing transparency. It records the parameter set for each test and tracks the live status of individual tests, which can be waiting (*todo*), *ongoing*, *successful*, or *failed*. Additionally, status information is recorded, such as the reason for failure. Furthermore, the beginning and end of each test processing are logged and (optionally) the running times for each embedder and measure invocation are saved.

D Details of the Evaluation Procedure

In the two subsections below, we outline the mathematical definitions of reliability r_{rel} and consistency r_{con}. An overview of the expected behavior of each measure used to calculate r_{rel} is provided in Table 3.

D.1 Calculating the Reliability Indicator

Below, we provide the formal definition of $r_{\text{rel}}(t)$ for task t with scores $s_0, \ldots, s_{k-1}$.

Improve, real.

$$r_{\text{rel}}(t) = \frac{2}{k(k-1)} \sum_{i=0}^{k-2} \sum_{j=i+1}^{k-1} \mathbf{1}\{s_i < s_j\} \tag{12}$$

Improve, boolean. Let $w^t = (w_0^t, \ldots, w_{k-1}^t)^T$, $w^f = (w_0^f, \ldots, w_{k-1}^f)^T$ with $w_i^t = i, w_i^f = k - i - 1$ be weight vectors to assign points to the scores based on their position on the modulation path. We define

$$r_{\text{rel}}(t) = \frac{r_{\text{nominal}} - r_{\text{min}}}{r_{\text{max}} - r_{\text{min}}} \tag{13}$$

for

$$r_{\text{nominal}} = \sum_{i=0}^{k-1} w_i^t \cdot \mathbf{1}\{s_i\} + w_i^f \cdot \mathbf{1}\{\neg s\} \tag{14}$$

and

$$r_{\text{min}} = \sum_{i=0}^{\lceil \frac{k}{2} \rceil - 1} w_i^t + \sum_{i=\lceil \frac{k}{2} \rceil}^{k-1} w_i^f, \quad r_{\text{max}} = \sum_{i=0}^{\lceil \frac{k}{2} \rceil - 1} w_i^f + \sum_{i=\lceil \frac{k}{2} \rceil}^{k-1} w_i^t. \tag{15}$$

Worsen, real. Analogous to *improve* with $>$.

Worsen, boolean. Analogous to *improve* with $w_i^t = k - i - 1, w_i^f = i$ and

$$r_{\text{min}} = \sum_{i=0}^{\lceil \frac{k}{2} \rceil - 1} w_i^f + \sum_{i=\lceil \frac{k}{2} \rceil}^{k-1} w_i^t, \quad r_{\text{max}} = \sum_{i=0}^{\lceil \frac{k}{2} \rceil - 1} w_i^t + \sum_{i=\lceil \frac{k}{2} \rceil}^{k-1} w_i^f. \tag{16}$$

Constant, real. Let μ be the median score of m on t and $\epsilon = 0.05$. We define

$$r_{\text{rel}}(t) = \frac{1}{k-1} \left| \left\{ (s_i, \mu) \mid \left| \frac{|\mu - s_i|}{\mu} \right| \leq \epsilon \wedge s_i \neq \mu \right\} \right| \tag{17}$$

Constant, boolean. Similar to real-valued s_i, for boolean values, we have

$$r_{\text{rel}}(t) = \frac{1}{k-1} \left| \{ (s_i, s_{i+1}) \mid \text{XNOR}(s_i, s_{i+1}), 0 \leq i < k-1 \} \right|. \tag{18}$$

Example. Assume a very small experiment of arbitrary measure m with just two tests t_0, t_1, resulting in eleven real-valued scores each,

$$s = [0, 0, 1, 2, 4, 3, 5, 6, 7, 8, 7] \quad \text{and} \quad \sigma = [2.8, 3.0, 2.9, 2.9, 3.0, 3.0, 3.1, 3.0, 3.2, 3.1, 3.0]. \tag{19}$$

s was calculated for transformation misalignment, σ for mode dropping. The value at position 0 was calculated with $\kappa = 0$, position 1 with $\kappa = 0.1$, etc. We will determine r_{rel} for category fidelity. According to Table 3, we expect diminishing fidelity as the misalignment of channels increases, i.e., s to get worse. Hence, we follow paragraph **Worsen, real** and calculate

$$r_{\text{rel}}(t_0) = \frac{2}{11(11-1)} \sum_{i=0}^{11-2} \sum_{j=i+1}^{11-1} \mathbf{1}\{s_i > s_j\} = 0.036. \tag{20}$$

On the other hand, we expect approximately constant fidelity of the remaining samples even if modes collapse. We calculate

$$r_{\text{rel}}(t_1) = \frac{1}{11-1} \left| \left\{ (s_i, 3) \mid \left| \frac{|3 - s_i|}{3} \right| \leq 0.05 \wedge s_i \neq 3 \right\} \right| = 0.8 \tag{21}$$

based on **Constant, real**. Hence,

$$r_{\text{rel}} = \frac{r_{\text{rel}}(t_0) + r_{\text{rel}}(t_1)}{2} = 0.418, \tag{22}$$

attesting m a mediocre to bad reliability in fidelity.

D.2 Calculating the Consistency Indicator

Now, the computation of consistency r_{con} is as follows, starting with the consistency regarding a changing random seed. First, group the $r_{\text{rel}}(t)$ by random seed for all t in the experiment. Assuming there are n random seeds, we have $G_0, \ldots, G_{n-1}$. Then, we apply the Kolmogorov–Smirnov test for two samples[18] to pairs $(G_i, G_j), i < j$, count the pairs which were identified as following the same distribution, and normalize it by the number of pairs:

$$r_{\text{con}} = \frac{2}{n(n-1)} \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} \mathbf{1}\{\text{ks\_2sample}(G_i, G_j)\}. \tag{23}$$

The consistency w.r.t. a changing dataset is analogous, just replace "random seed" by "dataset" above.

E Details of the Experiments

In this section, we provide further details on the experiments conducted, *Main* and *Embedders*. Listing 1 contains the configuration for *Main*. Parameters in the lower part are used for management purposes and influence only how the tests are executed, not what they do. Statistics computed for both experiments, including the number of tests and their success rate, can be found in Table 4. The reasons for failed tests, on the other hand, are listed in Table 5. They range from excessive running time and (GPU) memory overflow to measure-specific errors that can occur in its normal operation.

