Re-framing Time Series Augmentation Through the Lens of Generative Models

Ali Ismail-Fawaz[1] (✉), Maxime Devanne[1], Stefano Berretti[2], Jonathan Weber[1], and Germain Forestier[1,3]

[1] IRIMAS, Université de Haute-Alsace, France
ali-el-hadi.ismail-fawaz@uha.fr , maxime.devanne@uha.fr
jonathan.weber@uha.fr , germain.forestier@uha.fr
[2] MICC, University of Florence, Italy
stefano.berretti@unifi.it
[3] DSAI, Monash University, Australia
germain.forestier@monash.edu

Abstract. Time series classification is widely used in many fields, but it often suffers from a lack of labeled data. To address this, researchers commonly apply data augmentation techniques that generate synthetic samples through transformations such as jittering, warping, or resampling. However, with an increasing number of available augmentation methods, it becomes difficult to choose the most suitable one for a given task. In many cases, this choice is based on intuition or visual inspection. Assessing the impact of this choice on classification accuracy requires training models, which is time-consuming and depends on the dataset. In this work, we adopt a generative model perspective and evaluate augmentation methods prior to training any classifier, using metrics that quantify both fidelity and diversity of the generated samples. We benchmark 22 augmentation techniques on 131 public datasets using eight metrics. Our results provide a practical and efficient way to compare augmentation methods without relying solely on classifier performance. The source code is publicly available:
https://github.com/MSD-IRIMAS/Data-Augmentation-4-TSC

Keywords: Time Series Classification · Data Augmentation · Generative Models

1 Introduction

Time series data are ubiquitous, appearing in a wide range of applications including health monitoring, traffic forecasting, finance, and more. Time Series Classification (TSC) [27, 3] is the task of assigning a class label to an input time series and is central to many domains such as medicine, climate science, and astronomy. This task typically involves training a machine learning model to learn temporal patterns that distinguish one class from another. Deep learning models have been particularly successful in TSC [22, 19, 13, 2, 20, 5] due to their capacity to learn complex representations and efficiently scale via GPU acceleration. However, one of their major drawbacks is overfitting, often exacerbated by the scarcity of labeled time series data. Collecting labeled data for time series is challenging and costly, prompting researchers to explore data augmentation as a strategy to improve model generalization.

Data augmentation [8, 33] involves generating synthetic samples by applying transformations to existing labeled data, with the goal of enriching the training set and improving downstream performance. A wide range of augmentation techniques has been proposed for time series, some adapted from computer vision (e.g., jigsaw, rotation), and others specifically designed to preserve the temporal structure of the data [6, 31]. Most prior work in this area focuses on identifying the "best" augmentation method through empirical comparisons of model performance. Typically, this involves training models with and without augmented data and measuring the difference in classification accuracy. While informative, this approach is computationally expensive and offers no principled way to choose between many candidate methods ahead of time.

In this work, we propose re-framing time series augmentation methods as generative models. To the best of our knowledge, this is the first work to systematically evaluate time series augmentations from this perspective. Rather than relying solely on downstream performance, we assess augmentation methods using a generative evaluation framework based on relevant metrics. Generative models are algorithms that learn from real data to produce new samples that resemble the original distribution. Such models can be evaluated on two key aspects: fidelity and diversity. Fidelity measures how closely the generated data resembles real samples, while diversity ensures the model captures the full variability of the data. Both are essential for assessing the quality of synthetic data. We adopt the evaluation setup of Ismail-Fawaz et al. [17], leveraging eight metrics that quantify the quality of generated data in a pre-trained feature space. This analysis, independent of performance gain evaluation, provides a practical and computationally efficient way to select promising augmentation methods prior to classifier training.

Our analysis, conducted across 22 augmentation techniques evaluated on 131 time series classification datasets [4, 27], shows that Discriminative Guided Warping (DGW) [23], when combined with the Move-Split-Merge (MSM) similarity measure [32, 10], consistently outperforms all other methods in terms of average fidelity. In contrast, the Amplitude Scaling (AS) method ranks first in terms of average diversity.

This evaluation framework lays the foundation for future work, where we aim to investigate whether a causal relationship exists between the generative evaluation metrics and downstream performance gains. Such an extension would help determine whether better scores on fidelity and diversity metrics can predict improved model performance when trained on augmented data.

The rest of this paper is organized as follows: Section 2 presents the augmentation techniques evaluated in this study; Section 3 reports the experimental results across all evaluation metrics; and Section 4 concludes the paper.

2 Methodology

In this section, we first present some useful definitions to facilitate the reading of the rest of this work. Second, we present the seven data augmentation techniques for time series data included in this study.

2.1 Definitions

- A **Univariate Time Series (UTS)** is defined as $\mathbf{x} = \{x_1, x_2, \ldots, x_L\}$, a sequence of L real-valued observations sampled at regular time intervals.
- A **time series classification dataset** is a set $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, where each $\mathbf{x}^{(i)}$ is a UTS and $y^{(i)} \in \{1, \ldots, C\}$ is its corresponding class label, with C the number of distinct classes.
- A **time series augmentation function** is a mapping $\mathcal{F} : \mathbb{R}^L \to \mathbb{R}^L$ that transforms an input series $\mathbf{x}$ into an augmented series $\hat{\mathbf{x}} = \mathcal{F}(\mathbf{x})$, with the aim of increasing diversity, while preserving the original class distribution.

2.2 Time Series Data Augmentation Techniques

In what follows, we detail each of the data augmentation techniques that we use in our study. A detailed visualization of these techniques is presented in Figure 1.

2.2.1 Jittering

Jittering [33] is a data augmentation method for time series that adds random Gaussian noise to the signal. For each time step t, the augmented value is:

$$\hat{x}_t = x_t + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2), \tag{1}$$

where ϵ_t is Gaussian noise. The standard deviation σ is sampled from a uniform distribution, $\sigma \sim \mathcal{U}(0, 0.1)$.

2.2.2 Amplitude Scaling

Amplitude Scaling (AS) [33] modifies a time series by multiplying all values by a random scalar. For each time step t, the augmented value is:

$$\hat{x}_t = x_t \cdot \alpha, \tag{2}$$

where the scaling factor α is drawn from a Gaussian distribution, $\alpha \sim \mathcal{N}(1.0, 0.1^2)$.

2.2.3 Amplitude Warping

Amplitude Warping (AW) [33] modifies a time series by applying a smooth, non-linear scaling function over time, unlike the static scaling in AS (see Section 2.2.2). A random cubic curve is generated and used to vary the amplitude across time steps. For each time step t:

$$\hat{x}_t = x_t \cdot f(t), \tag{3}$$

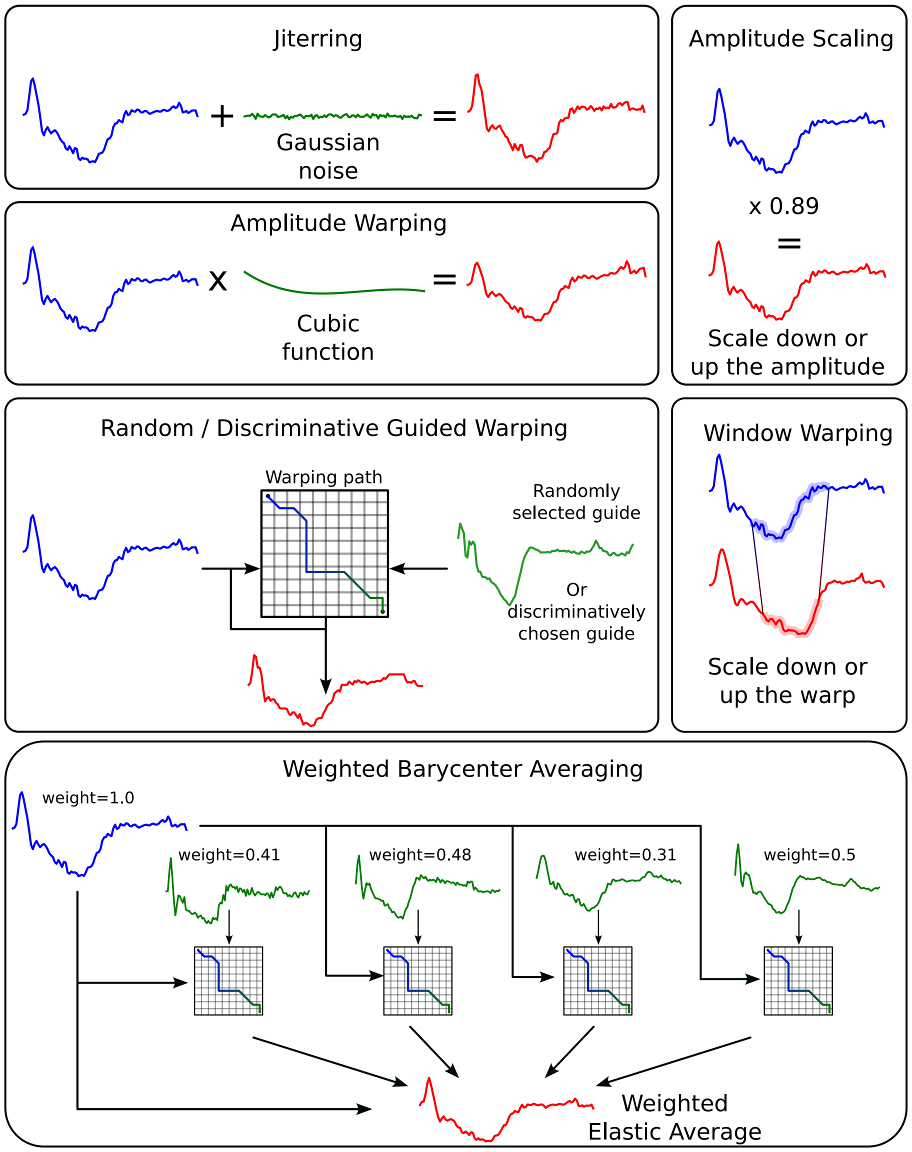


Fig. 1. Visual summary of all data augmentation methods applied to the same input time series (in blue), taken from the ECG200 dataset of the UCR archive [4]. Each subplot shows the transformation produced by a specific augmentation technique with its resulting augmented series (in red).

