StiefelGen: Time Series Data Augmentation Over the Stiefel Manifold

Prasad Cheema[1], **Mahito Sugiyama**[1,2]

[1]National Institute of Informatics, Japan
[2]Sokendai (The Graduate University for Advanced Studies), Japan
prasad@nii.ac.jp, mahito@nii.ac.jp

Abstract

In time series data augmentation, existing methodologies face significant challenges—lack of access to robust physical models, uncertainties in noise addition, and scarcity of representative datasets. Addressing these limitations head-on, this paper introduces a novel solution rooted in the matrix differential geometry of the Stiefel manifold. Our proposed approach places time series signals on the Stiefel manifold and smoothly perturbs them, providing a holistic solution to the challenges outlined. Through several illustrative use cases, we demonstrate the efficacy of our methodology, *StiefelGen*, showcasing its ability to harness the unique properties of the Stiefel manifold in order to enhance and transform time series data augmentation techniques.

Introduction

Deep learning, crucial in natural language processing (NLP), computer vision (CV), and speech recognition, heavily relies on access to extensive datasets. Synthetic data generation enhances available data by creating artificial samples resembling unexplored input space areas. In CV, common techniques include stretching, flipping, cropping, and hue adjustment in image datasets (Shorten and Khoshgoftaar 2019). Recent innovations involve "mix-up" strategies, which features randomized interpolation between input data or class labels (Zhang et al. 2017; Verma et al. 2018). However, time series data augmentation research is limited due to challenges in temporal dependency structures. Existing methods may yield unrealistic results and fail to capture system physics in time series tasks. Although improvements have been observed, these approaches lack the maturity seen in CV data augmentation methods (Iglesias et al. 2023; Iwana and Uchida 2021a). In the following sections, we provide a concise overview of classical and modern time series data generation approaches.

Classical Approaches

Classical time series data augmentation tends to involve direct signal modifications, including jittering, magnitude alterations, signal warping (e.g., Dynamic Time Warping - DTW), and signal flipping (Iglesias et al. 2023; Iwana and Uchida 2021a). These methods have drawbacks such as bias in jittering, unexpected semantic changes in magnitude alterations, and challenging hyperparameter optimization for signal warping (Balakrishnan 1962; Zhang 2007; Iglesias et al. 2023; Iwana and Uchida 2021b; Fujiwara et al. 2021; Small 2005). Combining these methods enhances accuracy but often requires careful hyperparameter engineering. Methods like GeneRAting TIme Series (GRATIS) (Kang, Hyndman, and Li 2020), dynamic linear modeling (Frühwirth-Schnatter 1994), and sampling over conditional and marginal distributions (Meng and Van Dyk 1999) provide alternatives for generating non-stationary time series instances. Nevertheless, they often require extensive hands on modeling, and or expert domain knowledge to effectively deploy.

Modern Approaches

Modern approaches to time series data generation primarily utilize deep learning (DL), tapping into vast time series databases and the nonlinear learning capabilities of deep generative model architectures. Encoder-decoder models, such as LSTM-AE (Tu et al. 2018) and LSTM VAE (DeVries and Taylor 2017), project high-dimensional input to a lower-dimensional latent space for data generation. Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) are widely used, with variations like fully-connected GANs (Lou, Qi, and Li 2018), GANs with LSTM-based RNNs (Haradal, Hayashi, and Uchida 2018), and GANs utilizing CNNs with sliding windows (Ramponi et al. 2018; Che et al. 2017). However, DL-based methods face challenges of data scarcity, high generation costs, and prolonged training times, especially in domains like aerospace engineering (Osburg, Ohlandt, and Light 2011). Some methods directly perturb signals in the frequency domain, like augmentation bank (Zhang et al. 2022) and Spectral and Time Augmentation (STAug) (Zhang et al. 2023) using empirical mode decomposition. Mixup strategies, including binary (Beckham et al. 2019), cut (Yun et al. 2019), weighted geometric (Verma et al. 2021), amplitude (Xu et al. 2021), and spectrogram mixup (Kim, Han, and Ko 2021), offer diverse mixing formulations. However, these methods face challenges in guaranteeing physically realizable signals, particularly in dynamical data generation physics (West, Prado, and Krystal 1999). DL approaches often demand large data amounts,

posing difficulties in fields like engineering and physics. Thus, whilst modern approaches excel in expressibility they often lack nuances afforded by the classical approach for effective signal generation.

Contributions: This paper introduces the novel time series data augmentation approach, *StiefelGen*. Key contributions and features include leveraging matrix differential geometry for (i) tailored aleatoric and epistemic uncertainty, (ii) minimal hyperparameter requirements, (iii) simplicity in implementation, (iv) interpretability, and its (v) model-agnostic nature. The proposed methodology will be showcased on empirical signals, highlighting each one of these properties.

Methodolody

StieflGen, the algorithm proposed in this paper is rooted in matrix differential geometry (Absil, Mahony, and Sepulchre 2008). For a concise understanding, refer to this section.

Mathematical Preliminaries

Consider a smooth manifold $\mathcal{M}$. If we take $\mathcal{M} \subseteq \mathbb{R}^{m\times n}$, then $\mathcal{M}$ is what is known as a *matrix manifold*. In particular, the *Stiefel manifold*, $\mathrm{St}_n^m \subseteq \mathbb{R}^{m\times n}$ is defined as the set of matrix elements in $\mathbb{R}^{m\times n}$ which satisfy common matrix orthogonality constraints. More specifically,

$$\mathrm{St}_n^m \coloneqq \{U \in \mathbb{R}^{m\times n} \mid U^\intercal U = I_n, m \geq n\}. \tag{1}$$

Notice that if $n = 1$, then one is working over the geometry of a hyper sphere, and thus St_n^m can be thought of as the matrix generalization of the hyper sphere. Alternately, St_n^m can be thought of as the geometry of the set of n orthonormal m-frames. Moreover, if one works with St_n^m considering $m = n$, then one is said to be working with the geometry of the *special orthogonal group*, defined formally as follows:

$$\mathcal{O}(m) \coloneqq \{U \in \mathbb{R}^{m\times m} \mid U^\intercal U = I_m = UU^\intercal\}. \tag{2}$$

In other words, $\mathrm{St}_m^m = \mathcal{O}(m)$ and similarly $\mathrm{St}_n^n = \mathcal{O}(n)$.

Given the manifold St_n^m, consider now a point, $U \in \mathrm{St}_n^m \subseteq \mathbb{R}^{m\times n}$, and define the tangent space local at U to be $\mathcal{T}_U\mathrm{St}_n^m$. A common parameterization of this tangent (vector) space at a point U is given by

$$\mathcal{T}_U\mathrm{St}_n^m = \{\Delta \in \mathbb{R}^{m\times n} \mid \Delta^\intercal U + U^\intercal \Delta = 0\}, \tag{3}$$

thus implying that $U^\intercal\Delta \in \mathbb{R}^{n\times n}$ is skew-symmetric. Further, consider a smooth curve $\gamma : [0, 1] \to \mathrm{St}_n^m$ such that $\gamma(0) = U$. Then, γ is considered to be geodesic over St_n^m if $\nabla_{\dot\gamma}\dot\gamma = 0$ (a condition known as auto-parallelism), where ∇ is some affine connection over St_n^m, as it can be shown that under this condition γ is indeed locally length minimizing. Let $\Delta \in \mathcal{T}_U\mathrm{St}_n^m$ be a tangent vector emanating from U on St_n^m, then there exists a geodesic $\gamma_\Delta : [0, 1] \to \mathrm{St}_n^m$ such that $\gamma_\Delta(0) = U$, and $\dot\gamma_\Delta(0) = \Delta$. The *exponential map* corresponding to this condition is defined as the function, $\mathrm{Exp}_U(\Delta) \coloneqq \gamma_\Delta(1)$. The infimum radius around U with which the calculation of $\mathrm{Exp}_U(\Delta)$ exists in a diffeomorphism with the manifold St_n^m is known as the *radius of injectivity*. Intuitively, it defines the largest possible distance one can travel along a geodesic starting at U whilst remaining in a well-behaved, one-to-one correspondence between points on the manifold St_n^m and the tangent space generated relative to $\mathcal{T}_U\mathrm{St}_n^m$. Finally, given $\mathcal{T}_U\mathrm{St}_n^m$ one can define a *canonical* inner product $\langle\cdot,\cdot\rangle_w : \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}$, with associated weight $w = \left(I - \frac{1}{2}UU^T\right) \in \mathbb{R}^{m\times m}$. The exact form of the canonical metric is given as follows:

$$\langle\Delta, \tilde\Delta\rangle_U = \mathrm{tr}\left(\Delta^T\left(I - \frac{1}{2}UU^T\right)\tilde\Delta\right), \tag{4}$$

where evidently the weighting is performed with respect to U. Given this structure one now has a way in which tangent vectors, can be worked with in a normalized fashion as follows:

$$\bar\Delta = \frac{\Delta}{\sqrt{\langle\Delta,\Delta\rangle_w}} = \frac{\Delta}{\|\Delta\|_w}. \tag{5}$$

The reason for considering this in the paper is so that one can scale randomly sampled Δ vectors with respect to the *radius of injectivity*, so that for example, if one desires a Δ within 0.4 times relative to the radius of injectivity, one can first normalize the sampled Δ then multiply this by 0.4. Interestingly (and rather luckily) for St_n^m the radius of injectivity is known globally to be 0.89π (to within a tight lower bound) (Rentmeesters et al. 2013).

The *StiefelGen* Algorithm Consider a uni-variate time series, $\mathcal{T} = (t_i)_{i=1}^N$. Our proposal, the S(tiefel)Gen(eration) algorithm, requires constructing what is known as the *page matrix* (Damen, Van den Hof, and Hajdasinski 1982) relative to $\mathcal{T}$. This involves choosing a particular window size, m, and reshaping it as:

$$\mathcal{T}_{\text{mat}} = \begin{bmatrix} t_1 & \cdots & t_m \\ t_{m+1} & & t_{2m} \\ \vdots & \ddots & \vdots \\ t_{m(n-1)+1} & \cdots & t_N \end{bmatrix}, \tag{6}$$

where we must naturally require that $m \mid N$ (m divides N). In the event that $m \nmid N$, the resolution entails either (i) implementing a rounding operation, (ii) applying padding, or (iii) introducing a slight signal overlap. Page matrices, introduced as a method to approximate Hankel matrices (Damen, Van den Hof, and Hajdasinski 1982), have historical significance in singular spectrum analysis (SSA), serving as spectrum-based principal components analysis (PCA) tailored for time series (Schoellhamer 2001). Recently, page matrices have been integral in Koopman-based data-driven simulation and control (Lian, Wang, and Jones 2021) and have played a pivotal role in model-agnostic analyses of time series signals (Agarwal et al. 2018). Our trajectory, distinct from forecasting and imputation, focuses on the innovative use of page matrices for synthetic data generation through matrix geometry.

After constructing the page matrix, *StiefelGen* conducts a singular value decomposition (SVD) on $\mathcal{T}\mathrm{mat}, 1$ as $\mathcal{T}\mathrm{mat}, 1 = U_1\Sigma V_1^\intercal$. Notably, U_1 and V_1 are unitary, and thus belong to St_n^m (Absil, Mahony, and Sepulchre 2008). *StiefelGen* then perturbs U_1 and V_1 such that they remain

on St_n^m, avoiding naive linear perturbations that would leave St_n^m. This ensures U_2 and V_2 are valid unitary rotation matrices. Thus, the perturbed page matrix will take form, $\mathcal{T}\mathrm{mat}, 2 = U_2 \Sigma V_2^\intercal$, and by reshaping, a newly generated time series $\mathcal{T}_2$ is obtained, leaving the singular values untouched therefore the original signal physics largely intact. This process smoothly generates new components of rotation matrices without altering the energy contribution toward each singular vector. An intuition of the impact of this can also be viewed from the perspective of the dyadic expansion of the SVD,

$$\mathcal{T}_{\text{mat}} = \Sigma_{i=1}^{n} \sigma_i u_i v_i^\intercal, \tag{7}$$

where $\sigma_i = \Sigma_{ii}$[1], and u_i and v_i are the i-th columns of matrices U and V. According to this expansion by changing only the u_i and v_i vectors, and leaving the σ_i as invariant, one is in smoothly perturbing the basis representations of the signal, but leaving the magnitude of the expansion coefficients unchanged.