Listing 1: This is the config file for the *Main* experiment. It specifies the setup of an experimental run, including which datasets to use (here: all ten), the transformations, embedders, and measures. Transformations can be nested once, i.e., they can be chained within one test and are sequentially applied. For more details, especially the other parameters, please see the documentation of the referenced repository.

```
name: main
datasets: ALL
transformations: [
  [shuffle, gn_moderate],
  [shuffle, salt_and_peper_noise],
  [shuffle, misalignment],
  [shuffle, substitution],
  [shuffle, mode_dropping],
  [shuffle, mode_collapse],
  [shuffle, reverse_sub_clean],
  corrupt_labels,
  [shuffle, segment_leaking],
  [shuffle, rare_event_drop],
  [shuffle, moving_average],
  [shuffle, stl_decomposition],
  [shuffle, wavelet_transform],
]
embedders: [ts2vec]
measures: [
  icd, ap_en, innd, onnd, spatial, temporal, c_t, sts, max_rts, rts,
  jsd, kld, wd_on_pmf, auto_corr, wcs, ndbou, ndb, m_top_div,
  Coverage, Density, improved_precision, improved_recall,
  distributional_metric, context_fid, discriminative, predictive,
  detection_mlp, detection_xgb, detection_gmm, detection_linear,
  tstr, trts, cas, c2st, alpha_precision, beta_recall, authenticity,
  acs, fbca, domias, sig_mmd
]
seeds: [42, 461900, 854324, 679123, 107460,
        952343, 580127, 893234, 560239, 501932]
data_feeds: [feed_train]
use_cache: true
workers: {cpu: [[4, 100], [4, 100]],
          gpu: [[4, 100], [4, 100], [4, 100]]}
use_database: true
rebuild_image: false
record_runtime: true
restart_failed: false
test_time_limit: 120
compare_results_to: {embedding: []}
```

F Complementary Results

Additional results for the experiments can be found here across different tables and figures. In Table 6, we provide a reliability ranking of the measures in the four categories. This is an alternative presentation of the information in Table 1, which is alphabetically sorted. Here, we see more clearly how close the reliability indicators of neighboring measures are. For lack of space in the main body of the paper, we report the consistency indicators in Table 7. They are discussed in Section 5.1. Similarly, the average running times recorded for measure executions are placed in Table 8, those for embedding procedures in Table 11. There are also long versions for both tables, Table 9, Table 10, and Table 12, which additionally include the standard deviation, number of valid measurements, and number of invalid measurements for each measure/embedder-dataset-combination.

Additionally, we used STEB's evaluation component to conducted a statistical analysis of the reliability indicators in each of the four categories. To this end, the Kruskal-Wallis H test [24] is employed as omnibus test to determine if there are statistical differences between any of the indicators and the Mann-Whitney U test [36] with Bonferroni correction is applied post-hoc to each pair of measures. The results are visualized in four critical difference diagrams [12], Figures 3 through 6.

Table 4: Statistics for the experiments: For each measure, this table lists the total number of tests attempted, the number of successful tests, and the success rate. The totals across all measures is provided at the table bottom. Only embedding-dependent measures were tested in the *embedders* experiment, entries for other measures are marked as not applicable (N/A).

		Main			Embedders	
Measure	#Total	#Succ	%Succ	#Total	#Succ	%Succ
α-Precision	1020	998	98	1190	1149	97
β-Recall	1020	1000	98	1171	1119	96
C_T	1020	988	97	1093	922	84
ACS	1020	1000	98		N/A	
ApEn	1020	1000	98		N/A	
Authenticity	1020	998	98	1197	1152	96
Autocorrelation	1020	1000	98		N/A	
C2ST	1020	996	98		N/A	
CAS	631	619	98		N/A	
Context-FID	1020	999	98	1159	895	77
Coverage	1020	1000	98	1100	1073	98
DOMIAS	1020	1	0	1038	89	9
Density	1020	999	98	1088	1064	98
Detection_GMM	1020	998	98	1061	921	87
Detection_MLP	1020	996	98	1190	1136	95
Detection_XGB	1020	1000	98	1083	1042	96
Detection_linear	1020	998	98	1086	1063	98
Discriminative score	1020	996	98		N/A	
Distr. metric	1020	999	98		N/A	
FBCA	1020	1000	98	1440	1424	99
ICD	1020	1000	98		N/A	
INND	1020	1000	98		N/A	
Improved precision	1020	999	98	1184	957	81
Improved recall	1020	999	98	1186	967	82
JSD	1020	999	98	1078	1056	98
KLD	1020	996	98	1060	1038	98
MTop-Div	1020	1000	98	1209	1160	96
Max-RTS	1020	999	98	1164	1040	89
NDB	1020	991	97	1071	1014	95
NDB-over/under	1020	924	91	1076	1053	98
ONND	1020	1000	98		N/A	
Predictive score	1020	999	98		N/A	
RTS	1021	999	98	1158	1112	96
STS	1020	998	98	1074	1053	98
Sig-MMD	1020	691	68		N/A	
Spatial correlation	1020	998	98		N/A	
TRTS	1020	992	97		N/A	
TSTR	1020	998	98		N/A	
Temporal correlation	1020	1000	98		N/A	
WCS	1020	880	86		N/A	
WD	1020	997	98	1078	1058	98
Total	41432	39044	94	27234	24557	90

Table 5: Reasons for test failure and the number of such failures for each experiment. Note that the overall number of tests also varies from experiment to experiment.

Failure	Main	Embedders
CUDA Out of Memory	1061	1369
Memory limit of 100 GB exceeded	7	123
Time limit of 120 minutes exceeded	1037	707
NDB-over/under: Too many cells in partition/No samples in a cell	81	0
Other CUDA/CUDNN Runtime error	0	0
Non-CUDA Runtime error	0	0
DOMIAS: BNAF density estimator produced illegal values.	191	114
C_T: Cell x is missing test or training samples.	9	105
Context-FID: Imaginary component in fréchet distance calculation	0	249
Detection_GMM: Fitting the mixture model failed	0	10
Spatial correlation: Cannot compute Pearson correlation for any of the given samples	2	0