where $f(t) = a \cdot t^3 + b \cdot t^2 + c \cdot t + d$ is a smooth cubic function whose shape is randomly determined. The function is normalized to remain positive and avoid extreme scaling. This ensures that the overall structure of the signal is preserved. The coefficients of $f(t)$ are sampled uniformly following:

$$a \sim \mathcal{U}(-0.2, 0.2) \quad \& \quad b, c, d \sim \mathcal{U}(-a, a). \tag{4}$$

2.2.4 Window Warping

Window Warping (WW) [25] modifies the time axis of a time series by stretching or compressing a randomly selected window. For each series, a window of size w starting at time step s is randomly selected, and a warp scale α is uniformly sampled from $\mathcal{U}(0.5, 2.0)$.

Given a time series $\mathbf{x}$ of length L, the selected window $\mathbf{x}_{win} = \mathbf{x}_{s:s+w-1} = \{x_s, x_{s+1}, \ldots, x_{s+w-1}\}$ is warped as follows:

$$\hat{\mathbf{x}}_{win} = interp(\mathbf{x}_{win}, \alpha \cdot w), \tag{5}$$

where $interp(\cdot, \ell)$ denotes resampling the input to length ℓ via linear interpolation.
The final augmented series is obtained by resampling the concatenation of the non-warped and warped segments back to length L:

$$\hat{\mathbf{x}} = interp(concat(\mathbf{x}_{1:s-1},\ \hat{\mathbf{x}}_{win},\ \mathbf{x}_{s+w:L}), L). \tag{6}$$

2.2.5 Random Guided Warping

Random Guided Warping (RGW) [23], unlike Window Warping (Section 2.2.4), warps the entire time series based on alignment with another sample. The guiding sample is chosen at random from the dataset, and an alignment path is computed between the two series using an elastic similarity measure. This path is then used to re-index the original series and warp it accordingly.

Let $\mathbf{x}$ be the input series of length L, and $\mathbf{x}'$ a randomly selected reference series. Using a distance metric such as Dynamic Time Warping (DTW) [28], ShapeDTW [35], or Move-Split-Merge (MSM) [32], an alignment path $\mathcal{P} = \{(i_k, j_k)\}_{k=1}^{L_\mathcal{P}}$ of length $L_\mathcal{P}$ is computed between $\mathbf{x}$ and $\mathbf{x}'$. The time steps i_k in $\mathbf{x}$ are then used to generate transformation indices: $\hat{\mathbf{t}} = \{i_k\}_{k=1}^{L_\mathcal{P}}$. These indices determine how to reorder and interpolate the original series. The warped series is obtained by resampling the aligned points to the original length:

$$\hat{\mathbf{x}} = interp(\{x_{\hat{t}_k}\}_{k=1}^{L_\mathcal{P}}, L). \tag{7}$$

2.2.6 Discriminative Guided Warping

Discriminative Guided Warping (DGW) [23] augments a time series by aligning it to a specially selected guiding series that is discriminative with respect to class boundaries. Unlike RGW (Section 2.2.5), where the guiding sample is chosen at random, DGW selects a reference series $\mathbf{x}^*$ that maximizes the contrast between positive and negative class distances.

For each input series $\mathbf{x}$ of length L, we sample a set of positive prototypes $\mathcal{S}_{pos}$ (from the same class) and negative prototypes $\mathcal{S}_{neg}$ (from different classes). For each $\mathbf{s}_{pos} \in \mathcal{S}_{pos}$, we compute its mean distance from the other positives and negatives:

$$d_{pos}(\mathbf{s}_{pos}) = \frac{1}{|\mathcal{S}_{pos}| - 1} \sum_{\substack{\mathbf{p} \in \mathcal{S}_{pos} \\ \mathbf{p} \neq \mathbf{s}_{pos}}} D(\mathbf{s}_{pos}, \mathbf{p}), \quad d_{neg}(\mathbf{s}_{pos}) = \frac{1}{|\mathcal{S}_{neg}|} \sum_{\mathbf{n} \in \mathcal{S}_{neg}} D(\mathbf{s}_{pos}, \mathbf{n}). \tag{8}$$

where $D(\cdot, \cdot)$ is an elastic similarity measure such as DTW [28], MSM [32], or ShapeDTW [35]. The reference series is selected as:

$$\mathbf{x}^* = \arg \max_{\mathbf{s}_{pos} \in \mathcal{S}_{pos}} [d_{neg}(\mathbf{s}_{pos}) - d_{pos}(\mathbf{s}_{pos})]. \tag{9}$$

Once the discriminative guiding series $\mathbf{x}^*$ is selected, the rest of the transformation proceeds identically to RGW (Section 2.2.5). That is, we compute the alignment path $\mathcal{P} = \{(i_k, j_k)\}_{k=1}^{L_\mathcal{P}}$ between $\mathbf{x}$ and $\mathbf{x}^*$ to generate transformation indices: $\hat{\mathbf{t}} = \{i_k\}_{k=1}^{L_\mathcal{P}}$. The warped series is obtained by resampling the aligned points to the original length:

$$\hat{\mathbf{x}} = interp(\{x_{\hat{t}_k}\}_{k=1}^{L_\mathcal{P}}, L). \tag{10}$$

2.2.7 Weighted Barycenter Averaging

Weighted Barycenter Averaging (WBA) [7, 15, 16] augments a time series by synthesizing a prototype from neighbors of the same class using Elastic Barycenter Averaging (EBA) [30, 21, 9] with weighted contributions. For each input series $\mathbf{x}$, a set of neighbors from the same class $\{\mathbf{x}^{(i)}\}_{i=1}^{n}$ is selected, and elastic similarity measures $D(\mathbf{x}, \mathbf{x}^{(i)})$ are used to assign similarity-based weights w_i as follows:

$$w_i = \exp\left(\ln(0.5) \cdot \frac{D(\mathbf{x}^{(i)}, \mathbf{x})}{D^*}\right), \tag{11}$$

where $D(\cdot, \cdot)$ is the elastic similarity measure (DTW, ShapeDTW, or MSM), and $D^* = \min_i D(\mathbf{x}^{(i)}, \mathbf{x})$ is the distance between the input series and its closest neighbor.

The prototype $\mathbf{z}^*$ is computed by minimizing a weighted elastic similarity measure objective:

$$\mathbf{z}^* = \arg \min_{\mathbf{z}} \sum_{i=1}^{n} w_i \cdot D(\mathbf{z}, \mathbf{x}^{(i)}). \tag{12}$$

Unlike arithmetic averaging, this objective aligns each neighbor $\mathbf{x}^{(i)}$ to a common reference series $\mathbf{z}^*$ via time warping. At each time step t in $\mathbf{z}$, a weighted average is computed over all time points in $\mathbf{x}^{(i)}$ that are aligned (via the warping path generated by $D(\mathbf{z}^*, \mathbf{x}^{(i)})$) to t.

Formally, let $\mathcal{A}_t^{(i)}$ denote the set of indices in $\mathbf{x}^{(i)}$ aligned to time step t in $\mathbf{z}^*$. Then the barycenter is updated iteratively by:

$$z_t^* = \frac{\sum_{i=1}^{n} \sum_{t' \in \mathcal{A}_t^{(i)}} w_i \cdot x_{t'}^{(i)}}{\sum_{i=1}^{n} w_i \cdot |\mathcal{A}_t^{(i)}|}. \tag{13}$$

This ensures that the prototype captures the shape variability between aligned neighbors, while being biased towards the input series $\mathbf{x}$ through initialization. After convergence, the final prototype is used as the augmented series $\hat{\mathbf{x}} = \mathbf{z}^*_{last}$.