In addressing the challenge of efficiently perturbing geodesics over the Stiefel manifold (St_n^m), *StiefelGen* adopts the *exponential map* formulation (Edelman, Arias, and Smith 1998). This approach enables a direct projection of a random tangent vector $\Delta \in \mathcal{T}_U \mathrm{St}_n^m$ onto St_n^m, bypassing the need to numerically integrate the geodesic equations. The exponential map calculation, while efficient, necessitates staying within the *radius of injectivity* to maintain validity. Beyond this radius, challenges arise, including non-injectivity, loss of invertibility and smoothness, and potential encounters with conjugate points (Spivak 1999; Do Carmo 2016). However, for practical applications in time series data augmentation, one usually stays comfortably within the globally known radius of injectivity (globally 0.89π (Rentmeesters et al. 2013)) is feasible and simple. Considerin this, a concise four-step process for generating new U and V matrices in *StiefelGen* is outlined as follows: (1) Randomly sample a matrix with respect to an origin point, which in this case is U_1 (or V_1). This sampled point is not initially constrained to St_n^m. (2) Project the matrix onto the tangent space generated by the base point, U_1 ($\mathcal{T}_U \mathrm{St}_n^m$), resulting in $\Delta_1 \in \mathcal{T}_U \mathrm{St}_n^m$. (3) Scale Δ_1 to within the radius of injectivity, yielding $\Delta_2 = \Delta_1 \, \|\Delta_1\|_w \times 0.89\pi\beta$, where $\beta \in [0, 1]$. (4) Utilize the matrix exponential map, $\mathrm{Exp}(\Delta_2)$, to compute a geometrically consistent U_2. A similar procedure can be applied to V_1 to obtain a new V_2, with both U_1 and V_1 residing on their respective Stiefel manifolds. These four steps are summarized in Figure 1.

Ultimately, the *core* of the *StiefelGen* algorithm involves generating new matrices U_2 and V_2. For practical use, it is advisable to apply a small smoothing filter over output, $\mathcal{T}_2$, since the majority of points over St_n^m appear as unstructured noise relative to the original signal. Meaningful data points do exist, but they can be rare as the manifold St_n^m encompasses a vast array of matrices. Smoothing becomes particularly useful as one moves far away from U_1 and V_1, almost approaching the injectivity radius. For small perturbations, smoothing may be unnecessary. The overall time complexity of *StiefelGen* is $O(mn^2)$[2], with efficiency boosts in the matrix exponential calculation for skew-symmetric systems in the retraction stage (Step 3 $\rightarrow$ Step 4)(Zimmermann and Hüper 2022; Edelman, Arias, and Smith 1998).

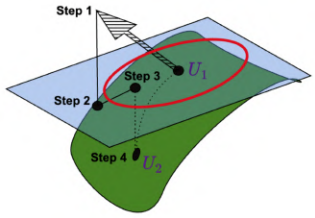


Figure 1: Summary of the basic steps taken in the *StiefelGen* algorithm.

Lastly, we clarify that Step 1 of Figure 1 has been practically implemented using functions written in the *Geomstats* library (Miolane et al. 2020), where they were in turn inspired by *PyManOpt* (Townsend, Koep, and Weichwald 2016) in that they first generate a random normal matrix in ambient $\mathbb{R}^{m \times n}$ space, then apply a vector projection onto $\mathcal{T}_U \mathrm{St}_n^m$. Furthermore, we again draw attention to the fact that *StiefelGen* requires *minimal* hyper parameters usage, and zero training. The only technically required hyper parameter is β, which describes how far towards the injectivity radius one intends to move. For generated signals which require a strong "outlier" or "novelty" flavour, one should opt for larger values of β, and for a standard data augmentation procedure, one would opt for a smaller value of β.

Results and Analysis

Throughout the rest of the main paper (and Appendix), diverse experiments will illustrate the versatile capabilities of *StiefelGen*. Applications range from perturbing time series signals along geodesics, seamlessly transitioning from conventional time series data augmentation to outlier signal generation—which is especially significant in synthetic structural health monitoring (SHM) scenarios. The approach is integrated into a UQ-driven dynamic mode decomposition (DMD) problem, as well as shown to be effective for generating synthetic data, enhancing an LSTM classification task's performance (a conventional dataa augmentation problem). This paper employs various datasets to showcase diverse experiments and use cases of *StiefelGen*, some in the main paper and others in the Appendix.

SteamGen: The dataset simulates steam flow telemetry from a fuzzy model mimicking a steam generator at Abbott Power IL. It includes output steam flow telemetry sampled every three seconds, totaling 9600 data points (Pellegrinetti and Bentsman 1996; Law 2023).

New York Taxi: Representing half-hourly averages of NYC taxi passengers recorded by the NYC government, this dataset spans a 75-day period in the Fall of 2014, containing 3600 data points. Anomalies include Columbus Day, Day-

[1]Be aware of the overloaded use of Σ as both, a summation symbol and that of the SVD singular value matrix.

[2]If $m < n$, then it will be $O(nm^2)$.

light Savings, and Thanksgiving (Law 2023, 2019; Cheema et al. 2023).

Synthetic SHM: Demonstrating practical applications of Stiefel geodesics in unconventional machine learning datasets, this synthetic Structural Health Monitoring (SHM) dataset emulates a bridge structure with five sensors recording 50 sinusoidal responses each, embedded with white noise and a constant bias. It spans 9 seconds at a recording frequency of 50Hz (Cheema et al. 2016; Anaissi et al. 2018). This dataset will be explored extensively in Appendix .

Spatio-Temporal DMD: Leveraging Stiefel manifolds in predicting future states of time series data, this dataset, inspired by dynamic mode decomposition (DMD), blends two spatio-temporal signals from a well-studied synthetic dataset (Schmid 2010; Kutz et al. 2016). Further details are in Appendix .

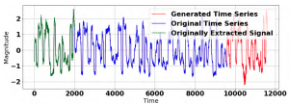
Japanese Vowels: This multivariate time series dataset captures the speech patterns of nine male speakers articulating the vowels 'a' and 'e' as a diphthong. Available on the UCI Machine Learning Repository (Kudo, Toyama, and Shimbo), it comprises 640 time series, each featuring 12 LPC coefficients derived from speech signals. Aimed at conventional time series data augmentation, it enhances the accuracy of an under-capacity LSTM model for classifying nine speakers. Further information and exploration is provided in Appendix .

StiefelGen: An Overview

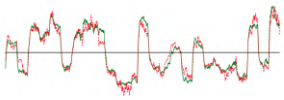
In this section, we explore the impact of generating augmentation instances with moderate perturbations over the Stiefel manifold and outlier instances with substantial perturbations. We focus on the SteamGen dataset for demonstration purposes.

Moderate Perturbation: For our initial examination, we apply a "moderate perturbation factor" ($\beta = 0.4$) and a smoothing factor ($\ell = 5$) to the first 2000 points of the SteamGen dataset. Reshaping the time series signal into a 50×40 page matrix and applying *StiefelGen* perturbation, the results are shown in Figure 2. Figure 2a presents a global view where the newly generated time series seamlessly integrates with the original signal through appending. To characterize the signal change, Figure 2d displays a histogram plot of the Δ values, unveiling a noise signature which deviates notably from standard probabilistic models. Figure 2b indicates shifts in the newly generated time series both in the aleatoric (noise) and epistemic (functional) sense. The reasons for this will be explored soon.

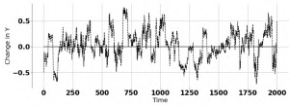
Large Perturbation: Shifting our attention to a larger perturbation factor ($\beta = 0.9$), we aim to generate seemingly anomalous signals that deviate significantly from the original without degrading into triviality (e.g., white noise). In contrast to the "moderate perturbation" discussed earlier, where the goal was signal augmentation, we now focus on generating signals with a more pronounced deviation. A slightly larger smoothing factor ($\ell = 9$) is employed to mitigate the impact of unwanted noise from the larger perturbation. Figure 3a immediately highlights a stark contrast in the globally viewed generated signal—both in magnitude and noise—compared to preceding portions. In particular, Figure 3b reveals a pronounced basis change with significant alterations in noise, offering valuable input for testing outlier detection systems. Examining the residuals in Figure 3d, a large deviation from standard probabilistic models is evident, with a broader dispersion of residual values and more pronounced tail behavior for the larger β value, indicating a generated signal significantly distant from the base reference signal.



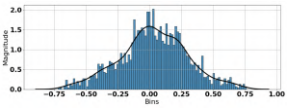
(a) A global view of the signal with the generated signal appended to the end.



(b) An over plot of the original signal with the newly generated signal.

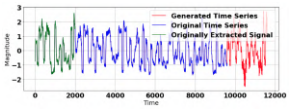


(c) A residual plot of the original and generated signals.

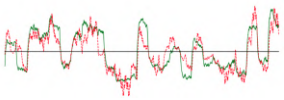


(d) A histogram over the residuals.

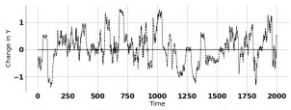
Figure 2: An overview of the effect of applying a moderate perturbation factor to generate a new signal based on the first 2000 points of input for the SteamGen data set.



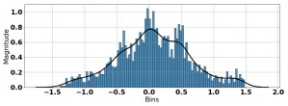
(a) A global view of the signal with the generated signal appended to the end.



(b) An over plot of the original signal with the newly generated signal.



(c) A residual plot of the original and generated signals.

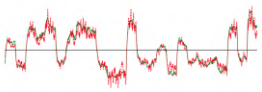


(d) A histogram over the residuals.

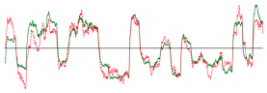
Figure 3: An overview of the effect of applying a large perturbation factor to generate a new signal based on the first 2000 points of input for the SteamGen data set.

Effect of Page Matrix Dimensions

In the prior investigations, we assumed $m = 50$ and $n = 40$ given the input signal's dimensions of $m \times n = 2000$ units long. Now, with the same β and ℓ hyperparameters, we opt for more extreme values of m and n to highlight the impact of working with tall-skinny page matrices or short-fat page matrices. The purpose is to demonstrate how the reshaped dimensions of the page matrix affect the generated time signal, with a focus on the moderate setting: a β factor of 0.4 (and thus $\ell = 5$). This provides a direct comparison with Figure 2b. The comparative results of these scenarios are illustrated in Figure 4.



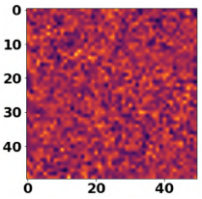
(a) The time series generated if a tall-skinny ($m = 100, n = 20$) page matrix were constructed and perturbed.



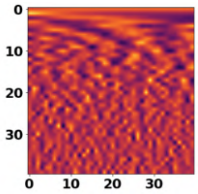
(b) The time series generated if a short-fat ($m = 20, n = 100$) page matrix were constructed and perturbed.

Figure 4: An overview of the effect of applying a moderate perturbation factor on the SteamGen data set for the two opposite cases of a tall-skinny, and short-fat shape respectively.

Contrasting the nearly square page matrix reshape scenario ($m = 50, n = 40$), a more extreme rearrangement produces distinct generated signals. One reshaping emphasizes noise dominance (Subfigure 4a), while the alternative introduces minimal noise but showcases a significant *basis change* (Subfigure 4b). This phenomenon becomes clearer when revisiting the square page matrix case ($m = 50$, $n = 40$). Visualizing the U and V matrices in Figure 5 reveals that the U matrix predominantly contains a noisy structure, while the V matrix exhibits more structured patterns, especially towards the top.