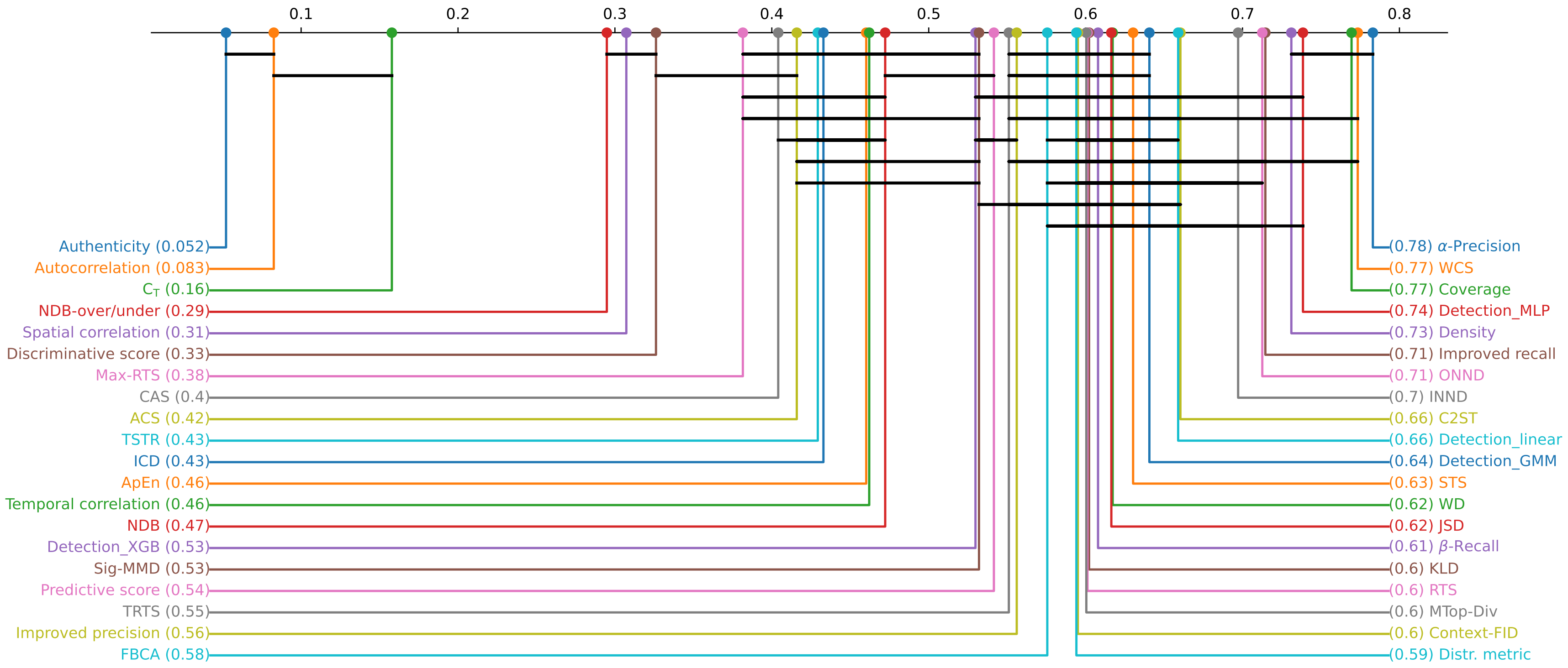

Figure 3: Critical difference diagram for reliability indicator r_{rel} in category fidelity as part of *Main*. The horizontal axis at the top depicts r_{rel}. Additional horizontal bars connect groups of measures with no significantly different r_{rel} value.

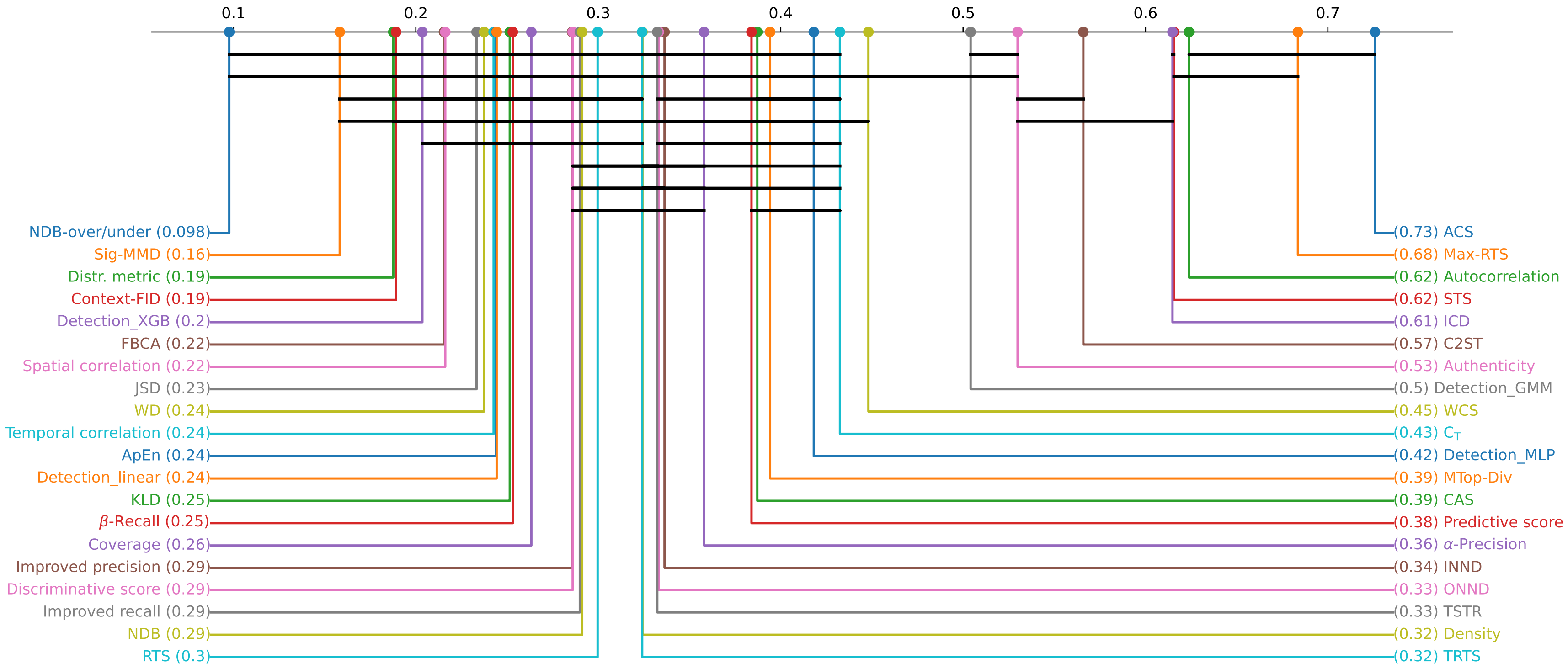

Figure 4: Critical difference diagram for reliability indicator r_{rel} in category Generalization as part of *Main*. The horizontal axis at the top depicts r_{rel}. Additional horizontal bars connect groups of measures with no significantly different r_{rel} value.