In the case where $D(\cdot,\cdot)$ is set to DTW [28], the method is called DBA [30]; when set to ShapeDTW [35], it is referred to as ShapeDBA [21]; and when set to MSM [32], it is referred to as MBA [9].

3 Experimental Evaluation

3.1 Evaluation Metrics

Evaluating generative models has been extensively studied across data modalities [29], including temporal data [14]. Recently, Ismail-Fawaz et al. [17] proposed a unified framework that collects and extends evaluation metrics, including one designed specifically for temporal data. Our study follows this framework and adopts eight metrics in total to assess the quality of generated samples.

These metrics are divided into two categories: three measure fidelity, and five assess diversity. For detailed formulations and parameter settings, we refer the reader to [17].

3.1.1 Fidelity Metrics

Fidelity measures how similar the generated samples are to real ones. The three metrics used are Fréchet Inception Distance (FID), Accuracy on Generated (AOG), and Density. FID compares the distributions of real and generated samples in latent space. AOG evaluates whether generated samples preserve the class label of their original counterparts. Density measures how well generated samples populate real-sample neighborhoods in latent space.

3.1.2 Diversity Metrics

Diversity reflects how much generated samples differ from each other. The five diversity metrics are Average Pair Distance (APD), Average per Class Pair Distance (ACPD), Coverage, Mean Maximum Similarity (MMS), and Warping Path Diversity (WPD). APD calculates the average distance between random pairs in latent space. ACPD computes the same measure within class, averaged across all classes. Coverage counts how many real-sample neighborhoods are populated by generated samples. MMS measures how distinct generated samples are from real ones by computing the mean of their maximum similarities. WPD uses Dynamic Time Warping (DTW) [28] to assess structural diversity, by measuring how far each warping path deviates from the diagonal of the DTW matrix.

All metrics, except WPD, are computed in the latent space of a feature extractor pre-trained on the real training data. WPD is computed directly on raw time series.

3.1.3 Metrics Interpretation

For each metric, we compute two versions: one on the generated data and one on the real training data. Following [17], we report the absolute difference between the two. A lower difference indicates that the augmentation method better preserves the characteristics of the original data. Due to space constraints, we limit the analysis to this difference-based comparison, and refer to [17] for deeper discussions on metric interpretation.

3.2 Experimental Setup

We evaluate all methods on 131 univariate time series classification datasets, primarily sourced from the UCR archive [4] (121 datasets), with additional datasets from the recent update by Middlehurst et al. [27] (10 datasets). A dataset was excluded during the selection phase if at least one class contained only a single sample in the training set. Additionally, a small number of datasets (11 in total) were excluded due to running time constraints, either because of a large number of samples, which reduced the need for data augmentation, or because of long series lengths, which significantly increased the computation time of warping-based methods. Each dataset includes a predefined train-test split, which we use without modification.

To reduce bias introduced by random seed selection during augmentation, we generate five different versions for each augmentation technique using only the training portion of each dataset. For each version, synthetic samples are generated to match the size and class distribution of the original training set. The test sets provided by the datasets are not used at any stage of this study, as our evaluation focuses solely on the generative quality of the augmented training data.

We use the LITE architecture [12, 18] as a feature extractor to compute generative model evaluation metrics. For each dataset, a LITE model is trained from scratch using the original training set, repeated across five different random initializations. Each augmented dataset (five per technique) is evaluated using each of the five LITE models, resulting in 25 combinations per augmentation method. Reported results are averaged over these 25 runs.

All experiments were conducted on a machine with an NVIDIA RTX 3090 (24GB VRAM), an AMD Ryzen 9 5950X 16-core processor, and 64GB of RAM, running Ubuntu 24.04. **The source-code of this work is publicly available on GitHub** [4]. We use the open-source package *aeon* [26] to compute warping paths for DTW, MSM, and ShapeDTW, as well as to compute the EBA algorithm used in the WBA augmentation method. To optimize computation, we employ *numba* [24] for Just-In-Time (JIT) compilation of CPU-based augmentation methods, and *tensorflow* [1] for GPU parallelization where applicable.

For three augmentation methods, RGW, DGW, and WBA, a similarity measure must be chosen to guide the transformation. In this study, we evaluate each of these methods using the same set of six configurations: DTW (with a window of 10% and with no window), ShapeDTW (with reach values of 7 and 15), and MSM (with cost values of 1.0 and 2.0). We refer to the resulting

[4] https://github.com/MSD-IRIMAS/Data-Augmentation-4-TSC

RGW variants as: RGW-DTW-0.1, RGW-DTW-None, RGW-SDTW-7, RGW-SDTW-15, RGW-MSM-1, and RGW-MSM-2. DGW variants follow the same naming convention. For WBA, we denote the methods as: WDBA-0.1, WDBA-None, WSDBA-7, WSDBA-15, WMBA-1, and WMBA-2, where DBA [30] uses DTW, SDBA [21] uses ShapeDTW, and MBA [9] uses MSM.

3.3 Experimental Results & Discussion

To facilitate interpretation and visualization of results, we use the Multi-Comparison Matrix (MCM) [11]. The MCM provides a detailed pairwise (1-vs-1) comparison between competing methods, while also ranking them from best to worst based on average performance. Rows and columns are ordered according to the average metric across all datasets, where a lower value indicates better performance. In our case, the evaluation statistic is the absolute difference between metric values computed on generated versus real samples; thus, smaller values indicate closer alignment with the real data distribution. Furthermore, the MCM includes in each cell the win/tie/loss count between the paired methods, the difference in average performance and a p-value to quantify the statistical significance in the difference of performance between the paired methods over all datasets. The used post-hoc test in this study is the two-sided Wilcoxon Signed Rank Test [34].

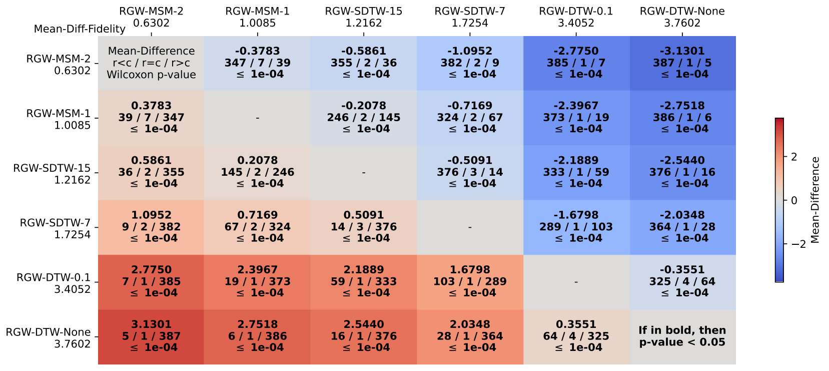
We first apply the MCM separately to the three parameterized methods: RGW, DGW, and WBA, to determine the best-performing configuration among the six variants for each. Once the best configuration is selected for each of these three methods, we add the remaining augmentation methods (Amplitude Warping (AW), Amplitude Scaling (AS), Window Warping (WW), and Jittering) and conduct a full comparison across all techniques.

3.3.1 Comparing Configuration Setups

For the RGW method, Figure 2 shows the MCM results on the fidelity metrics, and Figure 3 shows the results on the diversity metrics. In both cases, the RGW variant using the MSM similarity measure with a cost value of 2.0 (RGW-MSM-2) ranks first. It also statistically outperforms all other configurations. For this reason, we select RGW-MSM-2 as the representative configuration for the RGW method.

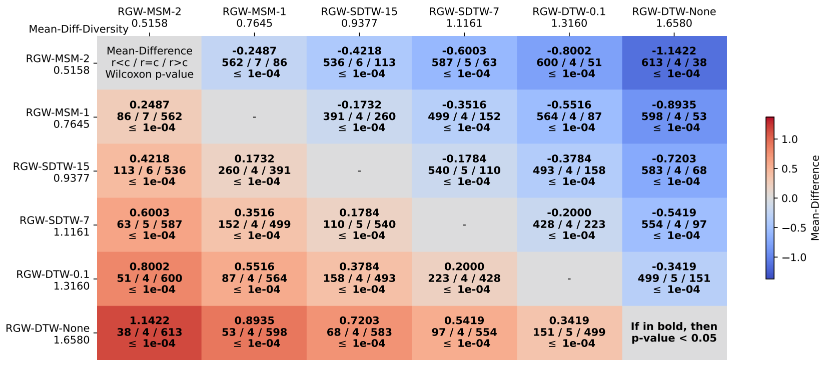
Similarly, for the DGW method, the MCM results for fidelity and diversity are shown in Figures 4 and 5, respectively. As with RGW, the DGW-MSM-2 configuration consistently ranks highest in both categories and is statistically superior to the other variants. We therefore select DGW-MSM-2 as the best-performing configuration for this method.