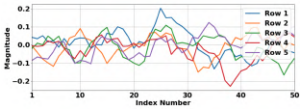
(a) A visualization of the U matrix.



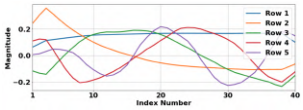
(b) A visualization of the V matrix.

Figure 5: A comparison between the U and V matrices as a result of an SVD of the $\mathcal{T}_{\text{mat}}$ Page matrix of the SteamGen input when $m = 50$ and $n = 40$.

Figure 6 reinforces this insight, especially in the V matrix (Figure 6b). The choice of reshaping in terms of rows (m) and columns (n) dictates whether the generated time series leans towards noise emphasis (aleatoric) or basis representation changes (epistemic).



(a) A plot of the first four rows of the U matrix.



(b) A plot of the first four rows of the V matrix.

Figure 6: A plot of the first four rows of the U and V matrices of the SVD of the SteamGen data, with smoothing factor set to $\ell = 3$ to clarify the presence of the basis information.

The distinct treatment of noise and basis information arises from the construction of the page matrix. In Equation 6, $\mathcal{T}_{\text{mat}}$ is formed by *stacking* portions of the uni-variate signal vertically. Reading $\mathcal{T}_{\text{mat}}$ from left to right reveals portions of the entire signal, while inspecting it top-to-bottom emphasizes noise variation as the signal compares itself every 50 data points ahead. This concept is illustrated in Figure 7.

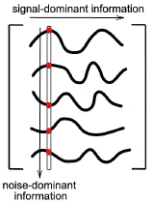


Figure 7: A demonstration of the stacking scheme used in the Page matrix construction of $\mathcal{T}_{\text{mat}}$.

Further, noting that a specific property of the SVD (and linear algebra in general) is that,

$$\mathcal{T}_{\text{mat},i} v_i = \sigma_i u_i, \quad \forall i = 1, 2, \ldots, n,$$
$$\mathcal{T}_{\text{mat},i}^{\intercal} u_i = \sigma_i v_i, \quad \forall i = 1, 2, \ldots, m,$$

we see that each u_i acts on $C(\mathcal{T}_{\text{mat}})$, the column space of $\mathcal{T}_{\text{mat}}$, and each v_i acts on $R(\mathcal{T}_{\text{mat}})$, the row space of $\mathcal{T}_{\text{mat}}$. And since signal dominant information is being stored row-wise, it is evident that this particular stacking scheme means short-fat stacking scheme (V-dominant) hyper-emphasises those perturbations which lead to a change in basis, whereas the tall-skinny stacking scheme (U-dominant) leads to perturbation-drive changes in signal information. Ultimately, m and n serve as hyperparameters, influencing noise or basis emphasis in generated signals. Combined with the scaling factor β, this yields a customizable spectrum of signal generation options. The approach is simple, interpretable, and requires no training. Inspection of U and V matrices provides direct insight into the empirical basis functions used for reconstruction. Notably, these basis functions differ from traditional wavelet or Fourier bases, being data-driven and aligning with maximal input data variation.

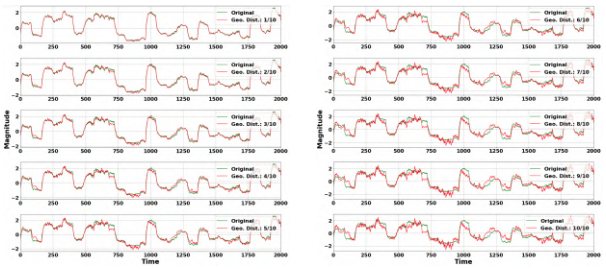
Working over Geodesics

In this section, we explore the smooth properties of Riemann manifolds, particularly on St_n^m, enabling precise control of signal perturbation by traversing geodesic paths between points U_1 and U_2 (or V_1 and V_2). This capability transitions signal augmentation into novel signal generation, proving valuable for potential applications in structural health monitoring. The Figure 8 illustrates this process with the *StiefelGen* algorithm, emphasizing the advantage that any generated time series signal can be adjusted seamlessly to achieve desired magnitude changes.

Applications of *StiefelGen*

Augmentation for an LSTM

For this task we built an LSTM model for classifying sequences in the *Japanese Vowels* dataset, sourced from



(a) The first 5 incremental geodesic steps.

(b) The last 5 incremental geodesic steps.

Figure 8: Incrementally deforming the generated signal until it arrives at the final position investigated previously in Figure 3.

the UCI machine learning repository (Kudo, Toyama, and Shimbo 1999). This multidimensional dataset involves nine male speakers articulating the vowels 'a' and 'e' in a phonetic diphthong common in Japanese. Each utterance was subjected to a "12-degree linear prediction analysis" (Atal and Hanauer 1971), resulting in a 12-dimensional feature vector per utterance. The dataset comprises 640 time series observations, with 270 for training and 370 for testing across nine speaker classes. Additional details are available online (Kudo, Toyama, and Shimbo). Evaluating *StiefelGen*'s impact involves varying the perturbation factor (β), exploring smaller to moderate percentages for data augmentation without strong outlier behavior. We consider $\beta \times 100$ values of $[0\%, 5\%, 10\%, 15\%, 30\%]$, with 0% representing the default condition. The experiment involves a multidimensional dataset without reshaping, and no additional smoothing due to short, minimally noisy sequences. The experiment unfolds as follows: (i) Investigating effects with varying n samples per class ($n = [5, 10, 15, 20, 25, 30]$, where $n = 30$ uses the entire training dataset). (ii) Selecting the synthetic data multiplier over n samples [gen $= [5, 10, 15, 20, 25, 30]$. (iii) Testing different percentage perturbations while iterating through (i) and (ii). For each β, an LSTM model is trained, and testing accuracy scores are averaged over 20 random seed iterations, providing mean and standard error estimates. Since this experiment is nested incredibly deeply, the exact problem setting (and the tables of all results) are made clear in Appendix . Regarding the LSTM model architecture, it comprises an initial bidirectional LSTM layer with 100 hidden units (Graves and Schmidhuber 2005). This is succeeded by a fully connected layer with 9 output units (matching the number of classes) and a softmax layer. Cross-entropy serves as the loss function, and training involves a mini-batch size of 64 over 50 epochs. These hyperparameters are chosen empirically for ample learning capacity, leading to high accuracy scores (testing accuracy ¿ 97%) (Mathworks 2023). However, in order to emphasize the effectiveness of the proposed data augmentation method using *StiefelGen*, the learning rate for the ADAM optimizer in *PyTorch* (Kingma and Ba 2014; Paszke et al. 2019) is deliberately set to 0.001. This choice ensures that the *PyTorch* LSTM model does not reach its full learning capacity, allowing for a simultaneous study of data augmentation impact on undertrained models.

Despite generating a substantial amount of data for the experiment, we focus on presenting the best results against the baseline model (non-augmented dataset). The baseline model's accuracy is detailed in Table 1, while Table 2 (corresponding to Figure 9) highlights the best training and testing scores, approximately doubling the vanilla model's performance. Quantitative results for further reference are available in Appendix . Notably, augmentation with *StiefelGen* consistently improves accuracy across all experimental runs, peaking at a training accuracy of 96% and testing accuracy of 94%, in contrast to the baseline model with 50% training accuracy and 42% testing accuracy.

Table 1: The result of using the first N elements from the each class of the data set, where $N = [5, 10, 15, 20, 25, 30]$. This is the base reference set of training and testing accuracies which have not received any data augmentation.

Pert. Level [%]	Accuracy [%]	5(×1)	10(×1)	15(×1)	20(×1)	25(×1)	30(×1)
0	Train	32.78±1.45	38.67±1.83	42.26±2.43	49.17±1.95	49.13±2.40	**50.02±2.22**
	Test	22.42±1.33	32.64±1.39	38.39±1.87	40.77±1.86	39.76±2.03	**41.53±2.33**

Table 2: Results of applying *StiefelGen* with various perturbation levels ($[5\%, 10\%, 15\%, 30\%]$, across various amounts of data taken from the dataset for generation $[5, 10, 15, 20, 25, 30]$ for augmentation, assuming 30 times the amount of data is generated.

Pert. Level [%]	Accuracy [%]	5(×30)	10(×30)	15(×30)	20(×30)	25(×30)	30(×30)
5	Train	94.79 ±1.70	93.94±3.04	**97.14±1.08**	87.02±4.49	93.76±2.43	88.88±2.76
	Test	61.12±2.12	76.53±3.06	**88.49±1.43**	80.62±4.29	88.05±2.02	85.20±2.61
10	Train	93.73±2.11	94.98±1.46	95.60±1.58	89.96±3.15	94.28±1.84	**96.35±1.01**
	Test	61.39±2.58	78.50±1.74	88.74±1.63	84.23±3.71	90.61±1.76	**93.55±0.77**
15	Train	92.28±2.22	96.79±1.27	94.24±1.71	91.29±4.44	**94.09±1.43**	93.16±1.49
	Test	61.97±2.56	80.49±2.10	89.46±1.13	87.35±4.25	**91.43±1.48**	90.49±1.72
30	Train	93.67±1.57	91.20±1.38	88.14±1.65	88.01±2.29	87.68±1.57	**90.47±0.88**
	Test	59.42±2.22	73.43±1.73	85.14±1.32	87.61±2.61	88.09±1.31	**91.14±0.83**

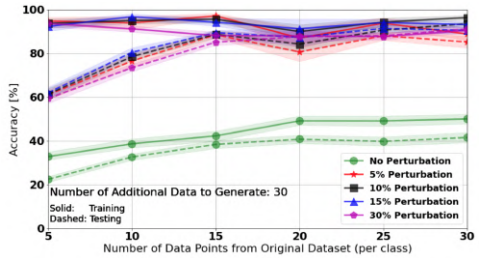


Figure 9: AAugmenting with gen $= 30$ times the number of observed points.

Spatio-Temporal Forecasting

DMD, originating from fluid dynamics, simplifies complex systems into spatio-temporal structures through regression over locally linear dynamics. It is widely used for short-term prediction and control. StiefelGen's compatibility with DMD lies in its mathematical structure. DMD involves snapshots, $\boldsymbol{\mathcal{T}_1}$ and $\boldsymbol{\mathcal{T}_2}$, assumed to vary linearly: $\boldsymbol{\mathcal{T}_2} = A\boldsymbol{\mathcal{T}_1} + \varepsilon^{\mathsf{T}}$. StiefelGen emerges as DMD requires SVD over $\boldsymbol{\mathcal{T}_1}$. Through a similarity transform, StiefelGen

perturbs the eigenbasis, impacting the linear dynamics (encoded in A). This perturbation projects the spatio-temporal signal forward in time through diverse pathways.

To illustrate the process, we employ an example from Kutz et al.'s textbook (Kutz et al. 2016) (Section 1.4). This example combines two spatio-temporal signals over the complex domain, as defined in Equation 11:

$$f(x,t) = f_1(x,t) + f_2(x,t) \tag{8}$$
$$= \text{sech}(x+3)\exp(i2.3t) + 2\text{sech}(x)\tanh(x)\exp(i2.8t), \tag{9}$$

Here, distinct spatial structures emerge in Equation 11 through the utilization of two frequencies, $\omega = 2.3$ and $\omega = 2.8$ (Kutz et al. 2016). This equation serves as the ground truth observation, initiating the iterative data-driven process of DMD. Figure 10 depicts some of the 200 total iteration frames. Figure 10 is formulated by perturbing the SVD process embedded in the DMD algorithm along 20 different paths with the UQ bounds, generated using functional data analysis (FDA). Specifically, a modified band depths approach for functional box plots is applied, extending the notion of a box plot to spatio-temporal dimensions.