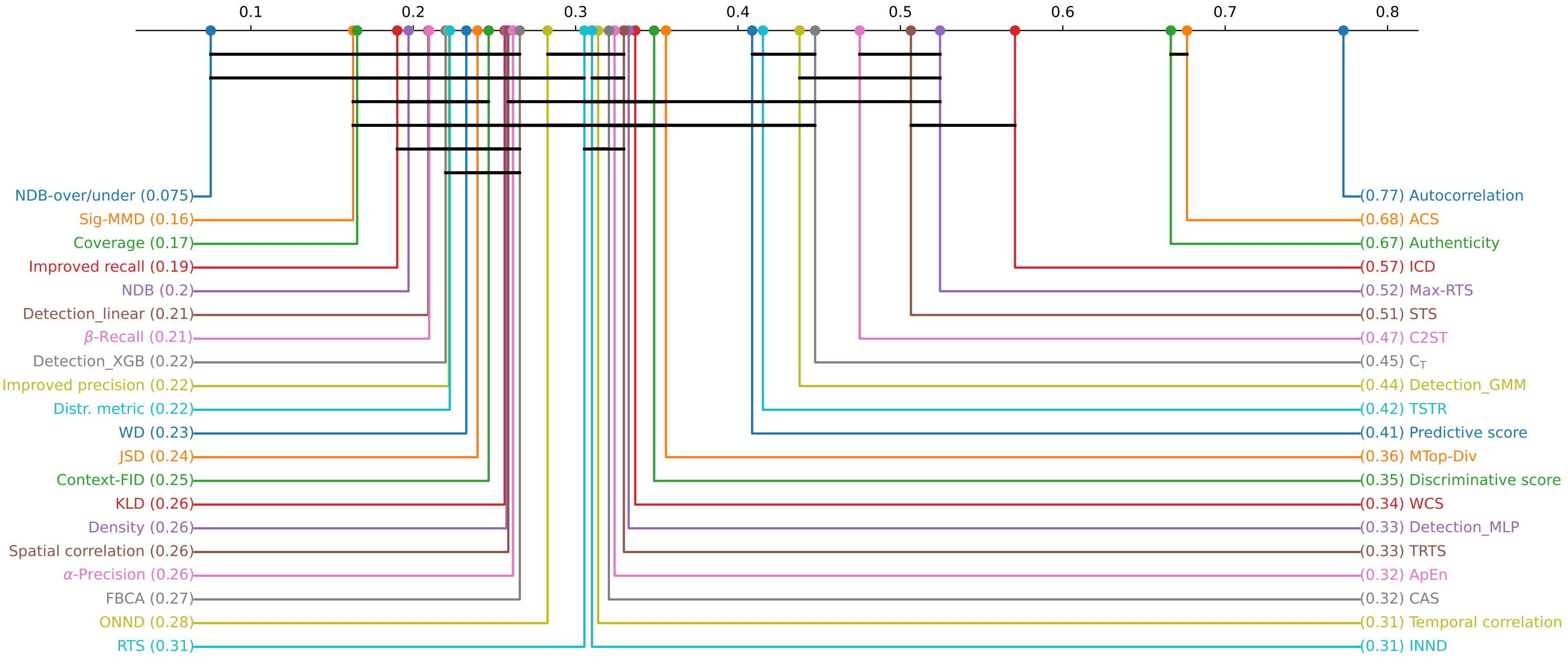

Figure 5: Critical difference diagram for reliability indicator r_{rel} in category privacy as part of *Main*. The horizontal axis at the top depicts r_{rel}. Additional horizontal bars connect groups of measures with no significantly different r_{rel} value.

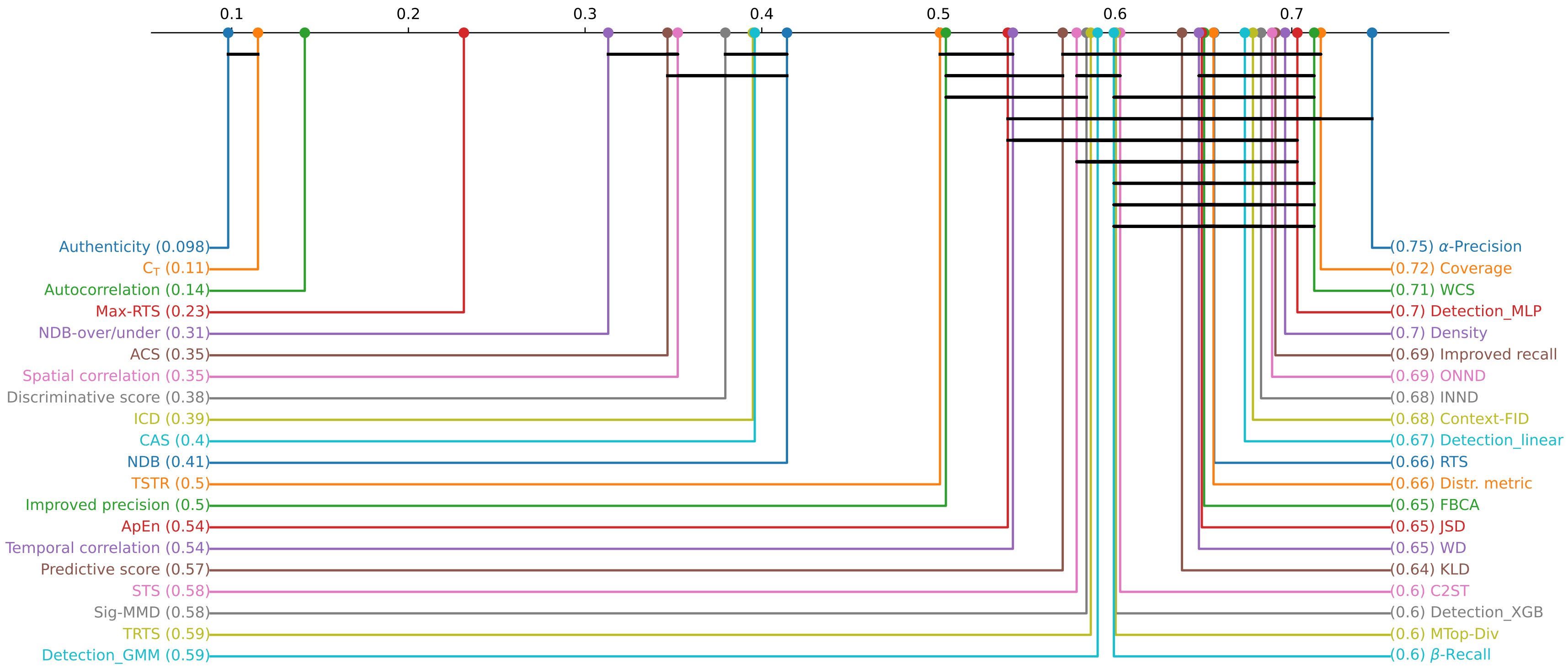

Figure 6: Critical difference diagram for reliability indicator r_{rel} in category Representativeness as part of *Main*. The horizontal axis at the top depicts r_{rel}. Additional horizontal bars connect groups of measures with no significantly different r_{rel} value.

Table 6: Measure reliability ranking for experiment *Main*. This is an alternative presentation of Table 1, ranking the measures in each of the four categories by r_{rel}. For ease of use, mean and standard deviation (Mean ± StD) are provided with each occurrence of the measure. DOMIAS is again excluded and placed at the bottom.