In contrast, for the WBA method, no single configuration dominates across all metrics. From the fidelity and diversity MCMs (Figures 6 and 7), we observe that three configurations: WDBA-None, WSDBA-7, and WSDBA-15, perform comparably well. This selection is supported by high p-values in the pairwise comparisons, indicating no statistically significant differences between these



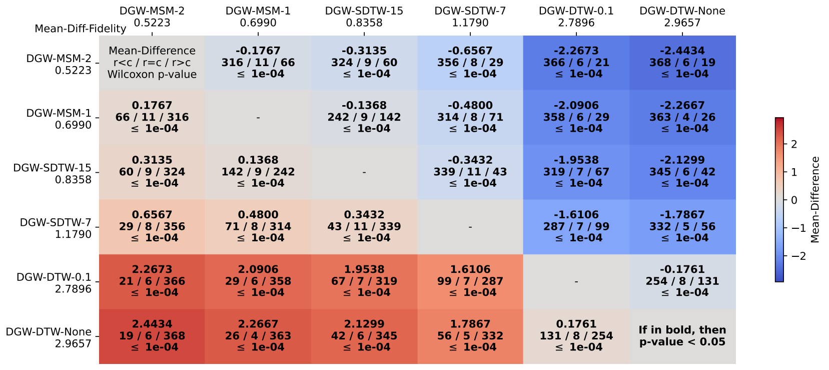
Mean-Diff-Fidelity	RGW-MSM-2 0.6302	RGW-MSM-1 1.0085	RGW-SDTW-15 1.2162	RGW-SDTW-7 1.7254	RGW-DTW-0.1 3.4052	RGW-DTW-None 3.7602
RGW-MSM-2 0.6302	Mean-Difference r<c / r=c / r>c Wilcoxon p-value	-0.3783 347 / 7 / 39 ≤ 1e-04	-0.5861 355 / 2 / 36 ≤ 1e-04	-1.0952 382 / 2 / 9 ≤ 1e-04	-2.7750 385 / 1 / 7 ≤ 1e-04	-3.1301 387 / 1 / 5 ≤ 1e-04
RGW-MSM-1 1.0085	0.3783 39 / 7 / 347 ≤ 1e-04	-	-0.2078 246 / 2 / 145 ≤ 1e-04	-0.7169 324 / 2 / 67 ≤ 1e-04	-2.3967 373 / 1 / 19 ≤ 1e-04	-2.7518 386 / 1 / 6 ≤ 1e-04
RGW-SDTW-15 1.2162	0.5861 36 / 2 / 355 ≤ 1e-04	0.2078 145 / 2 / 246 ≤ 1e-04	-	-0.5091 376 / 3 / 14 ≤ 1e-04	-2.1889 333 / 1 / 59 ≤ 1e-04	-2.5440 376 / 1 / 16 ≤ 1e-04
RGW-SDTW-7 1.7254	1.0952 9 / 2 / 382 ≤ 1e-04	0.7169 67 / 2 / 324 ≤ 1e-04	0.5091 14 / 3 / 376 ≤ 1e-04	-	-1.6798 289 / 1 / 103 ≤ 1e-04	-2.0348 364 / 1 / 28 ≤ 1e-04
RGW-DTW-0.1 3.4052	2.7750 7 / 1 / 385 ≤ 1e-04	2.3967 19 / 1 / 373 ≤ 1e-04	2.1889 59 / 1 / 333 ≤ 1e-04	1.6798 103 / 1 / 289 ≤ 1e-04	-	-0.3551 325 / 4 / 64 ≤ 1e-04
RGW-DTW-None 3.7602	3.1301 5 / 1 / 387 ≤ 1e-04	2.7518 6 / 1 / 386 ≤ 1e-04	2.5440 16 / 1 / 376 ≤ 1e-04	2.0348 28 / 1 / 364 ≤ 1e-04	0.3551 64 / 4 / 325 ≤ 1e-04	If in bold, then p-value < 0.05

Fig. 2. MCM comparing Random Guided Warping (RGW) configurations on fidelity.



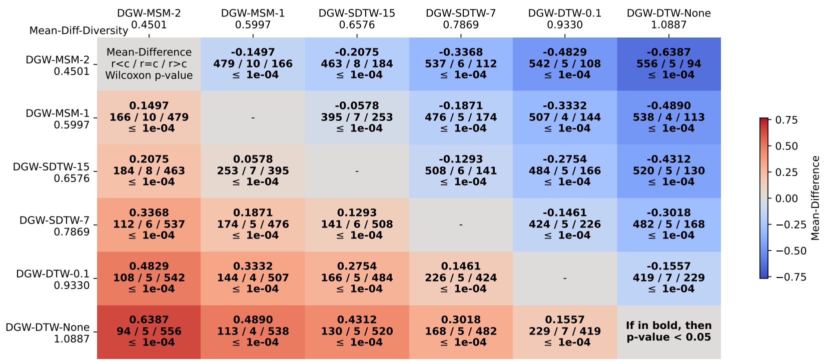
Mean-Diff-Diversity	RGW-MSM-2 0.5158	RGW-MSM-1 0.7645	RGW-SDTW-15 0.9377	RGW-SDTW-7 1.1161	RGW-DTW-0.1 1.3160	RGW-DTW-None 1.6580
RGW-MSM-2 0.5158	Mean-Difference r<c / r=c / r>c Wilcoxon p-value	-0.2487 562 / 7 / 86 ≤ 1e-04	-0.4218 536 / 6 / 113 ≤ 1e-04	-0.6003 587 / 5 / 63 ≤ 1e-04	-0.8002 600 / 4 / 51 ≤ 1e-04	-1.1422 613 / 4 / 38 ≤ 1e-04
RGW-MSM-1 0.7645	0.2487 86 / 7 / 562 ≤ 1e-04	-	-0.1732 391 / 4 / 260 ≤ 1e-04	-0.3516 499 / 4 / 152 ≤ 1e-04	-0.5516 564 / 4 / 87 ≤ 1e-04	-0.8935 598 / 4 / 53 ≤ 1e-04
RGW-SDTW-15 0.9377	0.4218 113 / 6 / 536 ≤ 1e-04	0.1732 260 / 4 / 391 ≤ 1e-04	-	-0.1784 540 / 5 / 110 ≤ 1e-04	-0.3784 493 / 4 / 158 ≤ 1e-04	-0.7203 583 / 4 / 68 ≤ 1e-04
RGW-SDTW-7 1.1161	0.6003 63 / 5 / 587 ≤ 1e-04	0.3516 152 / 4 / 499 ≤ 1e-04	0.1784 110 / 5 / 540 ≤ 1e-04	-	-0.2000 428 / 4 / 223 ≤ 1e-04	-0.5419 554 / 4 / 97 ≤ 1e-04
RGW-DTW-0.1 1.3160	0.8002 51 / 4 / 600 ≤ 1e-04	0.5516 87 / 4 / 564 ≤ 1e-04	0.3784 158 / 4 / 493 ≤ 1e-04	0.2000 223 / 4 / 428 ≤ 1e-04	-	-0.3419 499 / 5 / 151 ≤ 1e-04
RGW-DTW-None 1.6580	1.1422 38 / 4 / 613 ≤ 1e-04	0.8935 53 / 4 / 598 ≤ 1e-04	0.7203 68 / 4 / 583 ≤ 1e-04	0.5419 97 / 4 / 554 ≤ 1e-04	0.3419 151 / 5 / 499 ≤ 1e-04	If in bold, then p-value < 0.05

Fig. 3. MCM comparing Random Guided Warping (RGW) configurations on diversity.



Mean-Diff-Fidelity	DGW-MSM-2 0.5223	DGW-MSM-1 0.6990	DGW-SDTW-15 0.8358	DGW-SDTW-7 1.1790	DGW-DTW-0.1 2.7896	DGW-DTW-None 2.9657
DGW-MSM-2 0.5223	Mean-Difference r<c / r=c / r>c Wilcoxon p-value	-0.1767 316 / 11 / 66 ≤ 1e-04	-0.3135 324 / 9 / 60 ≤ 1e-04	-0.6567 356 / 8 / 29 ≤ 1e-04	-2.2673 366 / 6 / 21 ≤ 1e-04	-2.4434 368 / 6 / 19 ≤ 1e-04
DGW-MSM-1 0.6990	0.1767 66 / 11 / 316 ≤ 1e-04	-	-0.1368 242 / 9 / 142 ≤ 1e-04	-0.4800 314 / 8 / 71 ≤ 1e-04	-2.0906 358 / 6 / 29 ≤ 1e-04	-2.2667 363 / 4 / 26 ≤ 1e-04
DGW-SDTW-15 0.8358	0.3135 60 / 9 / 324 ≤ 1e-04	0.1368 142 / 9 / 242 ≤ 1e-04	-	-0.3432 339 / 11 / 43 ≤ 1e-04	-1.9538 319 / 7 / 67 ≤ 1e-04	-2.1299 345 / 6 / 42 ≤ 1e-04
DGW-SDTW-7 1.1790	0.6567 29 / 8 / 356 ≤ 1e-04	0.4800 71 / 8 / 314 ≤ 1e-04	0.3432 43 / 11 / 339 ≤ 1e-04	-	-1.6106 287 / 7 / 99 ≤ 1e-04	-1.7867 332 / 5 / 56 ≤ 1e-04
DGW-DTW-0.1 2.7896	2.2673 21 / 6 / 366 ≤ 1e-04	2.0906 29 / 6 / 358 ≤ 1e-04	1.9538 67 / 7 / 319 ≤ 1e-04	1.6106 99 / 7 / 287 ≤ 1e-04	-	-0.1761 254 / 8 / 131 ≤ 1e-04
DGW-DTW-None 2.9657	2.4434 19 / 6 / 368 ≤ 1e-04	2.2667 26 / 4 / 363 ≤ 1e-04	2.1299 42 / 6 / 345 ≤ 1e-04	1.7867 56 / 5 / 332 ≤ 1e-04	0.1761 131 / 8 / 254 ≤ 1e-04	If in bold, then p-value < 0.05