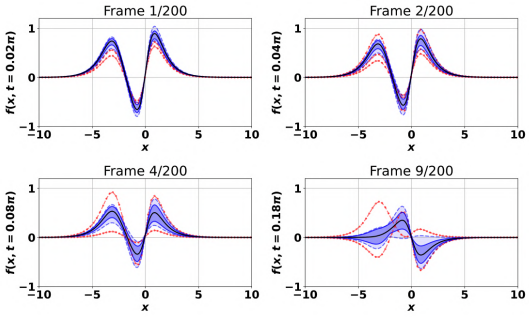


Figure 10: The first four exponentially spaced frames in a 200 frame sequence modeling the spatio-temporal dynamics of a signal, under the presence of induced uncertainty from *StiefelGen*, with dynamic functional box plots that change with time. The red dashed lines are outliers.

Conclusion

StiefelGen, is a model-agnostic time series data augmentation method based on Stiefel manifolds. It appears to be a very simple, flexible, and interpretable approach towards time series data augmentation. It requires minimal hyperparameter specification, (only technically necessitating β), and smoothly perturbs signals along geodesic paths. This approach appeared very successful in generating augmented data for model training and for outlier examples for robustness analysis.

Acknowledgments

This work was supported by JSPS, KAKENHI Grant Number JP21H03503, Japan and JST, CREST Grant Number JPMJCR22D3, Japan.

References

Absil, P.-A.; Mahony, R.; and Sepulchre, R. 2008. *Optimization algorithms on matrix manifolds*. Princeton University Press.

Agarwal, A.; Amjad, M. J.; Shah, D.; and Shen, D. 2018. Model agnostic time series analysis via matrix estimation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(3): 1–39.

Anaissi, A.; Makki Alamdari, M.; Rakotoarivelo, T.; and Khoa, N. L. D. 2018. A tensor-based structural damage identification and severity assessment. *Sensors*, 18(1): 111.

Atal, B. S.; and Hanauer, S. L. 1971. Speech analysis and synthesis by linear prediction of the speech wave. *The journal of the acoustical society of America*, 50(2B): 637–655.

Balakrishnan, A. v. 1962. On the problem of time jitter in sampling. *IRE transactions on information theory*, 8(3): 226–236.

Beckham, C.; Honari, S.; Verma, V.; Lamb, A. M.; Ghadiri, F.; Hjelm, R. D.; Bengio, Y.; and Pal, C. 2019. On adversarial mixup resynthesis. *Advances in neural information processing systems*, 32.

Chatfield, C. 2001. Prediction intervals for time-series forecasting. *Principles of forecasting: A handbook for researchers and practitioners*, 475–494.

Chatterjee, A. 2000. An introduction to the proper orthogonal decomposition. *Current science*, 808–817.

Che, Z.; Cheng, Y.; Zhai, S.; Sun, Z.; and Liu, Y. 2017. Boosting deep learning risk prediction with generative adversarial networks for electronic health records. In *2017 IEEE International Conference on Data Mining (ICDM)*, 787–792. IEEE.

Cheema, P.; Alamdari, M.; Chang, K.; Kim, C.; and Sugiyama, M. 2022a. Bridge indirect monitoring using Uniform Manifold Approximation and Projection (UMAP). In *Bridge Safety, Maintenance, Management, Life-Cycle, Resilience and Sustainability*, 997–1002. CRC Press.

Cheema, P.; Alamdari, M. M.; Chang, K.; Kim, C.; and Sugiyama, M. 2022b. A drive-by bridge inspection framework using non-parametric clusters over projected data manifolds. *Mechanical Systems and Signal Processing*, 180: 109401.

Cheema, P.; Alamdari, M. M.; Vio, G.; Azizi, L.; and Luo, S. 2023. On the use of matrix profiles and optimal transport theory for multivariate time series anomaly detection within structural health monitoring. *Mechanical Systems and Signal Processing*, 204: 110797.

Cheema, P.; Khoa, N. L. D.; Makki Alamdari, M.; Liu, W.; Wang, Y.; Chen, F.; and Runcie, P. 2016. On structural health monitoring using tensor analysis and support vector machine with artificial negative data. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, 1813–1822.

Damen, A.; Van den Hof, P.; and Hajdasinski, A. 1982. Approximate realization based upon an alternative to the Hankel matrix: the Page matrix. *Systems & Control Letters*, 2(4): 202–208.

DeVries, T.; and Taylor, G. W. 2017. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*.

Do Carmo, M. P. 2016. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications.

Edelman, A.; Arias, T. A.; and Smith, S. T. 1998. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2): 303–353.

Frühwirth-Schnatter, S. 1994. Data augmentation and dynamic linear models. *Journal of time series analysis*, 15(2): 183–202.

Fujiwara, K.; Takashima, R.; Sugiyama, C.; Tanaka, N.; Nohara, K.; Nozaki, K.; and Takiguchi, T. 2021. Data augmentation based on frequency warping for recognition of cleft palate speech. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 471–476. IEEE.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Graves, A.; and Schmidhuber, J. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks*, 18(5-6): 602–610.

Grosek, J.; and Kutz, J. N. 2014. Dynamic mode decomposition for real-time background/foreground separation in video. *arXiv preprint arXiv:1404.7592*.

Haradal, S.; Hayashi, H.; and Uchida, S. 2018. Biosignal data augmentation based on generative adversarial networks. In *2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, 368–371. IEEE.

Iglesias, G.; Talavera, E.; González-Prieto, Á.; Mozo, A.; and Gómez-Canaval, S. 2023. Data Augmentation techniques in time series domain: a survey and taxonomy. *Neural Computing and Applications*, 35(14): 10123–10145.

Iwana, B. K.; and Uchida, S. 2021a. An empirical survey of data augmentation for time series classification with neural networks. *Plos one*, 16(7): e0254841.

Iwana, B. K.; and Uchida, S. 2021b. Time series data augmentation for neural networks by time warping with a discriminative teacher. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 3558–3565. IEEE.

Kang, Y.; Hyndman, R. J.; and Li, F. 2020. GRATIS: GeneRAting TIme Series with diverse and controllable characteristics. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 13(4): 354–376.

Kim, G.; Han, D. K.; and Ko, H. 2021. Specmix: A mixed sample data augmentation method for training withtime-frequency domain features. *arXiv preprint arXiv:2108.03020*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kokoszka, P.; and Reimherr, M. 2017. *Introduction to functional data analysis*. CRC press.

Kudo, M.; Toyama, J.; and Shimbo, M. ???? Japanese Vowels. https://doi.org/10.24432/C5NS47. Accessed: 2025-01-11.

Kudo, M.; Toyama, J.; and Shimbo, M. 1999. Multidimensional curve classification using passing-through regions. *Pattern Recognition Letters*, 20(11-13): 1103–1111.

Kumar, D. N.; and Maity, R. 2008. Bayesian dynamic modelling for nonstationary hydroclimatic time series forecasting along with uncertainty quantification. *Hydrological Processes: An International Journal*, 22(17): 3488–3499.

Kutz, J. N.; Brunton, S. L.; Brunton, B. W.; and Proctor, J. L. 2016. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM.

Kutz, J. N.; Fu, X.; and Brunton, S. L. 2016. Multiresolution dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 15(2): 713–735.

Law, S. M. 2019. STUMPY: A powerful and scalable Python library for time series data mining. *Journal of Open Source Software*, 4(39): 1504.

Law, S. M. 2023. STUMPY Basics: Analyzing Motifs and Anomalies with STUMP. [Online; accessed 24-01-2024].

Lian, Y.; Wang, R.; and Jones, C. N. 2021. Koopman based data-driven predictive control. *arXiv preprint arXiv:2102.05122*.

López-Pintado, S.; and Romo, J. 2009. On the concept of depth for functional data. *Journal of the American statistical Association*, 104(486): 718–734.

Lou, H.; Qi, Z.; and Li, J. 2018. One-dimensional data augmentation using a Wasserstein generative adversarial network with supervised signal. In *2018 Chinese Control And Decision Conference (CCDC)*, 1896–1901. IEEE.

Lu, H.; and Tartakovsky, D. M. 2020. Prediction accuracy of dynamic mode decomposition. *SIAM Journal on Scientific Computing*, 42(3): A1639–A1662.

Mathworks. 2023. Sequence Classification Using Deep Learning. [Online; accessed 22-12-2023].

Maybank, P.; Peltzer, P.; Naumann, U.; and Bojak, I. 2020. MCMC for Bayesian uncertainty quantification from time-series data. In *Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part VII 20*, 707–718. Springer.

Meng, X.-L.; and Van Dyk, D. A. 1999. Seeking efficient data augmentation schemes via conditional and marginal augmentation. *Biometrika*, 86(2): 301–320.

Miolane, N.; Guigui, N.; Brigant, A. L.; Mathe, J.; Hou, B.; Thanwerdas, Y.; Heyder, S.; Peltre, O.; Koep, N.; Zaatiti, H.; Hajri, H.; Cabanes, Y.; Gerald, T.; Chauchat, P.; Shewmake, C.; Brooks, D.; Kainz, B.; Donnat, C.; Holmes, S.; and Pennec, X. 2020. Geomstats: A Python Package for Riemannian Geometry in Machine Learning. *Journal of Machine Learning Research*, 21(223): 1–9.

Osburg, J.; Ohlandt, C. J.; and Light, T. 2011. Pricing Strategies for NASA Wind-Tunnel Facilities.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.;

et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12: 2825–2830.

Pellegrinetti, G.; and Bentsman, J. 1996. Nonlinear control oriented boiler modeling-a benchmark problem for controller design. *IEEE transactions on control systems technology*, 4(1): 57–64.

Ramos-Carreño, C.; Torrecilla, J. L.; Carbajo-Berrocal, M.; Marcos, P.; and Suárez, A. 2022. scikit-fda: a Python package for functional data analysis. *arXiv preprint arXiv:2211.02566*.

Ramponi, G.; Protopapas, P.; Brambilla, M.; and Janssen, R. 2018. T-cgan: Conditional generative adversarial network for data augmentation in noisy time series with irregular sampling. *arXiv preprint arXiv:1811.08295*.

Rentmeesters, Q.; et al. 2013. *Algorithms for data fitting on some common homogeneous spaces*. Ph.D. thesis, Ph. D. thesis, Université Catholique de Louvain, Louvain, Belgium.

Schmid, P. J. 2010. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656: 5–28.

Schoellhamer, D. H. 2001. Singular spectrum analysis for time series with missing data. *Geophysical research letters*, 28(16): 3187–3190.

Shorten, C.; and Khoshgoftaar, T. M. 2019. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1): 1–48.

Small, M. 2005. *Applied nonlinear time series analysis: applications in physics, physiology and finance*, volume 52. World Scientific.

Spivak, M. 1999. *A comprehensive introduction to differential geometry*. Publish or Perish, Incorporated.

Stankeviciute, K.; M Alaa, A.; and van der Schaar, M. 2021. Conformal time-series forecasting. *Advances in neural information processing systems*, 34: 6216–6228.

Sun, Y.; and Genton, M. G. 2011. Functional boxplots. *Journal of Computational and Graphical Statistics*, 20(2): 316–334.

Townsend, J.; Koep, N.; and Weichwald, S. 2016. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *arXiv preprint arXiv:1603.03236*.

Tu, J.; Liu, H.; Meng, F.; Liu, M.; and Ding, R. 2018. Spatial-temporal data augmentation based on LSTM autoencoder network for skeleton-based human action recognition. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, 3478–3482. IEEE.

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Courville, A.; Mitliagkas, I.; and Bengio, Y. 2018. Manifold mixup: learning better representations by interpolating hidden states. *stat*, 1050: 4.

Verma, V.; Luong, T.; Kawaguchi, K.; Pham, H.; and Le, Q. 2021. Towards domain-agnostic contrastive learning. In *International Conference on Machine Learning*, 10530–10541. PMLR.

Wang, J.-L.; Chiou, J.-M.; and Müller, H.-G. 2016. Functional data analysis. *Annual Review of Statistics and its application*, 3: 257–295.

West, M.; Prado, R.; and Krystal, A. D. 1999. Evaluation and comparison of EEG traces: Latent structure in nonstationary time series. *Journal of the American Statistical association*, 375–387.

Xiao, Z. 2012. Time series quantile regressions. In *Handbook of statistics*, volume 30, 213–257. Elsevier.