	Fidelity		Generalization		Privacy		Representativeness	
1	α-Precision	.783 ± .305	ACS	.726 ± .320	Autocorrelation	.773 ± .306	α-Precision	.746 ± .314
2	WCS	.774 ± .274	Max-RTS	.684 ± .418	ACS	.676 ± .313	Coverage	.717 ± .360
3	Coverage	.770 ± .323	Autocorrelation	.624 ± .417	Authenticity	.667 ± .391	WCS	.713 ± .278
4	Detection_MLP	.739 ± .246	STS	.616 ± .383	ICD	.571 ± .303	Detection_MLP	.703 ± .236
5	Density	.731 ± .368	ICD	.615 ± .314	Max-RTS	.524 ± .436	Density	.696 ± .362
6	Improved recall	.715 ± .339	C2ST	.565 ± .203	STS	.506 ± .362	Improved recall	.691 ± .344
7	ONND	.713 ± .332	Authenticity	.530 ± .442	C2ST	.475 ± .102	ONND	.689 ± .322
8	INND	.697 ± .301	Detection_GMM	.504 ± .269	C_{T}	.448 ± .439	INND	.683 ± .298
9	C2ST	.660 ± .194	WCS	.448 ± .329	Detection_GMM	.438 ± .200	Context-FID	.678 ± .406
10	Detection_linear	.659 ± .366	C_{T}	.433 ± .459	TSTR	.415 ± .184	Detection_linear	.674 ± .345
11	Detection_GMM	.641 ± .278	Detection_MLP	.418 ± .300	Predictive score	.409 ± .186	RTS	.656 ± .348
12	STS	.630 ± .415	MTop-Div	.394 ± .320	MTop-Div	.355 ± .299	Distr. metric	.656 ± .405
13	WD	.617 ± .404	CAS	.387 ± .258	Discriminative score	.348 ± .204	FBCA	.650 ± .392
14	JSD	.617 ± .421	Predictive score	.384 ± .207	WCS	.337 ± .225	JSD	.649 ± .393
15	β-Recall	.608 ± .435	α-Precision	.358 ± .381	Detection_MLP	.333 ± .212	WD	.647 ± .376
16	KLD	.602 ± .408	INND	.336 ± .296	TRTS	.330 ± .339	KLD	.638 ± .380
17	RTS	.601 ± .388	ONND	.333 ± .320	ApEn	.324 ± .244	C2ST	.603 ± .170
18	MTop-Div	.601 ± .318	TSTR	.332 ± .228	CAS	.320 ± .197	MTop-Div	.601 ± .311
19	Context-FID	.595 ± .455	Density	.324 ± .415	Temporal correlation	.314 ± .229	Detection_XGB	.600 ± .401
20	Distr. metric	.594 ± .438	TRTS	.324 ± .358	INND	.310 ± .273	β-Recall	.599 ± .425
21	FBCA	.576 ± .432	RTS	.300 ± .334	RTS	.305 ± .307	Detection_GMM	.590 ± .257
22	Improved precision	.556 ± .423	NDB	.291 ± .343	ONND	.283 ± .259	TRTS	.586 ± .397
23	TRTS	.551 ± .416	Improved recall	.290 ± .386	FBCA	.266 ± .355	Sig-MMD	.584 ± .428
24	Predictive score	.542 ± .258	Discriminative score	.286 ± .229	α-Precision	.261 ± .311	STS	.578 ± .398
25	Sig-MMD	.532 ± .440	Improved precision	.286 ± .409	Spatial correlation	.258 ± .256	Predictive score	.570 ± .229
26	Detection_XGB	.530 ± .424	Coverage	.263 ± .390	Density	.257 ± .353	Temporal correlation	.542 ± .342
27	NDB	.473 ± .384	β-Recall	.253 ± .397	KLD	.256 ± .319	ApEn	.539 ± .347
28	Temporal correlation	.462 ± .382	KLD	.251 ± .337	Context-FID	.246 ± .365	Improved precision	.504 ± .417
29	ApEn	.460 ± .385	Detection_linear	.244 ± .361	JSD	.239 ± .328	TSTR	.501 ± .271
30	ICD	.433 ± .350	ApEn	.244 ± .247	WD	.233 ± .302	NDB	.415 ± .369
31	TSTR	.429 ± .309	Temporal correlation	.243 ± .241	Distr. metric	.222 ± .334	CAS	.396 ± .259
32	ACS	.416 ± .413	WD	.237 ± .330	Improved precision	.222 ± .354	ICD	.395 ± .316
33	CAS	.404 ± .271	JSD	.233 ± .346	Detection_XGB	.220 ± .331	Discriminative score	.379 ± .252
34	Max-RTS	.382 ± .458	Spatial correlation	.216 ± .252	β-Recall	.210 ± .349	Spatial correlation	.352 ± .346
35	Discriminative score	.326 ± .274	FBCA	.215 ± .341	Detection_linear	.209 ± .328	ACS	.347 ± .369
36	Spatial correlation	.307 ± .348	Detection_XGB	.204 ± .326	NDB	.197 ± .255	NDB-over/under	.313 ± .277
37	NDB-over/under	.295 ± .280	Context-FID	.189 ± .348	Improved recall	.190 ± .308	Max-RTS	.231 ± .389
38	C_{T}	.158 ± .314	Distr. metric	.188 ± .325	Coverage	.165 ± .311	Autocorrelation	.141 ± .225
39	Autocorrelation	.083 ± .165	Sig-MMD	.158 ± .265	Sig-MMD	.163 ± .263	C_{T}	.115 ± .216
40	Authenticity	.052 ± .124	NDB-over/under	.098 ± .160	NDB-over/under	.075 ± .124	Authenticity	.098 ± .189
-	DOMIAS	1. ± .000	DOMIAS	.000 ± .000	DOMIAS	.000 ± .000	DOMIAS	1. ± .000

Table 7: Measure Consistency indicators for experiment *Main*. For each measure, we list r_{con} computed for both changing dataset and random seed. The lower r_{con}, the more the measure scores sway with the choice of parameter. 1.0 means equal reliability on all datasets/random seeds.