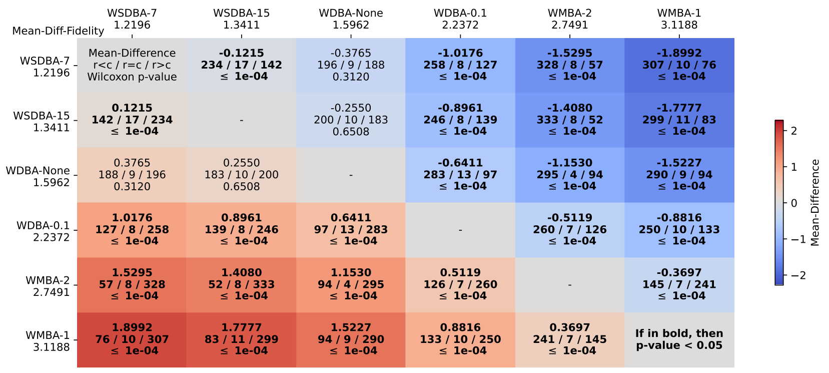
Fig. 4. MCM comparing Discriminative Guided Warping (DGW) configurations on fidelity.



Mean-Diff-Diversity	DGW-MSM-2 0.4501	DGW-MSM-1 0.5997	DGW-SDTW-15 0.6576	DGW-SDTW-7 0.7869	DGW-DTW-0.1 0.9330	DGW-DTW-None 1.0887
DGW-MSM-2 0.4501	Mean-Difference r<c / r=c / r>c Wilcoxon p-value	**-0.1497 479 / 10 / 166 ≤ 1e-04**	**-0.2075 463 / 8 / 184 ≤ 1e-04**	**-0.3368 537 / 6 / 112 ≤ 1e-04**	**-0.4829 542 / 5 / 108 ≤ 1e-04**	**-0.6387 556 / 5 / 94 ≤ 1e-04**
DGW-MSM-1 0.5997	**0.1497 166 / 10 / 479 ≤ 1e-04**	-	**-0.0578 395 / 7 / 253 ≤ 1e-04**	**-0.1871 476 / 5 / 174 ≤ 1e-04**	**-0.3332 507 / 4 / 144 ≤ 1e-04**	**-0.4890 538 / 4 / 113 ≤ 1e-04**
DGW-SDTW-15 0.6576	**0.2075 184 / 8 / 463 ≤ 1e-04**	**0.0578 253 / 7 / 395 ≤ 1e-04**	-	**-0.1293 508 / 6 / 141 ≤ 1e-04**	**-0.2754 484 / 5 / 166 ≤ 1e-04**	**-0.4312 520 / 5 / 130 ≤ 1e-04**
DGW-SDTW-7 0.7869	**0.3368 112 / 6 / 537 ≤ 1e-04**	**0.1871 174 / 5 / 476 ≤ 1e-04**	**0.1293 141 / 6 / 508 ≤ 1e-04**	-	**-0.1461 424 / 5 / 226 ≤ 1e-04**	**-0.3018 482 / 5 / 168 ≤ 1e-04**
DGW-DTW-0.1 0.9330	**0.4829 108 / 5 / 542 ≤ 1e-04**	**0.3332 144 / 4 / 507 ≤ 1e-04**	**0.2754 166 / 5 / 484 ≤ 1e-04**	**0.1461 226 / 5 / 424 ≤ 1e-04**	-	**-0.1557 419 / 7 / 229 ≤ 1e-04**
DGW-DTW-None 1.0887	**0.6387 94 / 5 / 556 ≤ 1e-04**	**0.4890 113 / 4 / 538 ≤ 1e-04**	**0.4312 130 / 5 / 520 ≤ 1e-04**	**0.3018 168 / 5 / 482 ≤ 1e-04**	**0.1557 229 / 7 / 419 ≤ 1e-04**	**If in bold, then p-value < 0.05**

Fig. 5. MCM comparing Discriminative Guided Warping (DGW) configurations on diversity.

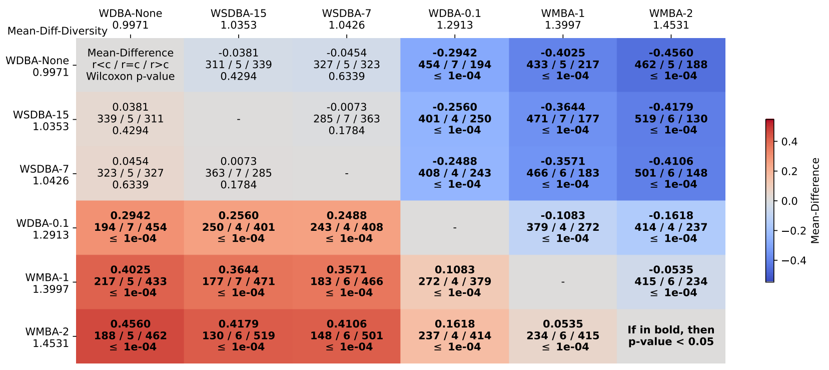
configurations across datasets. As a result, we retain all three as representative configurations for WBA.



Mean-Diff-Fidelity	WSDBA-7 1.2196	WSDBA-15 1.3411	WDBA-None 1.5962	WDBA-0.1 2.2372	WMBA-2 2.7491	WMBA-1 3.1188
WSDBA-7 1.2196	Mean-Difference r<c / r=c / r>c Wilcoxon p-value	**-0.1215 234 / 17 / 142 ≤ 1e-04**	-0.3765 196 / 9 / 188 0.3120	**-1.0176 258 / 8 / 127 ≤ 1e-04**	**-1.5295 328 / 8 / 57 ≤ 1e-04**	**-1.8992 307 / 10 / 76 ≤ 1e-04**
WSDBA-15 1.3411	**0.1215 142 / 17 / 234 ≤ 1e-04**	-	-0.2550 200 / 10 / 183 0.6508	**-0.8961 246 / 8 / 139 ≤ 1e-04**	**-1.4080 333 / 8 / 52 ≤ 1e-04**	**-1.7777 299 / 11 / 83 ≤ 1e-04**
WDBA-None 1.5962	0.3765 188 / 9 / 196 0.3120	0.2550 183 / 10 / 200 0.6508	-	**-0.6411 283 / 13 / 97 ≤ 1e-04**	**-1.1530 295 / 4 / 94 ≤ 1e-04**	**-1.5227 290 / 9 / 94 ≤ 1e-04**
WDBA-0.1 2.2372	**1.0176 127 / 8 / 258 ≤ 1e-04**	**0.8961 139 / 8 / 246 ≤ 1e-04**	**0.6411 97 / 13 / 283 ≤ 1e-04**	-	**-0.5119 260 / 7 / 126 ≤ 1e-04**	**-0.8816 250 / 10 / 133 ≤ 1e-04**
WMBA-2 2.7491	**1.5295 57 / 8 / 328 ≤ 1e-04**	**1.4080 52 / 8 / 333 ≤ 1e-04**	**1.1530 94 / 4 / 295 ≤ 1e-04**	**0.5119 126 / 7 / 260 ≤ 1e-04**	-	**-0.3697 145 / 7 / 241 ≤ 1e-04**
WMBA-1 3.1188	**1.8992 76 / 10 / 307 ≤ 1e-04**	**1.7777 83 / 11 / 299 ≤ 1e-04**	**1.5227 94 / 9 / 290 ≤ 1e-04**	**0.8816 133 / 10 / 250 ≤ 1e-04**	**0.3697 241 / 7 / 145 ≤ 1e-04**	**If in bold, then p-value < 0.05**

Fig. 6. MCM comparing Weighted Barycenter Averaging (WBA) configurations on fidelity.

While MSM yields strong results in RGW and DGW, where warping paths are used to re-index individual time series, its performance in WBA is notably weaker. This difference highlights how the role of the warping path affects method suitability: RGW and DGW apply the MSM path to a single alignment, where its flexibility enables discriminative, class-preserving warps. In contrast, WBA relies on multiple consistent alignments across samples to compute stable averages. MSM's edit-based operations may produce inconsistent local correspondences, making it less suitable for barycenter averaging, where alignment noise is amplified during aggregation.