Xu, Q.; Zhang, R.; Zhang, Y.; Wang, Y.; and Tian, Q. 2021. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14383–14392.

Yuan, Y.; Zhou, K.; Zhou, W.; Wen, X.; and Liu, Y. 2021. Flow prediction using dynamic mode decomposition with time-delay embedding based on local measurement. *Physics of Fluids*, 33(9).

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6023–6032.

Zhang, G. P. 2007. A neural network ensemble method with jittered training data for time series forecasting. *Information Sciences*, 177(23): 5329–5346.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, X.; Chowdhury, R. R.; Shang, J.; Gupta, R.; and Hong, D. 2023. Towards Diverse and Coherent Augmentation for Time-Series Forecasting. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Zhang, X.; Zhao, Z.; Tsiligkaridis, T.; and Zitnik, M. 2022. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems*, 35: 3988–4003.

Zimmermann, R.; and Hüper, K. 2022. Computing the Riemannian Logarithm on the Stiefel Manifold: Metrics, Methods, and Performance. *SIAM Journal on Matrix Analysis and Applications*, 43(2): 953–980.

Applications of *StiefelGen* to Structural Health Monitoring

This subsection delves into the application of *StiefelGen* for studying both robustness and adversarial data generation in the context of structural health monitoring (SHM). We shall focus on a classic problem and approach in SHM involving stacking sensor data into a matrix format. Subsequently, dimensionality reduction (PCA) is applied to this

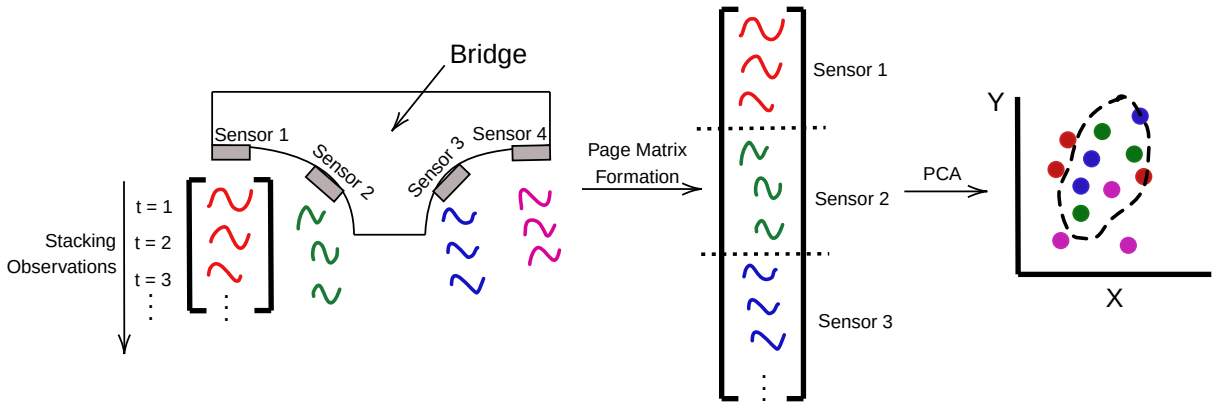


Figure 11: An example of the conventional data-drive approach in SHM which involves collecting data across a sensor array network, stacking the data into a matrix-like form, and then projecting the data for analysis in lower dimensional spaces.

matrix, followed by training a one-class support vector machine (OCSVM). The workflow for this problem is exemplified in Figure 11.

Figure 11 illustrates the collection of multiple observations at each sensor location over time, organized into a structured data format. Typically, all the data is consolidated into a larger matrix structure, followed by projection to lower-dimensional spaces (Cheema et al. 2022a,b). In these spaces, the conventional approach involves training a one-class support vector machine (OCSVM) algorithm. This is due to the ethical constraint that prevents engineers from intentionally damaging deployed structures for the purpose of simply obtaining "damage data" in order to facilitate a two-class SVM analysis.

The use of *StiefelGen* is particularly well-suited for this scenario because the collected data is often stored in a structured matrix form, interpretable as an "already formed" page matrix. Given the structured nature of the data, it can be treated as a page matrix with pre-selected values for m and n, eliminating the need for reshaping as required for univariate time series signals. Therefore one set of hyper parameters (selection of m and n) can be completely ignored in practice (if desired). For the SHM problem, without loss of generality, we shall assume identical frequency and duration of measurements across sensors for simple stacking. If there is non-uniformity in the sensor array network, *StiefelGen* analysis can be performed on a per-sensor basis.

In the context of civil engineering, two critical questions arise: (i) How much signal deviation in the original measurement space can be accommodated before official "damage detection" is triggered? This poses a model robustness problem, specifically gauging the tolerance for deviation before the OCSVM detects damage. (ii) Are there model signals, severely perturbed to the extent that they should be recognized as damage, but remain undetected by the OCSVM model? Such cases represent adversarial signals for the model. We posit that *StiefelGen* can simultaneously generate sets of time series signals to address both tasks, providing engineers with a nuanced understanding of their chosen outlier detection model, which would be otherwise challenging to achieve in practice.

For this investigation, we employ a simplified model of a SHM problem. The model consists of a toy bridge structure equipped with five sensors, with fifty observations recorded per sensor. The sensor readings have a frequency of 50Hz, and each observation spans a total duration of nine seconds. We assume a small bridge size with symmetrically placed sensors, ensuring similar modal information per sensor. Consequently, variations in observations across different sensors primarily stem from noise, indicating a predominantly aleatorically imposed uncertainty distribution in sensor space. The data generation model applied across the sensor space is as follows,

$$S = 4\sin\left(6\pi t^{0.5}\right) + \sin\left(15\pi t\right) + \mathcal{N}(1, 0.5), \qquad (10)$$

where $t \in \mathbb{R}_{\geq 0}$ represents the time variable, and the mean of 1 used in the Gaussian term represents an arbitrary bias term. As mentioned earlier, if the observed noise model (or underlying epistemic uncertainty) varies greatly between sensors, then *StiefelGen* should be used on a per-sensor basis instead. Lastly, the OCSVM model used in these analysis is based on that found in the Scikit-learn package with $\nu = 0.1$, and $\gamma = 10^{-3}$ (Pedregosa et al. 2011).

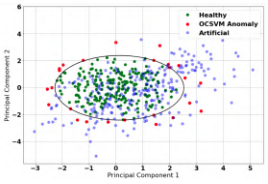
StiefelGen: Robustness in SHM

For the *StiefelGen* analysis in this context, a single large perturbation is applied to each data point once, taking $\varepsilon = 1$. Subsequently, the final outputs are examined in a 2D PCA plot. The expectation is that most, if not all, input data points in the 2D projected space will have undergone a significant movement, potentially surpassing the trained OCSVM boundary. To analyze the impact of this shift, the data point

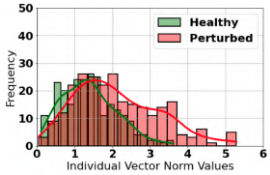
with the largest deviation (with respect to the Euclidean norm) is identified by comparing the norms of the projected data points before and after perturbation, and then considering the top K data points changes. In this case, we shall just focus on $K = 1$, that is, the signal which experienced the most substantial change for demonstration purposes. While analyzing a single data point that underwent considerable deviation might not seem advantageous on the surface, because these perturbations occurred over a smooth manifold within/on the radius of injectivity, there then exists a unique geodesic path from the starting position to the end position. This concept allows for the gradual tracking of signal deviations over the data manifold, determining the point at which the original signal's deviations become substantial enough to reach the edge of the OCSVM boundary. This approach facilitates the inspection of the OCSVM model's efficacy and addresses questions such as: "what levels of deviation are acceptable until the OCSVM boundary is breached?" This is a crucial consideration for model robustness ("what exactly is happening on the OCSVM boundary and what data points cause the model to fail?"), and the ability to follow a data geodesic is essential for providing meaningful answers to this question. The impact of this approach in the context of structural health monitoring (SHM) is illustrated in Figure 12(c).

In Figure 12(c), the initial projected point is denoted by the dark green star, and its final position after a large perturbation is represented by the solid red star. Intermediate points, totaling eighteen stars (there are twenty stars in total considering the initial and final points), are depicted as light-red stars. Leveraging the properties of *StiefelGen*, we can pinpoint the moment when perturbations become significant enough to push the initial point just outside the OCSVM boundary. This capability allows SHM engineers to critically assess the model's efficacy by gauging how much signal change is allowable at the point of reaching this boundary. Consequently, they can comment on the overall robustness of the model concerning permissible levels of perturbations before triggering an alarm for potential damage.

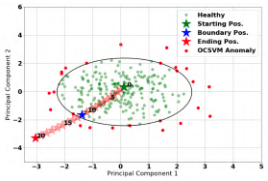
However, a key signal to inspect for assessing the OCSVM model's robustness is that of the blue star. Upon plotting its corresponding time series signal, it is apparent that it has also shifted significantly from its original position yet remains within the OCSVM boundary. In principle, it should technically be classified as a healthy signal according to the OCSVM model, but for the SHM engineer this signal has deviated so much that it should invariably represent a damage state for the structure (Cheema et al. 2023). This paradox underscores the insufficiency of the learned OCSVM model, as expected in this somewhat naïve implementation. Drawing this conclusion would be challenging without the ability to smoothly deform a signal from its starting to its ending position, showcasing an application benefit of *StiefelGen*. These results are particularly positive, given that no model assumptions were made except for the diffeomorphic nature of signal changes, and that the set of allowable deformations shall lie on the Stiefel manifold with respect to its corresponding SVD decomposition. Moreover, in this case, the m and n values typically needed to reshape



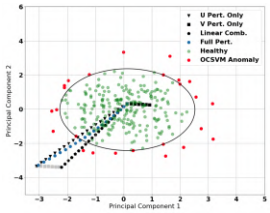
(a) Plot of the initial projected data samples (green), and the consequence of perturbing to the input data to radius of injectivity (blue). The red points refer to the outlier points used in the training of the OCSVM of the original dataset (green), given the OCSVM hyper parameters.



(b) A histogram of the L_2 norms before and after the perturbation (that is comparing the healthy and the perturbed datapoints).



(c) Showcase of following a geodesic between a point which is initially approximately in the centre of the OCSVM boundary, and its resultant point. The point at which the OCSVM boundary is to be crossed over is given by the blue star.

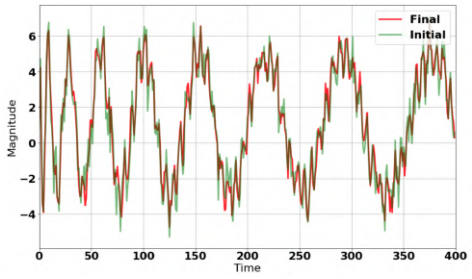


(d) Outline of the difference between either following the U perturbation only, and the V perturbation only, in relation to when the $U - V$ perturbation is jointly performed.

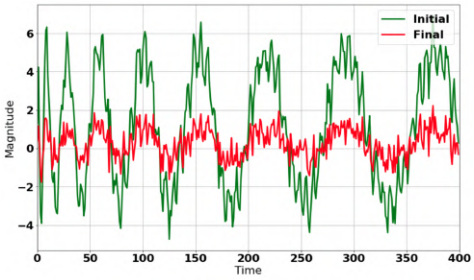
Figure 12: An example of the effect of exploiting the geodesics in the StiefelGen algorithm.

a 1D signal were not required, as they were implicitly chosen when stacking signals from each sensor. Thus, the sole hyper parameter required was that of the percentage of perturbation. However, this choice is easily justified by simply considering the we desire the maximum possible perturbation of 100%, placing us at the edge of the radius of injectivity, because then we were able to smoothly follow along the geodesic until reaching the OCSVM boundary. In principle then, no parameters, no hyper parameters, or any model training was required. Further, minimal data assumptions had to be made in practice. This approach to studying the robustness properties of the OCSVM boundary of allowable signals, is significantly more straightforward than the alternative: which would be attempting to estimate the level of perturbation required to push the initial point onto the OCSVM boundary by varying the perturbation levels and "shooting" the signal forward.