	Fidelity		Generalization		Privacy		Representativeness	
Measure	Dataset	Seed	Dataset	Seed	Dataset	Seed	Dataset	Seed
α-Precision	.533	1.	.400	1.	.400	1.	.489	1.
β-Recall	.422	1.	.444	1.	.511	1.	.444	1.
C_T	.467	1.	.311	1.	.311	1.	.467	1.
ACS	.133	1.	.267	1.	.133	1.	.111	1.
ApEn	.244	1.	.333	1.	.400	1.	.400	1.
Authenticity	.600	1.	.400	1.	.444	1.	.422	1.
Autocorrelation	.889	1.	.400	1.	.356	1.	.511	1.
C2ST	.422	1.	.289	1.	.444	1.	.511	1.
CAS	.200	1.	.400	1.	.400	1.	.400	1.
Context-FID	.467	1.	.822	1.	.578	1.	.711	1.
Coverage	.200	1.	.289	1.	.444	1.	.289	1.
Density	.244	1.	.178	1.	.200	1.	.244	1.
Detection_GMM	.422	.689	.622	.533	.800	.533	.311	.689
Detection_MLP	.333	1.	.578	1.	.756	.978	.289	1.
Detection_XGB	.444	1.	.511	1.	.578	1.	.422	1.
Detection_linear	.156	1.	.511	1.	.511	1.	.178	1.
Discriminative score	.422	1.	.333	1.	.333	.956	.378	.978
Distr. metric	.511	1.	.733	1.	.600	1.	.511	1.
FBCA	.556	1.	.778	1.	.689	1.	.756	1.
ICD	.111	1.	.222	1.	.267	1.	.133	1.
INND	.289	1.	.444	1.	.400	1.	.267	1.
Improved precision	.133	1.	.400	1.	.400	1.	.156	1.
Improved recall	.178	1.	.578	1.	.600	1.	.156	1.
JSD	.844	1.	.667	1.	.600	1.	.622	1.
KLD	.822	.956	.800	1.	.933	1.	.711	1.
MTop-Div	.200	1.	.289	1.	.311	1.	.222	1.
Max-RTS	.444	1.	.511	1.	.356	1.	.578	1.
NDB	.222	1.	.511	.933	.422	.933	.244	1.
NDB-over/under	.378	1.	.467	1.	.489	1.	.422	1.
ONND	.333	1.	.467	1.	.533	1.	.333	1.
Predictive score	.178	1.	.511	1.	.667	1.	.422	1.
RTS	.311	1.	.356	1.	.489	1.	.400	1.
STS	.244	1.	.311	1.	.267	1.	.267	1.
Sig-MMD	.472	1.	.417	1.	.444	1.	.528	1.
Spatial correlation	.111	1.	.244	1.	.200	1.	.244	1.
TRTS	.378	1.	.333	1.	.222	1.	.244	1.
TSTR	.489	1.	.578	1.	.667	1.	.822	.978
Temporal correlation	.378	1.	.356	1.	.556	1.	.444	1.
WCS	.306	1.	.472	1.	.500	1.	.361	1.
WD	.711	.978	.622	1.	.556	1.	.556	1.
DOMIAS	N/A							

Table 8: Running time of the measure executions recorded on the *Main* experiment sorted by average rank. The listed values are the average time required to apply the measure to given data across multiple modulation steps and tests. All values are rounded to seconds. Asterisks (*) indicate embedder-dependence, the embedding time is excluded. N/A indicates the absence of successful tests.

	Measure	Appliances energy	ElectricDevices	Exchange rate	Google stock	PPG and respiration	PTB diagnostic ECG	Sine	StarLightCurves	UniMiB SHAR	Wikipedia web traffic
1	Temporal correlation	0	0	0	0	0	0	0	0	0	0
2	Spatial correlation	0	0	0	0	0	0	0	0	0	0
3	Context-FID*	0	0	0	0	0	0	0	0	0	0
4	FBCA*	0	0	0	0	0	0	0	0	0	0
5	Improved recall*	0	0	0	0	0	0	0	0	0	1
6	Improved precision*	0	0	0	0	0	0	0	0	0	1
7	Detection_linear*	0	0	0	0	0	0	0	0	0	1
8	Distr. metric	0	0	0	0	0	3	0	0	0	0
9	JSD*	1	0	0	0	0	1	0	0	0	6
10	WD*	1	1	0	0	0	1	0	0	0	7
11	KLD*	1	1	0	0	0	2	0	0	0	6
12	ACS	2	0	0	0	1	28	0	0	0	7
13	NDB-over/under*	1	0	0	0	1	3	0	0	1	4
14	Sig-MMD	1	1	0	0	1	N/A	0	1	0	2
15	NDB*	1	1	0	0	1	2	0	0	0	9
16	Autocorrelation	5	0	0	0	1	660	0	1	0	12
17	C_T*	1	1	0	0	1	3	0	0	1	6
18	Density*	1	2	0	0	2	3	1	0	1	3
19	ApEn	4	0	1	1	1	16	0	1	0	0
20	Coverage*	1	1	0	0	2	3	1	1	1	3
21	Detection_XGB*	1	2	1	1	2	4	1	1	2	6
22	Max-RTS*	3	3	1	0	3	12	1	1	2	30
23	STS*	4	3	1	1	4	11	2	2	2	21
24	ICD	3	3	3	3	3	11	3	12	3	4
25	INND	3	3	3	3	3	12	3	11	3	4
26	ONND	3	3	3	3	3	11	3	11	3	4
27	RTS*	5	4	2	1	5	18	2	2	3	49
28	Detection_GMM*	8	11	2	1	17	49	4	3	3	71
29	β-Recall*	9	8	4	2	15	33	6	5	6	67
30	α-Precision*	9	9	3	2	14	40	5	5	7	70
31	Authenticity*	8	10	4	2	11	42	5	4	6	77
32	TRTS	25	11	9	4	26	59	12	14	15	88
33	Predictive score	26	10	11	5	24	63	11	13	16	87
34	Discriminative score	25	23	9	5	36	124	16	22	19	216
35	C2ST	22	23	10	5	40	125	16	22	20	219
36	Detection_MLP*	34	32	12	6	43	85	18	17	20	157
37	MTop-Div*	100	38	54	40	40	42	38	42	40	40
38	TSTR	46	20	19	8	51	122	23	25	31	168
39	WCS	430	9	26	18	54	N/A	19	130	48	77
40	CAS	N/A	80	N/A	N/A	N/A	466	30	41	90	N/A
41	DOMIAS*	N/A									

Table 9: Running time statistics of the measure executions extending Table 8 for *Appliances energy*, *ElectricDevices*, *Exchange rate*, *Google stock*, and *PPG and respiration*. In addition to the average running time (Mean), we provide for each dataset and measure the standard deviation (StD) of the measurements, the number of complete, un-aided executions (Valid), and the cache-aided executions (Cached). Aided means that the execution was accelerated by the use of previously computed and cached artifacts. Running times for DOMIAS not available.