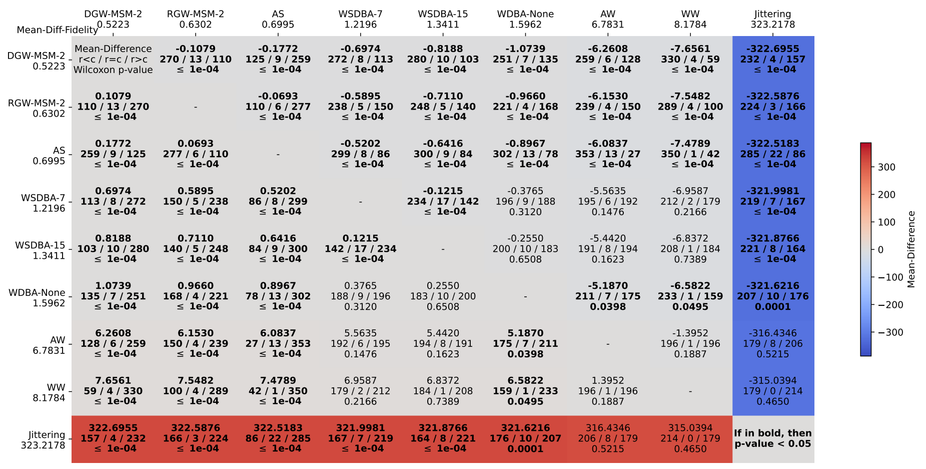


Mean-Diff-Diversity	WDBA-None 0.9971	WSDBA-15 1.0353	WSDBA-7 1.0426	WDBA-0.1 1.2913	WMBA-1 1.3997	WMBA-2 1.4531
WDBA-None 0.9971	Mean-Difference r<c / r=c / r>c Wilcoxon p-value	-0.0381 311 / 5 / 339 0.4294	-0.0454 327 / 5 / 323 0.6339	**-0.2942 454 / 7 / 194 ≤ 1e-04**	**-0.4025 433 / 5 / 217 ≤ 1e-04**	**-0.4560 462 / 5 / 188 ≤ 1e-04**
WSDBA-15 1.0353	0.0381 339 / 5 / 311 0.4294	-	-0.0073 285 / 7 / 363 0.1784	**-0.2560 401 / 4 / 250 ≤ 1e-04**	**-0.3644 471 / 7 / 177 ≤ 1e-04**	**-0.4179 519 / 6 / 130 ≤ 1e-04**
WSDBA-7 1.0426	0.0454 323 / 5 / 327 0.6339	0.0073 363 / 7 / 285 0.1784	-	**-0.2488 408 / 4 / 243 ≤ 1e-04**	**-0.3571 466 / 6 / 183 ≤ 1e-04**	**-0.4106 501 / 6 / 148 ≤ 1e-04**
WDBA-0.1 1.2913	**0.2942 194 / 7 / 454 ≤ 1e-04**	**0.2560 250 / 4 / 401 ≤ 1e-04**	**0.2488 243 / 4 / 408 ≤ 1e-04**	-	**-0.1083 379 / 4 / 272 ≤ 1e-04**	**-0.1618 414 / 4 / 237 ≤ 1e-04**
WMBA-1 1.3997	**0.4025 217 / 5 / 433 ≤ 1e-04**	**0.3644 177 / 7 / 471 ≤ 1e-04**	**0.3571 183 / 6 / 466 ≤ 1e-04**	**0.1083 272 / 4 / 379 ≤ 1e-04**	-	**-0.0535 415 / 6 / 234 ≤ 1e-04**
WMBA-2 1.4531	**0.4560 188 / 5 / 462 ≤ 1e-04**	**0.4179 130 / 6 / 519 ≤ 1e-04**	**0.4106 148 / 6 / 501 ≤ 1e-04**	**0.1618 237 / 4 / 414 ≤ 1e-04**	**0.0535 234 / 6 / 415 ≤ 1e-04**	**If in bold, then p-value < 0.05**

Fig. 7. MCM comparing Weighted Barycenter Averaging (WBA) configurations on diversity.

3.4 Full Comparison

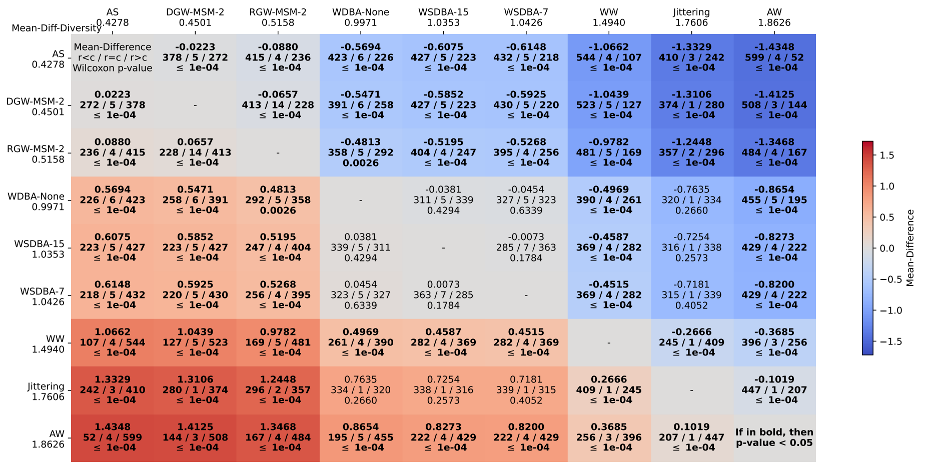
In this section, we compare the best-performing configurations of DGW, RGW, and WBA against the remaining augmentation methods: AS, AW, WW, and Jittering. Figures 8 and 9 present the MCM results for fidelity and diversity, respectively.



Mean-Diff-Fidelity	DGW-MSM-2 0.5223	RGW-MSM-2 0.6302	AS 0.6995	WSDBA-7 1.2196	WSDBA-15 1.3411	WDBA-None 1.5962	AW 6.7831	WW 8.1784	Jittering 323.2178
DGW-MSM-2 0.5223	Mean-Difference r<c / r=c / r>c Wilcoxon p-value	**-0.1079 270 / 13 / 110 ≤ 1e-04**	**-0.1772 125 / 9 / 259 ≤ 1e-04**	**-0.6974 272 / 8 / 113 ≤ 1e-04**	**-0.8188 280 / 10 / 103 ≤ 1e-04**	**-1.0739 251 / 7 / 135 ≤ 1e-04**	**-6.2608 259 / 6 / 128 ≤ 1e-04**	**-7.6561 330 / 4 / 59 ≤ 1e-04**	**-322.6955 232 / 4 / 157 ≤ 1e-04**
RGW-MSM-2 0.6302	**0.1079 110 / 13 / 270 ≤ 1e-04**	-	**-0.0693 110 / 6 / 277 ≤ 1e-04**	**-0.5895 238 / 5 / 150 ≤ 1e-04**	**-0.7110 248 / 5 / 140 ≤ 1e-04**	**-0.9660 221 / 4 / 168 ≤ 1e-04**	**-6.1530 239 / 4 / 150 ≤ 1e-04**	**-7.5482 289 / 4 / 100 ≤ 1e-04**	**-322.5876 224 / 3 / 166 ≤ 1e-04**
AS 0.6995	**0.1772 259 / 9 / 125 ≤ 1e-04**	**0.0693 277 / 6 / 110 ≤ 1e-04**	-	**-0.5202 299 / 8 / 86 ≤ 1e-04**	**-0.6416 300 / 9 / 84 ≤ 1e-04**	**-0.8967 302 / 13 / 78 ≤ 1e-04**	**-6.0837 353 / 13 / 27 ≤ 1e-04**	**-7.4789 350 / 1 / 42 ≤ 1e-04**	**-322.5183 285 / 22 / 86 ≤ 1e-04**
WSDBA-7 1.2196	**0.6974 113 / 8 / 272 ≤ 1e-04**	**0.5895 150 / 5 / 238 ≤ 1e-04**	**0.5202 86 / 8 / 299 ≤ 1e-04**	-	**-0.1215 234 / 17 / 142 ≤ 1e-04**	-0.3765 196 / 9 / 188 0.3120	-5.5635 195 / 6 / 192 0.1476	-6.9587 212 / 2 / 179 0.2166	**-321.9981 219 / 7 / 167 ≤ 1e-04**
WSDBA-15 1.3411	**0.8188 103 / 10 / 280 ≤ 1e-04**	**0.7110 140 / 5 / 248 ≤ 1e-04**	**0.6416 84 / 9 / 300 ≤ 1e-04**	**0.1215 142 / 17 / 234 ≤ 1e-04**	-	-0.2550 200 / 10 / 183 0.6508	-5.4420 191 / 8 / 194 0.1623	-6.8372 208 / 1 / 184 0.7389	**-321.8766 221 / 8 / 164 ≤ 1e-04**
WDBA-None 1.5962	**1.0739 135 / 7 / 251 ≤ 1e-04**	**0.9660 168 / 4 / 221 ≤ 1e-04**	**0.8967 78 / 13 / 302 ≤ 1e-04**	0.3765 188 / 9 / 196 0.3120	0.2550 183 / 10 / 200 0.6508	-	**-5.1870 211 / 7 / 175 0.0398**	**-6.5822 233 / 1 / 159 0.0495**	**-321.6216 207 / 10 / 176 0.0001**
AW 6.7831	**6.2608 128 / 6 / 259 ≤ 1e-04**	**6.1530 150 / 4 / 239 ≤ 1e-04**	**6.0837 27 / 13 / 353 ≤ 1e-04**	5.5635 192 / 6 / 195 0.1476	5.4420 194 / 8 / 191 0.1623	**5.1870 175 / 7 / 211 0.0398**	-	-1.3952 196 / 1 / 196 0.1887	-316.4346 179 / 8 / 206 0.5215
WW 8.1784	**7.6561 59 / 4 / 330 ≤ 1e-04**	**7.5482 100 / 4 / 289 ≤ 1e-04**	**7.4789 42 / 1 / 350 ≤ 1e-04**	6.9587 179 / 2 / 212 0.2166	6.8372 184 / 1 / 208 0.7389	**6.5822 159 / 1 / 233 0.0495**	1.3952 196 / 1 / 196 0.1887	-	-315.0394 179 / 0 / 214 0.4650
Jittering 323.2178	**322.6955 157 / 4 / 232 ≤ 1e-04**	**322.5876 166 / 3 / 224 ≤ 1e-04**	**322.5183 86 / 22 / 285 ≤ 1e-04**	**321.9981 167 / 7 / 219 ≤ 1e-04**	**321.8766 164 / 8 / 221 ≤ 1e-04**	**321.6216 176 / 10 / 207 0.0001**	316.4346 206 / 8 / 179 0.5215	315.0394 214 / 0 / 179 0.4650	**If in bold, then p-value < 0.05**