Subfigure 12(d) illustrates the distinct effects of following individual U or V geodesics. As established earlier, depending on the stacking procedure, either the U (column space) or V (row space) geodesic tends to influence a basis or noise change more prominently. The consequences of traversing these directions concerning the original signal space are elucidated in Figure 13.



(a) Noise deviation direction (V).



(b) Basis deviation direction (U).

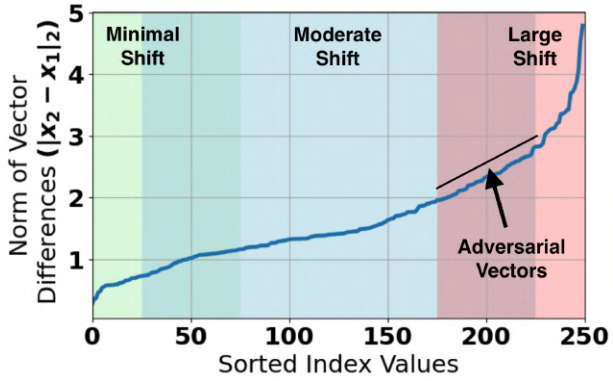
Figure 13: The difference between following along either U and V directions individually, or in a combined fashion, in relation to the original signal space.

Here, we see that indeed there has been changes to the variation in the noise pattern for Subfigure 13(a), and indeed a change in the basis functions in Figure 13(b). However, it is crucial to reiterate that the terms "noise direction" and "basis direction" are empirical approximations.

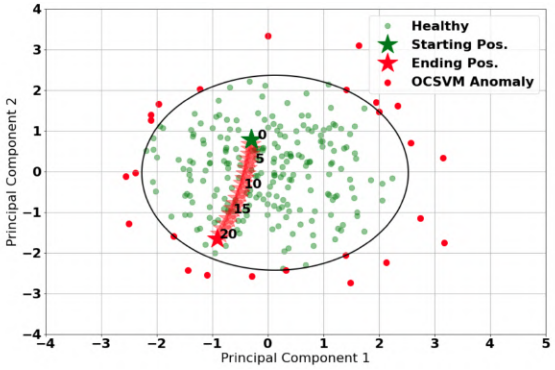
StiefelGen: Adversarial Data Generation in SHM

In the SHM domain, addressing the adversarial data generation problem involves answering a critical question: *What does it take for damage signals to pass through my model and be misclassified as healthy?* To leverage *StiefelGen* for this purpose, we follow a simple two-step procedure as a means to generate and search for potential adversarial examples. First, we calculate the Euclidean norm of the difference between the maximally perturbed ($\varepsilon = 1$) and non-perturbed ($\varepsilon = 0$) data points in the projected space. For each data point x_1 in its healthy state and x_2 in its perturbed state, the quantity $|x_2 - x_1|_2$ is computed and sorted, as illustrated in Subfigure 14(a). Upon inspecting the plot, notable features indicative of adversarial vectors are identified. The plot exhibits a sigmoidal shape with two inflection points, roughly dividing it into three distinct portions. The first region corresponds to minimal norm changes, suggesting little alteration in the underlying time series. The final portion represents time series that have undergone significant shifts, likely resulting in the data point leaving the OCSVM boundary. In order to explore for adversarial samples focus is placed upon the intersection between large norm shifts (indicating the presence of substantial data movement) and moderate norm shifts (suggesting that whilst the data exhibited large movement, there is a reasonable chance it did not leave the OCSVM boundary). While not mathematically rigorous, this heuristic approach parallels the elbow method used in k-means clustering, albeit instead of looking for diminishing returns, we look for the point of rapidly increasing (norm) returns. In this study, the 85th percentile, centered with the second inflection point in Figure 14(a), leading to the selection of the first data point after the 85th percentile as the adversarial example for exploration.

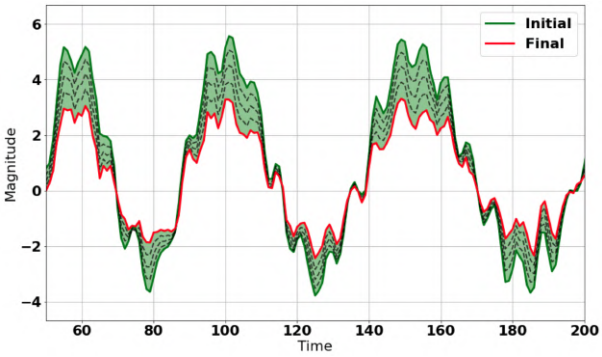
This data point exhibits perturbation geodesics in the projected 2D space as shown in Subfigure 14(b), accompanied by the corresponding time series signals visualized in Subfigure 14(c). Notably, two key observations emerge: (i) Throughout its geodesic path, the perturbed data point consistently remains within the healthy boundary defined by the OCSVM, and (ii) The change in the time series function escalate fairly rapidly, indicating an early manifestation of damage well before reaching its final position. Such an analysis serves as a valuable case study for the comprehensive examination of the structural health monitoring (SHM) problem. Therefore, *StiefelGen* not only facilitates the exploration of model robustness by scrutinizing the boundary behavior of the OCSVM model concerning potential signals (prior subsection) but also enables the generation of adversarial examples. These nuanced investigations are challenging to conduct with other time series data augmentation methods, as (to the best of the author's knowledge) they tend to lack the combined capability of smoothly deforming signals from an augmentation state to a novelty state, without the need for pre-training or relying on large data sets.



(a) The change in L_2 norms before and after perturbation is applied. Here x_1 refers to the healthy state, and x_2 the perturbed (damage) state.



(b) The geodesic path of the first data point which satisfied the condition that $\|x_2 - x_1\|$ is over the 85-th percentile.



(c) The signal as it moved from the healthy to the maximally perturbed state. The black dashed lines represent every 5-th step along the geodesic.

Figure 14: Plots which show how to identify, generate, and analyse adversarial time series signal using StiefelGen for the SHM case study.

StiefelGen: Uncertainty Quantification and Linear Time Series Dynamics

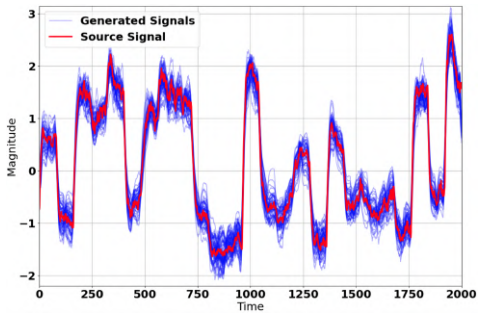
In this subsection, we shall delve into the compatibility of *StiefelGen* for a joint application of uncertainty quantification (UQ) and dynamic mode decomposition (DMD) over spatio-temporal signals. Traditional UQ approaches in time series often involve creating prediction intervals that expand over time (Chatfield 2001; Xiao 2012). These methods often incorporate Bayesian analysis by providing a probabilistic framework over the signal by leveraging prior knowledge of the signal generating process (Kumar and Maity 2008). Monte Carlo methods are also frequently employed (Maybank et al. 2020), and more recently, conformal predictions have gained attention due to its theoretical guarantees and distribution-free claims (Stankeviciute, M Alaa, and van der Schaar 2021).

Funtional Data Analysis

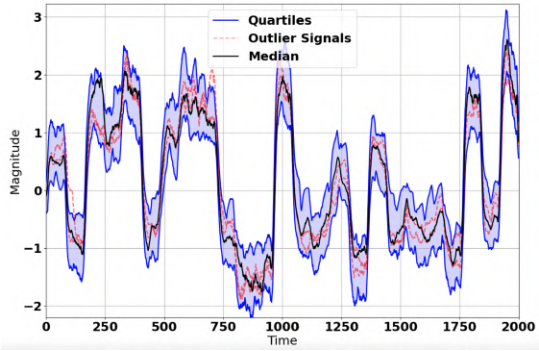
At first glance, *StiefelGen* may not seem to work well with conventional UQ methods. However, its strength lies in its ability to swiftly generate novel time series augmentations, with the required level of signal novelty scaled against the sampling distance away from the injectivity radius. Consequently, *StiefelGen* is well-suited for a UQ-based approach to time series analysis through functional data analysis (FDA) (Wang, Chiou, and Müller 2016). FDA encompasses a suite of statistical methods designed for analyzing data that varies over a curve, surface, or continuum. In this context, observations are treated as *functions*, emphasizing the analysis of datasets where the primary units of measurement are curves or functions, often observed over a continuous domain such as time. Mathematically we shall consider that *StiefelGen* has generated a set of time series augmentations: $\boldsymbol{\mathcal{T}}(t) = \{\mathcal{T}_i(t)\}_i^K$, for K possible input time series, defined over a closed interval $t \in [0, 1]$. Using this construction, $\{\mathcal{T}_i(t)\}_i$ may be interpreted as a collection of stochastic random variables. Then, assuming a finite square integrability condition as: $\mathbb{E}\left(\int_0^1 |\boldsymbol{\mathcal{T}}(t)|^2\, dt\right) < \infty$, one can generalize typical statistical quantities over function spaces. For example, the mean function of $\boldsymbol{\mathcal{T}}$ can be taken to be as $\mathbb{E}\langle \boldsymbol{\mathcal{T}}, h\rangle = \langle \mu, h\rangle$, where μ is unique, and $\mu, h \in \mathcal{H}$, are separable Hilbert spaces of square-integrable functions (Kokoszka and Reimherr 2017).

In employing FDA for UQ, we extend the conventional statistical box plot to *functional box plots* (Sun and Genton 2011) as part of our approach. Specifically, we leverage *StiefelGen* to swiftly generate diverse instances of the provided reference uni-variate time signal. Subsequently, a functional box plot is constructed to encapsulate key descriptive statistics, such as the median signal and the envelope of the 50% central region. This methodology becomes evident when applied to the SteamGen dataset. Figure 15(a) illustrates the rapid generation of 500 time series signals, employing a scaling factor of $\beta = 0.3$, and identical m (50) and n (40) reshape values as outlined in Section . The resulting functional box plot is illustrated in Figure 15(b).

In order to create Figure 15(b), the sci-kit fda library was utilized (Ramos-Carreño et al. 2022), and adopted the mod-



(a) Over plotting of multiple generated signals based on the original reference signal.



(b) An example of plotting the 50% central envelope of the functional box plot around the median signal.

Figure 15: An example of applying FDA to allow for *StiefelGen* to be used for the purpose of UQ.

ified band depth (MBD) approach for functional box plot generation. This method was chosen for its simplicity, interpretability, and parameter-free nature. Briefly, the MBD computation relies on the arrangement of points concerning a reference point or curve in the data set. Typically, the median function serves as the reference point when aiming to calculate the functional box plot, offering a natural choice for assessing depth centrality within a distribution (López-Pintado and Romo 2009). Given this reference point, MBD then calculates the proportion of curves inside bands surrounding the target curve, providing a depth measure that reflects its position relative to the distribution of functional data in the dataset. Finally, it is crucial to emphasize that functional data lacks a universally accepted definition for outliers. This ambiguity arises because a curve can be identified as an outlier based on different criteria, including but not limited to substantial distance from the mean (magnitude outlier) or a distinctive pattern compared to other curves (shape outlier). In essence, a curve is typically classified as an outlier if it stems from a distinct underlying process (López-Pintado and Romo 2009).

The ability for *StiefelGen* to perform UQ in this manner will be shown to be highly beneficial when implemented along side dynamic mode decomposition (DMD) for the purpose of perturbing system dynamics to obtain a multitude of different spatio-temporal signal evolutions.