	Measure	Appliances energy				ElectricDevices				Exchange rate				Google stock				PPG and respiration			
		Mean	StD	Valid	Cached	Mean	StD	Valid	Cached	Mean	StD	Valid	Cached	Mean	StD	Valid	Cached	Mean	StD	Valid	Cached
1	Temporal correlation	0	0	20	970	0	0	20	1190	0	0	20	970	0	0	20	970	0	0	20	970
2	Spatial correlation	0	0	20	970	0	0	20	1190	0	0	20	970	0	0	20	948	0	0	20	970
3	Improved recall	0	0	990	0	0	0	1210	0	0	0	990	0	0	0	990	0	0	0	990	0
4	Context-FID	0	0	990	0	0	0	1210	0	0	0	990	0	0	0	990	0	0	0	990	0
5	FBCA	0	0	20	970	0	0	20	1190	0	0	20	970	0	0	20	970	0	0	20	970
6	Improved precision	0	0	20	970	0	0	20	1190	0	0	20	970	0	0	20	970	0	0	20	970
7	Distr. metric	0	0	20	970	0	0	20	1190	0	0	20	970	0	0	20	970	0	0	20	970
8	Detection_linear	0	0	990	0	0	0	1210	0	0	0	990	0	0	0	990	0	0	0	990	0
9	JSD	1	0	20	959	0	0	20	1190	0	0	20	970	0	0	20	970	0	0	20	970
10	ACS	2	0	990	0	0	0	1210	0	0	0	990	0	0	0	990	0	1	0	990	0
11	Autocorrelation	5	2	20	970	0	0	20	1190	0	0	20	970	0	0	20	970	1	0	20	970
12	WD	1	0	20	948	1	0	20	1190	0	0	20	970	0	0	20	970	0	0	20	970
13	KLD	1	0	20	970	1	0	20	1190	0	0	20	970	0	0	20	970	0	0	20	970
14	NDB	1	0	20	970	1	0	20	1190	0	0	20	970	0	0	20	970	1	0	20	970
15	Sig-MMD	1	0	737	0	1	0	1210	0	0	0	990	0	0	0	990	0	1	0	473	0
16	NDB-over/under	1	0	20	849	0	0	20	1190	0	0	20	871	0	0	20	849	1	0	20	948
17	$\mathrm{C_T}$	1	0	20	937	1	0	20	1190	0	0	20	970	0	0	20	904	1	0	20	970
18	Density	1	0	10	980	2	1	8	1202	0	0	11	979	0	0	9	981	2	1	11	979
19	Coverage	1	0	10	980	1	1	12	1198	0	0	9	981	0	0	11	979	2	1	9	981
20	ApEn	4	0	20	970	0	0	20	1190	1	0	20	970	1	0	20	970	1	0	20	970
21	Detection_XGB	1	1	990	0	2	0	1210	0	1	0	990	0	1	0	990	0	2	0	990	0
22	Max-RTS	3	4	990	0	3	3	1210	0	1	1	990	0	0	1	990	0	3	4	990	0
23	STS	4	1	968	0	3	1	1210	0	1	0	990	0	1	0	990	0	4	1	990	0
24	ICD	3	0	990	0	3	0	1210	0	3	0	990	0	3	0	990	0	3	0	990	0
25	INND	3	0	990	0	3	0	1210	0	3	0	990	0	3	0	990	0	3	0	990	0
26	RTS	5	1	20	970	4	1	20	1190	2	0	20	970	1	0	20	970	5	1	20	959
27	ONND	3	0	990	0	3	1	1210	0	3	0	990	0	3	1	990	0	3	1	990	0
28	Detection_GMM	8	7	990	0	11	7	1199	0	2	2	990	0	1	1	990	0	17	10	990	0
29	α-Precision	9	2	8	982	9	2	6	1204	3	1	6	984	2	0	9	981	14	6	4	986
30	β-Recall	9	2	5	985	8	2	4	1206	4	1	8	982	2	0	6	984	15	5	8	982
31	Authenticity	8	2	7	972	10	2	10	1200	4	1	6	984	2	0	5	985	11	2	8	982
32	TRTS	25	5	20	970	11	2	20	1190	9	2	20	970	4	1	20	970	26	7	20	970
33	Predictive score	26	11	979	0	10	2	1210	0	11	4	990	0	5	3	990	0	24	6	990	0
34	C2ST	22	6	979	0	23	14	1210	0	10	6	990	0	5	2	990	0	40	28	990	0
35	Discriminative score	25	9	990	0	23	12	1210	0	9	5	979	0	5	2	990	0	36	27	990	0
36	Detection_MLP	34	23	990	0	32	21	1199	0	12	7	990	0	6	3	990	0	43	27	979	0
37	TSTR	46	9	20	970	20	4	20	1190	19	4	20	970	8	1	20	970	51	10	20	970
38	WCS	430	5	880	0	9	2	1210	0	26	2	990	0	18	2	990	0	54	6	990	0
39	MTop-Div	100	90	990	0	38	7	1210	0	54	62	990	0	40	11	990	0	40	7	990	0
40	CAS	NaN	NaN	NaN	NaN	80	27	20	1300	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 10: Running time statistics of the measure executions extending Table 8 for *PTB diagnostic ECG*, *Sine*, *StarLightCurves*, *UniMiB SHAR*, and *Wikipedia web traffic*. In addition to the average running time (Mean), we provide for each dataset and measure the standard deviation (StD) of the measurements, the number of complete, un-aided executions (Valid), and the cache-aided executions (Cached). Aided means that the execution was accelerated by the use of previously computed and cached artifacts. Running times for DOMIAS not available.