Fig. 8. MCM presenting the comparison between all methods with their best configuration in terms of fidelity.

For fidelity, DGW-MSM-2 ranks first, indicating its ability to generate samples that closely resemble the real distribution. RGW-MSM-2 follows closely, reinforcing MSM's effectiveness for pairwise alignment. AS ranks third with a low



Mean-Diff-Diversity	AS 0.4278	DGW-MSM-2 0.4501	RGW-MSM-2 0.5158	WDBA-None 0.9971	WSDBA-15 1.0353	WSDBA-7 1.0426	WW 1.4940	Jittering 1.7606	AW 1.8626
AS 0.4278	Mean-Difference r<c / r=c / r>c Wilcoxon p-value	**-0.0223 378 / 5 / 272 ≤ 1e-04**	**-0.0880 415 / 4 / 236 ≤ 1e-04**	**-0.5694 423 / 6 / 226 ≤ 1e-04**	**-0.6075 427 / 5 / 223 ≤ 1e-04**	**-0.6148 432 / 5 / 218 ≤ 1e-04**	**-1.0662 544 / 4 / 107 ≤ 1e-04**	**-1.3329 410 / 3 / 242 ≤ 1e-04**	**-1.4348 599 / 4 / 52 ≤ 1e-04**
DGW-MSM-2 0.4501	**0.0223 272 / 5 / 378 ≤ 1e-04**	-	**-0.0657 413 / 14 / 228 ≤ 1e-04**	**-0.5471 391 / 6 / 258 ≤ 1e-04**	**-0.5852 427 / 5 / 223 ≤ 1e-04**	**-0.5925 430 / 5 / 220 ≤ 1e-04**	**-1.0439 523 / 5 / 127 ≤ 1e-04**	**-1.3106 374 / 1 / 280 ≤ 1e-04**	**-1.4125 508 / 3 / 144 ≤ 1e-04**
RGW-MSM-2 0.5158	**0.0880 236 / 4 / 415 ≤ 1e-04**	**0.0657 228 / 14 / 413 ≤ 1e-04**	-	**-0.4813 358 / 5 / 292 0.0026**	**-0.5195 404 / 4 / 247 ≤ 1e-04**	**-0.5268 395 / 4 / 256 ≤ 1e-04**	**-0.9782 481 / 5 / 169 ≤ 1e-04**	**-1.2448 357 / 2 / 296 ≤ 1e-04**	**-1.3468 484 / 4 / 167 ≤ 1e-04**
WDBA-None 0.9971	**0.5694 226 / 6 / 423 ≤ 1e-04**	**0.5471 258 / 6 / 391 ≤ 1e-04**	**0.4813 292 / 5 / 358 0.0026**	-	-0.0381 311 / 5 / 339 0.4294	-0.0454 327 / 5 / 323 0.6339	**-0.4969 390 / 4 / 261 ≤ 1e-04**	-0.7635 320 / 1 / 334 0.2660	**-0.8654 455 / 5 / 195 ≤ 1e-04**
WSDBA-15 1.0353	**0.6075 223 / 5 / 427 ≤ 1e-04**	**0.5852 223 / 5 / 427 ≤ 1e-04**	**0.5195 247 / 4 / 404 ≤ 1e-04**	0.0381 339 / 5 / 311 0.4294	-	-0.0073 285 / 7 / 363 0.1784	**-0.4587 369 / 4 / 282 ≤ 1e-04**	-0.7254 316 / 1 / 338 0.2573	**-0.8273 429 / 4 / 222 ≤ 1e-04**
WSDBA-7 1.0426	**0.6148 218 / 5 / 432 ≤ 1e-04**	**0.5925 220 / 5 / 430 ≤ 1e-04**	**0.5268 256 / 4 / 395 ≤ 1e-04**	0.0454 323 / 5 / 327 0.6339	0.0073 363 / 7 / 285 0.1784	-	**-0.4515 369 / 4 / 282 ≤ 1e-04**	-0.7181 315 / 1 / 339 0.4052	**-0.8200 429 / 4 / 222 ≤ 1e-04**
WW 1.4940	**1.0662 107 / 4 / 544 ≤ 1e-04**	**1.0439 127 / 5 / 523 ≤ 1e-04**	**0.9782 169 / 5 / 481 ≤ 1e-04**	**0.4969 261 / 4 / 390 ≤ 1e-04**	**0.4587 282 / 4 / 369 ≤ 1e-04**	**0.4515 282 / 4 / 369 ≤ 1e-04**	-	**-0.2666 245 / 1 / 409 ≤ 1e-04**	**-0.3685 396 / 3 / 256 ≤ 1e-04**
Jittering 1.7606	**1.3329 242 / 3 / 410 ≤ 1e-04**	**1.3106 280 / 1 / 374 ≤ 1e-04**	**1.2448 296 / 2 / 357 ≤ 1e-04**	0.7635 334 / 1 / 320 0.2660	0.7254 338 / 1 / 316 0.2573	0.7181 339 / 1 / 315 0.4052	**0.2666 409 / 1 / 245 ≤ 1e-04**	-	**-0.1019 447 / 1 / 207 ≤ 1e-04**
AW 1.8626	**1.4348 52 / 4 / 599 ≤ 1e-04**	**1.4125 144 / 3 / 508 ≤ 1e-04**	**1.3468 167 / 4 / 484 ≤ 1e-04**	**0.8654 195 / 5 / 455 ≤ 1e-04**	**0.8273 222 / 4 / 429 ≤ 1e-04**	**0.8200 222 / 4 / 429 ≤ 1e-04**	**0.3685 256 / 3 / 396 ≤ 1e-04**	**0.1019 207 / 1 / 447 ≤ 1e-04**	**If in bold, then p-value < 0.05**

Fig. 9. MCM presenting the comparison between all methods with their best configuration in terms of diversity.

average difference of 0.6995, which is expected given its minimal transformation. WBA methods perform worse than DGW and RGW, with average differences above 1.2, possibly due to noise propagation during the averaging process. However, ShapeDBA outperforms WDBA, suggesting that ShapeDTW may be more robust to outliers. AW and WW exhibit higher distortion in both amplitude and time, resulting in an average performance above 6.7, and Jittering performs the worst, exceeding 300, due to its disruptive nature.

For diversity, AS surprisingly ranks first with an average of 0.4278, indicating it preserves variability close to that of the original data. DGW-MSM-2 and RGW-MSM-2 remain strong, ranking second and third, respectively, showing they balance both fidelity and diversity well. Among WBA variants, DBA slightly outperforms ShapeDBA, though the margin is small (< 0.05) and not statistically significant. Interestingly, WW and Jittering improve in diversity ranking compared to fidelity. This may be due to the convolutional nature of the LITE feature extractor, which can filter out local noise or warping effects, preserving distances in latent space, helpful for diversity, but not fidelity. AW performs worst in diversity (avg. $\simeq 1.9$), likely because amplitude changes significantly alter distances in latent space, which many diversity metrics rely on.