Dynamic Mode Decomposition

Dynamic mode decomposition (DMD) originated within the fluid dynamics community as a means to break down intricate flows into a simplified representation based on spatio-temporal coherent structures (Schmid 2010). The increasing success of DMD over time can be attributed to its equation-free, data-driven nature, allowing for an accurate deconstruction of complex systems into spatio-temporal coherent structures (Kutz et al. 2016), as the DMD algorithm can be leveraged for short-term future-state prediction and control (Lu and Tartakovsky 2020; Yuan et al. 2021; Grosek and Kutz 2014). Algorithmically, DMD relies upon collecting a set of data snapshots, denoted as x_k, from a dynamical system at various time points, where $k \in \mathbb{N}$. DMD then performs a regression of these snapshots over locally linear dynamics, represented as, $x_{k+1} = Ax_k$, where the A matrix holds the locally linear system physics, and is chosen to minimize $\|x_{k+1} - Ax_k\|_2$. DMD's advantages lie in its simplicity of execution and the minimal assumptions it imposes on the underlying system, which parallels the approach taken by *StiefelGen*. It has also seen a recent re-surge of interest due to its close relationship to the *Koopman operator* theory (Kutz et al. 2016). The reason why *SteifelGen* readily works with the DMD framework can be seen if one inspects its mathematics. Assume one has two separate snapshots of input data as:

$$\boldsymbol{\mathcal{T}_1} = [\mathcal{T}_1, ..., \mathcal{T}_{N-1}]$$
$$\boldsymbol{\mathcal{T}_2} = [\mathcal{T}_2, ..., \mathcal{T}_N]$$

As made clear, these snapshots are assumed to vary ac-

cording to a *linear* dynamical system with aleatoric uncertainty orthogonal :

$$\mathcal{T}_{1,i+1} = A\mathcal{T}_{1,i}$$

for some state space matrix, A. Written in its matrix form and adding aleatoric uncertainty, $\boldsymbol{\mathcal{T}_2} = A\boldsymbol{\mathcal{T}_1} + \varepsilon^{\intercal}$. Now the manner in which *StiefelGen* arises, is that the base DMD algorithm requires calculating the SVD over $\boldsymbol{\mathcal{T}_1}$, leading to: $\boldsymbol{\mathcal{T}_2} = AU\Sigma V + \varepsilon^{\intercal}$. Assuming that the residuals ε, remain orthogonal to the minimizing basis found through DMD (often termed as the proper orthogonal decomposition modes (Chatterjee 2000)), one can left-right multiply the matrix, A, in order to discover the quintessential DMD relationship: $U^{\intercal}AU = U^{\intercal}\boldsymbol{\mathcal{T}_2}V\Sigma^{-1} = \tilde{S}$. Now, since A and $\tilde{S}$ are related through a *similarity transform* (Kutz et al. 2016), one can find the eigenbasis for $\tilde{S}$, in order to obtain precisely the required eigenbasis for A. If one were to use *StiefelGen* to *perturb* the U and V matrices in this similarity transform, then what one is doing is *perturbing* the linear dynamics of the system (encapsulated in the A matrix), thereby projecting the spatio-temporal signal ahead in time through a multitude of pathways.

As A and $\tilde{S}$ are connected through a similarity transform, finding the eigenbasis for $\tilde{S}$ allows one to precisely determine the required eigenbasis for A. Thus information regarding the linear state evolution matrix A, can be directly obtained simply be observing the empirical relationship between $\boldsymbol{\mathcal{T}_1}$, and $\boldsymbol{\mathcal{T}_2}$. Ultimately, since the SVD is used to determine the evolutionary dynamics of the spatio-temporal system, *StiefelGen* may be used to perturb the U and V matrices within this similarity transform, leading to different spatio-temporal path ways of the signal. In effect this imbues the dynamics with a non-trivial form of epistemic uncertainty. This integration presents a noteworthy extension of *StiefelGen*'s utility beyond simply conventional time series data augmentation.

In order to demonstrate how this works we utilize the example shown in Section 1.4 of the textbook by Kutz et al. (Kutz et al. 2016) which involves mixing two spatio-temporal signals over the complex domain, clarified in Equation 11.

$$\begin{aligned} f(x,t) &= f_1(x,t) + f_2(x,t) \qquad (11) \\ &= \text{sech}(x+3)\exp(i2.3t) + 2\text{sech}(x)\text{tanh}(x)\exp(i2.8t), \end{aligned}$$

where the use of two separate frequencies, $\omega = 2.3$ and $\omega = 2.8$ allows for distinct spatial structures to arise (Kutz et al. 2016). It is crucial to emphasize that Equation 11 functions solely as the ground truth observation, initiating the iterative data-driven process of DMD. Furthermore, in alignment with the methodology outlined in (Kutz et al. 2016), we adopt the reduced rank versions of matrices U and $V^{\intercal}$, specifically retaining only the first two columns after performing the manifold perturbation. Lastly, it's worth observing that the integration of *StiefelGen* into DMD does not necessitate any smoothing or rearrangement of the page matrix, re-emphasizing the lack of need for hyper-parameter selection. The only hyper parameter technically required would once again be that of β which was chosen to be 0.2 without loss of generality.

To illustrate the impact of *StiefelGen* on the spatio-temporal evolution, we generated a visual representation by plotting the ground truth signal against a diverse set of solution pathways. These pathways correspond to different perturbed A matrices, as depicted in Figure . The real component of the spatio-temporal signal is presented over a 4π seconds interval for clarity. To ensure a varied degree of spacing across the 200 iterations, an exponential spacing scheme was employed for the "frames". This choice results in closely spaced initial frames, gradually growing apart, effectively highlighting the increasingly deviant paths taken by the perturbed signals.

As observed in Figure , the perturbed signals consistently align with the overall modal behavior and exhibit smoothness properties akin to those of the original signal. This smoothness arises naturally from the inherent properties of the Stiefel manifold, thereby offering an empirical validation of the influence exerted by *StiefelGen* on the spatio-temporal signal evolution.

Expanding on this analysis to include a temporal functional box plot for the spatio-temporal signal, Figure illustrates the effective tracking of the 50% central region over time. To showcase the versatility of the approach and highlight the dynamic emergence of functional outlier data, an additional outer envelope representing the 75% central region was chosen. This envelope not only underscores the flexibility of the method but also demonstrates the dynamic nature in which functional outlier data may manifest, as defined relative to the aforementioned Modified Band Depth (MBD) method.

Ultimately, the *StiefelGen* algorithm appears to seamlessly integrate into the DMD framework, facilitating the exploration of induced epistemic uncertainty levels within linear dynamical models in a manner which respects the underlying geometry of its SVD step. This integration holds promise as a valuable data augmentation tool for projecting time series forward. Nevertheless, it's crucial to acknowledge the current limitations of DMD in projecting forward sets of signals with pronounced nonlinear behavior, excessive chaos, or non-stationary properties (Kutz et al. 2016). A notable effort within the DMD community to address some of these limitations include multi-resolution DMD, a recursive, hierarchical extension of the conventional DMD algorithm, designed to mitigate drawbacks by separating microscale and macroscale effects (Kutz, Fu, and Brunton 2016).

On Applications to LSTM Training

In addition to the unique ways in which *StiefelGen* can be used in many applications, we explore its conventional, implied usage as a synthetic data generator for the purpose of increasing the amount of overall training data of a deep, network model. In order to investigate this, we constructed an LSTM model for the purpose of sequence data classification for the *Japanese Vowels* dataset from the UCI machine learning repository (Kudo, Toyama, and Shimbo 1999). This dataset is a multidimensional dataset in which nine male

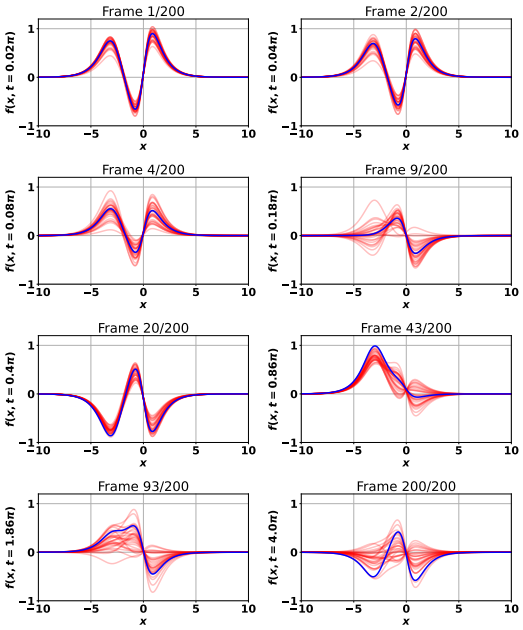


Figure 16: A plot of different spatio-temporal states of the ground truth signal (blue) against thirty separate solution pathways for a perturbed A matrix with $\beta = 0.2$.

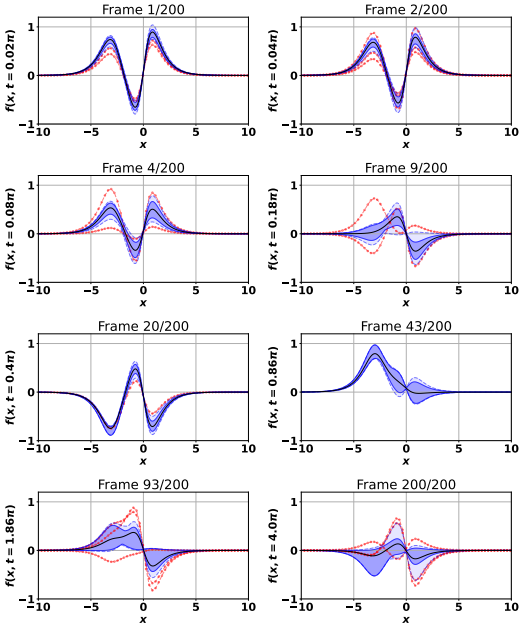


Figure 17: A plot of different spatio-temporal states of the functional box plot. The median signal (black), is encompassed by the 50% central region (dark blue shading with solid blue boundary), as well as the 75% envelope (ligh blue shading with dashed blue lines). The outlier signals are shown in red, and are defined to be those signals outside of the inter-quartile range

speakers uttered two vowels in sequence ('a' and 'e') reflective of a phonetic dipthong, /ae/ that is common in Japanese. On each utterance of the dipthong, a "12 degree linear prediction analysis" (Atal and Hanauer 1971) was performed by splitting each utterance waveform into a 12 dimensional feature vector (12 separate coefficients) input per utterance, where each individual dimension (coefficient) may be anywhere from 7 to 29 units long. The total number of time series (experimental observations) used in this study was 640, with 270 time series for training (30 measurement instances spread amongst the 9 speaker classes) and the other set of 370 time series for testing (24 - 88 measurement instances for each class, spread this time non-equally amongst the same 9 speaker classes). Further information such as sampling rate, frame lengths, and shift lengths (which are not relevant to the current discussion) may be found online (Kudo, Toyama, and Shimbo).

Given that *StiefelGen* operates primarily through a perturbation factor/percentage (where $\beta = 1 \rightarrow 100\%$ implies lying directly on the radius of injectivity), a comprehensive assessment of *StiefelGen* necessitates exploring the impact of varying this percentage value. The goal is to generate additional data that is "similar enough" to the input data for augmentation. To achieve this, we focus on smaller to moderate percentage ranges, avoiding training on strong outlier behavior. Hence, we selected percentage perturbation factors ($\beta \times 100$) of: $[0\%, 5\%, 10\%, 15\%, 30\%]$, where 0% signifies the default operating condition (applied perturbation). The increasing percentages reflect an escalation in variance within the augmented datasets. It's noteworthy that, since we are working with a multidimensional dataset, there is no need for any *reshape* operation on the input data, as it naturally stacks in a matrix format (each $\boldsymbol{x}_i$ is already a matrix). Additionally, we did not introduce any additional smoothing, as the data sequences are relatively short on a per-experiment/observation basis and exhibit minimal noise.