	Measure	PTB diagnostic ECG				Sine				StarLightCurves				UniMiB SHAR				Wikipedia web traffic			
		Mean	StD	Valid	Cached	Mean	StD	Valid	Cached	Mean	StD	Valid	Cached	Mean	StD	Valid	Cached	Mean	StD	Valid	Cached
1	Spatial correlation	0	0	19	1191	0	0	20	1300	0	0	20	1190	0	0	20	1300	0	0	19	751
2	Temporal correlation	0	0	35	1175	0	0	20	1300	0	0	20	1190	0	0	20	1300	0	0	16	754
3	Context-FID	0	0	1210	0	0	0	1320	0	0	0	1199	0	0	0	1320	0	0	0	770	0
4	Improved recall	0	0	1210	0	0	0	1320	0	0	0	1210	0	0	0	1320	0	1	0	759	0
5	FBCA	0	0	25	1185	0	0	20	1300	0	0	20	1190	0	0	20	1300	0	0	16	754
6	Improved precision	0	0	20	1190	0	0	20	1300	0	0	20	1179	0	0	20	1300	1	0	20	750
7	Detection_linear	0	0	1188	0	0	0	1320	0	0	0	1210	0	0	0	1320	0	1	1	770	0
8	Distr. metric	3	1	17	1182	0	0	20	1300	0	0	20	1190	0	0	20	1300	0	0	32	738
9	JSD	1	1	19	1191	0	0	20	1300	0	0	20	1190	0	0	20	1300	6	2	18	752
10	WD	1	0	23	1187	0	0	20	1300	0	0	20	1190	0	0	20	1300	7	6	17	742
11	KLD	2	1	23	1176	0	0	20	1300	0	0	20	1179	0	0	20	1300	6	3	17	731
12	NDB-over/under	3	1	20	750	0	0	20	1300	0	0	20	1179	1	0	20	1300	4	1	18	730
13	ApEn	16	7	20	1190	0	0	20	1300	1	0	20	1190	0	0	20	1300	0	0	20	750
14	Sig-MMD	N/A	N/A	N/A	N/A	0	0	1320	0	1	0	330	0	0	0	1320	0	2	0	231	0
15	Density	3	0	9	1201	1	0	13	1307	0	0	10	1189	1	0	13	1307	3	0	14	756
16	NDB	2	1	17	1171	0	0	20	1300	0	0	20	1168	0	0	20	1300	9	4	18	697
17	Coverage	3	0	11	1199	1	0	7	1313	1	0	10	1200	1	0	7	1313	3	0	20	750
18	ACS	28	26	1210	0	0	0	1320	0	0	0	1210	0	0	0	1320	0	7	30	770	0
19	C_T	3	1	18	1181	0	0	20	1300	0	0	20	1179	1	0	20	1300	6	2	19	740
20	Autocorrelation	660	165	37	1173	0	0	20	1300	1	1	20	1190	0	0	20	1300	12	6	19	751
21	Detection_XGB	4	1	1210	0	1	0	1320	0	1	0	1210	0	2	0	1320	0	6	2	770	0
22	Max-RTS	12	11	1210	0	1	1	1320	0	1	1	1210	0	2	1	1320	0	30	20	759	0
23	STS	11	2	1210	0	2	0	1320	0	2	0	1210	0	2	0	1320	0	21	4	770	0
24	ICD	11	7	1210	0	3	0	1320	0	12	10	1210	0	3	1	1320	0	4	0	770	0
25	ONND	11	8	1210	0	3	0	1320	0	11	10	1210	0	3	1	1320	0	4	0	770	0
26	INND	12	9	1210	0	3	0	1320	0	11	10	1210	0	3	0	1320	0	4	0	770	0
27	RTS	18	5	20	1190	2	0	20	1300	2	0	20	1190	3	1	20	1300	49	19	18	752
28	Detection_GMM	49	25	1210	0	4	3	1320	0	3	1	1199	0	3	1	1320	0	71	36	770	0
29	β-Recall	33	9	8	1202	6	1	7	1313	5	1	7	1203	6	1	9	1311	67	19	7	763
30	Authenticity	42	16	6	1204	5	1	5	1315	4	1	4	1195	6	2	5	1315	77	26	7	763
31	α-Precision	40	7	6	1193	5	1	8	1312	5	1	9	1190	7	1	6	1314	70	16	6	764
32	Predictive score	63	18	1210	0	11	4	1320	0	13	5	1210	0	16	6	1320	0	87	19	770	0
33	TRTS	59	14	14	1119	12	3	20	1300	14	4	20	1179	15	4	20	1300	88	19	17	753
34	MTop-Div	42	6	1210	0	38	14	1320	0	42	7	1210	0	40	6	1320	0	40	5	770	0
35	Detection_MLP	85	45	1210	0	18	11	1320	0	17	12	1144	66	20	14	1320	0	157	107	748	0
36	Discriminative score	124	71	1199	0	16	9	1320	0	22	11	1210	0	19	11	1320	0	216	129	748	0
37	C2ST	125	72	1199	0	16	9	1320	0	22	11	1199	0	20	13	1320	0	219	126	759	0
38	TSTR	122	27	17	1193	23	6	20	1289	25	8	20	1190	31	9	20	1300	168	38	19	740
39	WCS	N/A	N/A	N/A	N/A	19	3	1320	0	130	7	1210	0	48	3	1320	0	77	3	770	0
40	CAS	466	163	17	1292	30	5	20	1410	41	16	20	1300	90	22	20	1410	N/A	N/A	N/A	N/A

Table 11: Running time of the embedder usage recorded on the Main and Embedders experiments sorted by average rank. The listed values are the average time required to employ the embedder across multiple tests, combining inference and training. All values are rounded to seconds.

	Embedding	Appliances energy	ElectricDevices	Exchange rate	Google stock	PPG and respiration	PTB diagnostic ECG	Sine	StarLightCurves	UniMiB SHAR	Wikipedia web traffic
1	Concat	0	0	0	0	0	0	0	0	0	0
2	Catch22	99	2	3	1	14	889	3	12	7	86
3	TS2Vec	327	391	41	38	515	3697	252	640	256	4812

G Measure Selection Guide

Ultimately, STEB analyzes and compares synthetic TS evaluation measures to allow users a better selection of measures. Below, we provide some step-by-step instructions on how this selection process can look like using STEB output.

1. Determine which categories are relevant for the given use case. Usually, one measure is needed per category. For each category, repeat all following steps.
2. Start with the measure ranked highest reliability (see Table 6).
3. If the measure lacks consistency in this category (see Table 7), move on to the next best measure. We suggest a minimum of $r_{\text{con}} = 0.5$.
4. If the use case dictates running time constraints, for instance, because many parametrizations of a generative model must be evaluated during optimization, measures with excessive running time should be skipped (see Table 8 in combination with Table 11). For this purpose, the running time on the dataset most similar in size, TS length, and number of feature channels to the use case's target dataset should be considered.
5. Further, in case ease-of-use is an important selection criterion, it makes sense to skip embedder-dependent measures or only use them in case embeddings can be reused in other categories. Similarly, some measures are prone to errors such as DOMIAS and C_T, which requires additional effort and knowledge to fix (see Table 4 and Table 5).

Based on the experimental results in this paper and assuming running time is no limiting factor, the best measure suit for synthetic TS evaluation currently comprises

- *α-Precision* for fidelity,
- *ACS* for generalization,
- *Autocorrelation* for privacy, and
- *Context-FID* for representativeness.

Table 12: Running time statistics of the embedder usage extending Table 11. In addition to the average running time (Mean), we provide for each dataset and measure the standard deviation (StD) of the measurements, the number of complete, the number of completed, un-aided executions (Valid), and the cache-aided executions (Cached). Aided means that the model training was skipped due to loading a cached pre-trained model.

Dataset	Embedding	Mean	StD	Valid	Cached
Appliances energy	Concat	0	0	4543	0
	Catch22	99	26	7442	0
	TS2Vec	327	125	40	22110
ElectricDevices	Concat	0	0	9099	0
	Catch22	2	2	9855	0
	TS2Vec	391	107	40	27368
Exchange rate	Concat	0	0	4574	0
	Catch22	3	2	7859	0
	TS2Vec	41	16	41	22220
Google stock	Concat	0	0	4193	0
	Catch22	1	2	7221	0
	TS2Vec	38	20	40	22154
PPG and respiration	Concat	0	0	5537	0
	Catch22	14	4	7817	0
	TS2Vec	515	259	40	22286
PTB diagnostic ECG	Concat	0	0	11700	0
	Catch22	889	162	10394	0
	TS2Vec	3697	1035	39	26884
Sine	Concat	0	0	10071	0
	Catch22	3	2	10682	0
	TS2Vec	252	94	40	29920
StarLightCurves	Concat	0	0	9727	0
	Catch22	12	6	10622	0
	TS2Vec	640	364	40	27269
UniMiB SHAR	Concat	0	0	5717	0
	Catch22	7	3	10633	0
	TS2Vec	256	113	41	29909
Wikipedia web traffic	Concat	0	0	7509	0
	Catch22	86	23	5182	0
	TS2Vec	4812	1475	36	17149