4 Conclusion

In this paper, we presented an extensive analysis of 22 data augmentation techniques for time series classification, each evaluated across multiple configuration setups. We framed augmentation methods as generative models and assessed them accordingly using eight evaluation metrics that quantify fidelity and diversity, applied to 131 publicly available datasets.

To the best of our knowledge, this is the first work to systematically assess time series augmentation from a generative model perspective. This evaluation framework enables a more systematic and objective selection of augmentation methods based on quantitative analysis rather than intuition or visual inspection, prior to their use in performance-driven pipelines.

Our results suggest that DGW, when combined with the MSM similarity measure, is the most effective method in terms of both fidelity and diversity. As future work, we plan to conduct extensive experiments measuring the downstream performance impact of each method in both offline and online augmentation settings. This would allow us to investigate potential causal links between generative evaluation metrics and actual performance gains, further strengthening the selection criteria.

5 Acknowledgments

This work was supported by the ANR DELEGATION project (grant ANR-21-CE23-0014) of the French Agence Nationale de la Recherche. The authors would like to acknowledge the High Performance Computing Center of the University of Strasbourg for supporting this work by providing scientific sup- port and access to computing resources. Part of the computing resources were funded by the Equipex Equip@Meso project (Programme Investissements d'Avenir) and the CPER Alsacalcul/Big Data. The authors would also like to thank the creators and providers of the UCR Archive.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/, software available from tensorflow.org
2. Badi, O., Devanne, M., Ismail-Fawaz, A., Abdullayev, J., Lemaire, V., Berretti, S., Weber, J., Forestier, G.: Cocalite: A hybrid model combining catch22 and lite for time series classification. In: 2024 IEEE International Conference on Big Data (BigData). pp. 1229–1236. IEEE (2024)
3. Bagnall, A., Middlehurst, M., Forestier, G., Ismail-Fawaz, A., Guillaume, A., Guijo-Rubio, D., Tan, C.W., Dempster, A., Webb, G.I.: A hands-on introduction to time series classification and regression. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 6410–6411 (2024)
4. Dau, H.A., Bagnall, A., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Keogh, E.: The ucr time series archive. IEEE/CAA Journal of Automatica Sinica **6**(6), 1293–1305 (2019)

5. Fawaz, H.I., Del Grosso, G., Kerdoncuff, T., Boisbunon, A., Saffar, I.: Deep unsupervised domain adaptation for time series classification: A benchmark. arXiv preprint arXiv:2312.09857 (2023)
6. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Data augmentation using synthetic data for time series classification with deep residual networks. In: International Workshop on Advanced Analytics and Learning on Temporal Data (2018)
7. Forestier, G., Petitjean, F., Dau, H.A., Webb, G.I., Keogh, E.: Generating synthetic time series to augment sparse datasets. In: 2017 IEEE international conference on data mining (ICDM). pp. 865–870. IEEE (2017)
8. Gao, Z., Liu, H., Li, L.: Data augmentation for time-series classification: An extensive empirical study and comprehensive survey. arXiv preprint arXiv:2310.10060 (2023)
9. Holder, C., Guijo-Rubio, D., Bagnall, A.J.: Barycentre averaging for the move-split-merge time series distance measure. In: KDIR. pp. 51–62 (2023)
10. Holder, C., Middlehurst, M., Bagnall, A.: A review and evaluation of elastic distance functions for time series clustering. Knowledge and Information Systems **66**(2), 765–809 (2024)
11. Ismail-Fawaz, A., Dempster, A., Tan, C.W., Herrmann, M., Miller, L., Schmidt, D.F., Berretti, S., Weber, J., Devanne, M., Forestier, G., et al.: An approach to multiple comparison benchmark evaluations that is stable under manipulation of the comparate set. arXiv preprint arXiv:2305.11921 (2023)
12. Ismail-Fawaz, A., Devanne, M., Berretti, S., Weber, J., Forestier, G.: Lite: Light inception with boosting techniques for time series classification. In: 2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA). pp. 1–10. IEEE (2023)
13. Ismail-Fawaz, A., Devanne, M., Berretti, S., Weber, J., Forestier, G.: Finding foundation models for time series classification with a pretext task. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 123–135. Springer (2024)
14. Ismail-Fawaz, A., Devanne, M., Berretti, S., Weber, J., Forestier, G.: A supervised variational auto-encoder for human motion generation using convolutional neural networks. In: International Conference on Pattern Recognition and Artificial Intelligence. pp. 166–181. Springer (2024)
15. Ismail-Fawaz, A., Devanne, M., Berretti, S., Weber, J., Forestier, G.: Weighted average of human motion sequences for improving rehabilitation assessment. In: International Workshop on Advanced Analytics and Learning on Temporal Data. pp. 131–146. Springer (2024)
16. Ismail-Fawaz, A., Devanne, M., Berretti, S., Weber, J., Forestier, G.: Weighted elastic barycetner averaging to augment time series data. https://github.com/MSD-IRIMAS/Augmenting-TSC-Elastic-Averaging (2024)
17. Ismail-Fawaz, A., Devanne, M., Berretti, S., Weber, J., Forestier, G.: Establishing a unified evaluation framework for human motion generation: A comparative analysis of metrics. Computer Vision and Image Understanding **254**, 104337 (2025)
18. Ismail-Fawaz, A., Devanne, M., Berretti, S., Weber, J., Forestier, G.: Look into the lite in deep learning for time series classification. International Journal of Data Science and Analytics pp. 1–21 (2025)
19. Ismail-Fawaz, A., Devanne, M., Weber, J., Forestier, G.: Deep learning for time series classification using new hand-crafted convolution filters. In: 2022 IEEE International Conference on Big Data (Big Data). pp. 972–981. IEEE (2022)

20. Ismail-Fawaz, A., Devanne, M., Weber, J., Forestier, G.: Enhancing time series classification with self-supervised learning. In: International Conference on Agents and Artificial Intelligence (ICAART). pp. 40–47. SCITEPRESS-Science and Technology Publications (2023)
21. Ismail-Fawaz, A., Ismail Fawaz, H., Petitjean, F., Devanne, M., Weber, J., Berretti, S., Webb, G.I., Forestier, G.: Shapedba: Generating effective time series prototypes using shapedtw barycenter averaging. In: International Workshop on Advanced Analytics and Learning on Temporal Data. pp. 127–142. Springer (2023)
22. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data mining and knowledge discovery **33**(4), 917–963 (2019)
23. Iwana, B.K., Uchida, S.: Time series data augmentation for neural networks by time warping with a discriminative teacher. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 3558–3565. IEEE (2021)
24. Lam, S.K., Pitrou, A., Seibert, S.: Numba: A llvm-based python jit compiler. In: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC. pp. 1–6 (2015)
25. Le Guennec, A., Malinowski, S., Tavenard, R.: Data augmentation for time series classification using convolutional neural networks. In: ECML/PKDD workshop on advanced analytics and learning on temporal data (2016)
26. Middlehurst, M., Ismail-Fawaz, A., Guillaume, A., Holder, C., Guijo-Rubio, D., Bulatova, G., Tsaprounis, L., Mentel, L., Walter, M., Schäfer, P., et al.: aeon: a python toolkit for learning from time series. Journal of Machine Learning Research **25**(289), 1–10 (2024)
27. Middlehurst, M., Schäfer, P., Bagnall, A.: Bake off redux: a review and experimental evaluation of recent time series classification algorithms. Data Mining and Knowledge Discovery **38**(4), 1958–2031 (2024)
28. Müller, M.: Dynamic time warping. Information retrieval for music and motion pp. 69–84 (2007)
29. Naeem, M.F., Oh, S.J., Uh, Y., Choi, Y., Yoo, J.: Reliable fidelity and diversity metrics for generative models. In: International conference on machine learning. pp. 7176–7185. PMLR (2020)
30. Petitjean, F., Ketterlin, A., Gançarski, P.: A global averaging method for dynamic time warping, with applications to clustering. Pattern recognition **44**(3), 678–693 (2011)
31. Pialla, G., Devanne, M., Weber, J., Idoumghar, L., Forestier, G.: Data augmentation for time series classification with deep learning models. In: International Workshop on Advanced Analytics and Learning on Temporal Data. pp. 117–132. Springer (2022)
32. Stefan, A., Athitsos, V., Das, G.: The move-split-merge metric for time series. IEEE transactions on Knowledge and Data Engineering **25**(6), 1425–1438 (2012)
33. Um, T.T., Pfister, F.M., Pichler, D., Endo, S., Lang, M., Hirche, S., Fietzek, U., Kulić, D.: Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In: Proceedings of the 19th ACM international conference on multimodal interaction. pp. 216–220 (2017)
34. Wilcoxon, F.: Individual comparisons by ranking methods. In: Breakthroughs in statistics: Methodology and distribution, pp. 196–202. Springer (1992)
35. Zhao, J., Itti, L.: shapedtw: Shape dynamic time warping. Pattern Recognition **74**, 171–184 (2018)