The nested structure of this experiment unfolds as follows: (i) Investigating the effects of taking the first n samples per class, which allows the experiment to explore low data cardinality settings ($n = \{5, 10, 15, 20, 25, 30\}$ where $n = 30$ implies utilizing the entire training dataset consisting of 30 observations across 9 classes). (ii) Selecting the amount of additional data to generate, acting as the synthetic data multiplier over the n samples (gen $= \{5, 10, 15, 20, 25, 30\}$). For instance, choosing $n = 10$ samples per training class and opting to generate gen $= 15$ times more data results in working with $n$gen $= 1015 = 150$ synthetic data samples, across each of the 9 classes (The original $n = 15$ data samples used for generating synthetic data are entirely discarded for simplicity so that one doesn't need to work with $15 + 150 = 165$ augmented data samples per class). (iii) Testing various levels of percentage perturbations whilst iterating through points (i) and (ii) concurrently. For each new percentage perturbation factor, an LSTM model was trained and its testing accuracy scores were grouped 20 random seed iterations. Naturally, this allows for the generation of reasonable standard error uncertainty bar estimates and mean value estimates per experiment. The overall nested structure of the experiment's flow

is visually represented in Figure 18.

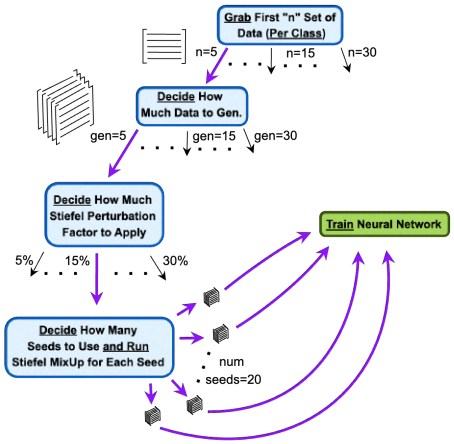
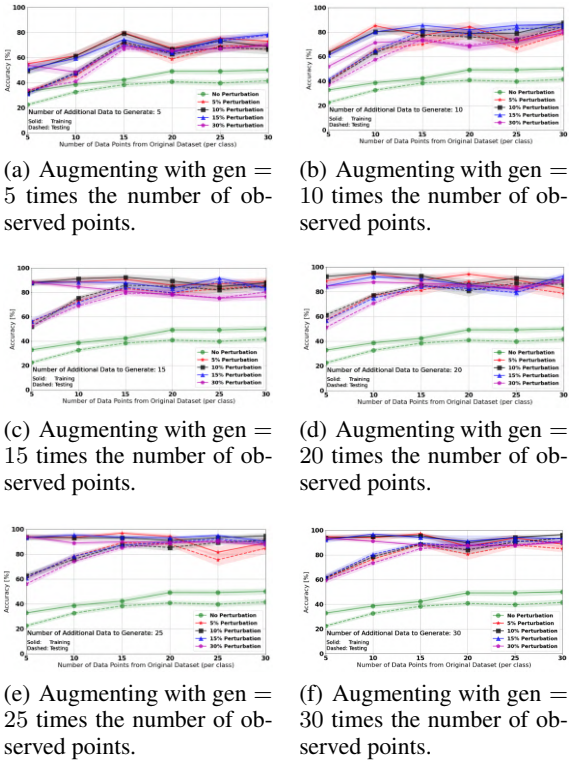


Figure 18: A diagram showcasing the nested workflow of experiments designed to validate the effectiveness of *StiefelGen* for a numerical time series classification problem. The primary objective is to address the necessity for data augmentation in this context, ultimately enhancing the learned capacity of the LSTM model for improved performance.

Concerning the architecture of the complete end-to-end LSTM model, the initial layer features a bidirectional LSTM with 100 hidden units in total (Graves and Schmidhuber 2005). This was followed by a fully connected layer with 9 output units (matching the number of classes) and a subsequent softmax layer. The chosen loss function was cross-entropy, and a mini-batch size of 64 was selected without loss of generality, and the training time spanned 50 epochs in total. These hyper parameters were selected as empirical evidence suggests they can lead to an LSTM model with ample learning capacity to achieve high accuracy scores on the given dataset (with testing accuracy scores above that of 97%) (Mathworks 2023).

However, since the primary aim is to illustrate that the proposed data augmentation method through *StiefelGen* can significantly boost training and testing accuracy scores, the learning rate for the ADAM optimizer in *PyTorch* (Kingma and Ba 2014; Paszke et al. 2019) was selected to be 0.001. This choice ensures that the *PyTorch* LSTM model does not attain its full learning capacity so that the impact of data augmentation on under trained models could also be studied. Although a comparable learning rate was used in (Mathworks 2023) in its MATLAB® implementation, it is essential to note that the default hyper parameter inputs of the ADAM function differ significantly across the *PyTorch* and MATLAB® implementations, which leads to vastly different end states of the 50 epoch learning process across both implementations. Consequently, equivalent ADAM learning rates do not necessarily imply reaching an equivalent capacity LSTM model within the specified 50 epochs, which in the case of *PyTorch* for this experiment arrives at an under capacity LSTM model given the chosen random seed. The empirical results of performing this nested set of experiments is show in Figure 19.



(a) Augmenting with gen $=$ 5 times the number of observed points.

(b) Augmenting with gen $=$ 10 times the number of observed points.

(c) Augmenting with gen $=$ 15 times the number of observed points.

(d) Augmenting with gen $=$ 20 times the number of observed points.

(e) Augmenting with gen $=$ 25 times the number of observed points.

(f) Augmenting with gen $=$ 30 times the number of observed points.

Figure 19: Plots of the empirical results as the number of augmented datasets (gen) increases. The x-axis of each plot represents, n, the number of observed data taken from the training data set. Different levels of perturbation are shown on each plot, as well as training and testing accuracy results.

Firstly, it should be noted that in Figure 19 each subplot's data augmentation level (gen) does not apply to the green curves representing the original dataset. For instance, at $n = 10$ on the x-axis in Figure 19(a), the green curve will only see $n = 10$ observations per class (resulting in $10 \times 9 = 90$ data points in total) for LSTM model training. This will be the same for every subplot in Figure 19, as this curve represents the non-augmented, vanilla, reference model control, and is plotted across all plots for the sake of comparison. Conversely, each *perturbed* curve in Figure 19(a) will involve the generation of $n\text{gen} = 105 = 50$ augmented data points *per class* for each perturbation level in Figure 19(a), resulting in $10 \times 5 \times 9 = 450$ data points in total for LSTM model training. Note that we do not have $450 + 15 = 465$ data points because the original data set was discarded after augmentation (for simplicity).

Looking at Figure 19 holistically, it's evident that by applying the *StiefelGen* augmentation procedure, one consistently outperforms the training and testing scores of the original dataset across all perturbation levels, for every augmentation generation level. Also, increasing the number of augmented datasets (gen) for a fixed n tends to lead to a gradual rise in accuracy scores for both training and testing sets.

However, this effect seems to plateau around $gen = 15$, as the augmentation cases (gen $= \{15, 20, 25, 30\}$) tend to display similar score magnitudes and overall shapes (from Figures 19(c) to 19(f))), in that they all seemingly share an elbow-like effect at $n = 15$. Beyond $n = 15$, testing scores appear to approximately level out with a steady, albeit slightly noisy increase in the test score as n grows. This behavior is shared by both the original non-augmented dataset and the augmented datasets, suggesting that from $n = 15$ onwards, the provided dataset perhaps offers limited additional variety for modeling out-of-sample scenarios which may be present in the testing set. Despite this, the *StiefelGen* methodology, with its more rapid gain in percentage testing accuracy per unit n, from $n = 5$ up to the elbow point of $n = 15$, suggests it is performing an effective generation of out-of-sample data even in the event of working with a severe sub sample of the original data set, which suggets that *StiefelGen* is augmenting the data set with effective (and or realistic) signal data.

Full Table of Values: LSTM Experiment

This subsection provides the quantitative summary of all the statistical values used to generate Figure 19. The bold values in each row indicate the experimental run which received the highest testing score for that row (which is often strongly correlated with the highest training scores for that row as well). It is clear that every single experimental augmentation setting outperformed the corresponding non-augmented experiment by a very large margin. The largest improvement was observed in the experiment which used a perturbation level of 10% and the first $N = 30$ data elements per class for the augmentation procedure (which coincidentally is all the available data), and then perturbing this 30 times over per class. This resulted in a mean training and test score of 96.35% and 93.55% respectively. This is more than double the test score of the vanilla, non-augmented model which had its best result occur similarly at $N = 30$, with the training score being 50.02%, and its testing score of 41.53%.

Table 3: The result of using the first N elements from the each class of the data set, where $N = [5, 10, 15, 20, 25, 30]$. This is the base reference set of training and testing accuracies which have not received any data augmentation.

Pert. Level [%]	**Accuracy [%]**	$5(\times1)$	$10(\times1)$	$15(\times1)$	$20(\times1)$	$25(\times1)$	$30(\times1)$
0	**Train**	32.78±1.45	38.67±1.83	42.26±2.43	49.17±1.95	49.13±2.40	**50.02±2.22**
	Test	22.42±1.33	32.64±1.39	38.39±1.87	40.77±1.86	39.76±2.03	**41.53±2.33**

Table 4: Results of applying *StiefelGen* with various perturbation levels ($[5\%, 10\%, 15\%, 30\%]$, across various amounts of data taken from the dataset for generation $[5, 10, 15, 20, 25, 30]$ for augmentation, assuming 5 times the amount of data is generated.

Pert. Level [%]	**Accuracy [%]**	$5(\times5)$	$10(\times5)$	$15(\times5)$	$20(\times5)$	$25(\times5)$	$30(\times5)$
5	**Train**	55.04±2.14	61.48±3.14	**79.82±2.26**	67.58±3.94	75.45±2.29	73.09±4.55
	Test	33.38±1.65	45.54±2.37	**71.80±2.14**	58.82±3.72	69.07±2.70	69.39±3.77
10	**Train**	49.58±2.98	61.31±3.76	**79.33±2.57**	66.78±3.98	73.27±3.15	68.78±3.70
	Test	31.73±1.75	47.99±2.90	**72.51±2.32**	62.96±4.08	67.99±3.01	66.66±3.82
15	**Train**	50.93±3.45	59.46±2.19	74.36±2.86	64.49±3.03	74.58±2.83	**78.61±2.30**
	Test	31.19±1.74	47.46±1.81	70.30±2.72	64.15±3.27	73.14±2.58	**78.07±2.29**
30	**Train**	53.91±3.17	48.48±3.13	68.96±1.44	65.31±1.97	67.22±2.65	**69.60±1.92**
	Test	34.15±1.48	41.05±2.86	66.80±1.67	65.68±2.61	67.18±3.14	**70.70±1.90**

Table 5: Results of applying *StiefelGen* with various perturbation levels ($[5\%, 10\%, 15\%, 30\%]$, across various amounts of data taken from the dataset for generation $[5, 10, 15, 20, 25, 30]$ for augmentation, assuming 10 times the amount of data is generated.

Pert. Level [%]	**Accuracy [%]**	$5(\times10)$	$10(\times10)$	$15(\times10)$	$20(\times10)$	$25(\times10)$	$30(\times10)$
5	**Train**	63.39±3.66	85.22±2.14	77.24±5.26	**84.41±4.18**	72.50±3.88	82.39±4.71
	Test	39.80±2.20	65.66±2.80	70.16±4.79	**79.14±3.64**	66.97±4.13	78.89±4.66
10	**Train**	63.19±2.46	80.43±3.26	81.37±1.72	79.02±2.82	79.12±4.58	**87.77±1.86**
	Test	39.69±2.09	63.31±3.27	77.66±1.41	76.01±2.58	73.84±4.82	**84.54±2.04**
15	**Train**	61.12±2.20	80.14±2.97	85.63±2.05	81.93±2.88	85.50±3.34	**86.40±2.43**
	Test	41.35±1.82	64.97±2.89	81.23±1.92	78.77±3.14	82.81±3.50	**83.86±2.80**
30	**Train**	52.02±2.37	71.56±3.06	73.16±2.10	68.24±3.56	72.23±3.58	**79.53±2.06**
	Test	36.81±1.82	57.57±2.55	74.11±2.26	69.27±4.08	74.77±3.86	**81.01±2.33**