International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

A UNIFIED MATHEMATICAL MODEL BRIDGING GENERATIVE ADVERSARIAL NETWORKS AND CENTURY-OLD STOCHASTIC THEORIES

Mr. Md Afzal[*1], Dr. R Udaya Kumar[2], Dr. Mohammed Abdul Bari[3], Dr. Mohd Ahmed[4]

[1] Research scholar, Kalinga University, Raipur, Chhattisgarh

mohammedafzal.ku@gmail.com

[2] Assistant Professor, Department of Computer Science and Engineering, Kalinga University, Raipur, Chhattisgarh

rsukumar2007@gmail.com, directoripr@kalingauniversity.ac.in

[3] Professor & Dean Academics, Department of Computer Science and Engineering, ISL Engineering College, Hyderabad
abdulbarimohammed11@gmail.com

[4] Assistant Professor, Department of Mathematics, Spoorthy Engineering College, Hyderabad
ahmed.nitw@gmail.com

[*1]Corresponding Author Mail ID: mohammedafzal.ku@gmail.com

Abstract

The utilization of generative adversarial networks (GANs) for the synthesizing of time-series data has gained increasing prominence in recent years, with applications extending across diverse domains such as financial modelling, music composition, and textual content analysis. Despite these advances, systematic evaluations contrasting GANs with alternative artificial intelligence (AI) methodologies or established mathematical frameworks remain relatively scarce. In this work, we undertake a comparative assessment of GANs against a canonical stochastic model, the Markov chain. Employing metrics derived from one- and two-point statistical analyses, we examine the capacity of each approach to replicate salient properties of time-series data. Our results indicate that, consistent with broader observations of AI-based generative models, GANs exhibit limitations in reproducing rare events and capturing cross-feature dependencies, thereby constraining their ability to generate fully faithful synthetic sequences. Although GANs demonstrate moderate proficiency in replicating auto-correlation structures, their performance remains markedly inferior to that of simple Markov chains. We further provide a qualitative discussion elucidating the structural reasons underlying these limitations, thereby contributing to a deeper understanding of the challenges inherent in AI-driven generative modelling of time-series data.

1. Introduction

The enhancement of computer capacity and the increasing accessibility of extensive datasets have elevated the significance of non-parametric models in time-series research. The allure of these models resides in their extensive applicability: they circumvent the need to reduce complex systems to a limited set of fundamental descriptors, a process that often requires simplifying assumptions which may compromise accuracy.

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Within this context, artificial neural network (ANN)-based algorithms have been widely adopted for diverse predictive tasks, including wind speed forecasting [1], wind power estimation [2], air quality assessment [3], financial asset price evolution [4], epidemic infection rates [5], and patient arrivals in hospital emergency departments [6]. Beyond prediction, these methods have proven valuable for data augmentation in scenarios where original datasets are insufficient, and in sensitive domains such as medicine, where they can generate anonymized data suitable for sharing without stringent ethical constraints.

Generative adversarial networks (GANs) [7] have become one of the most popular non-parametric techniques. GANs have recently been expanded to time-series replication due to their ability in creating extremely realistic visuals [8]. In theory, GANs are made up of two connected ANNs playing a zero-sum game: the discriminator tries to discern between generated and real samples, while the generator creates fake data to trick it. Numerous fields have used this adversarial framework, such as natural language processing [10], music production [11,12], pedestrian trajectory prediction [13], financial asset forecasting [14], and biomedical signal modelling (electrocardiogram, electroencephalogram, electromyography, and photoplethysmography) [9].

In biomedical contexts, eye-tracking data represent a particularly compelling test case for GAN-based modelling. From personality features [15] and drug use patterns [16,17] to diagnostic markers for attention-deficit hyperactivity disorder [18,19] and autism spectrum disorder [20], such data capture a variety of behavioural and health-related signs. Moreover, eye-tracking signals exhibit high kurtosis, reflecting the prevalence of extreme events such as rapid gaze shifts between areas of interest. These heavy-tailed distributions pose significant challenges for ANN-based generative methods, which often struggle to reproduce rare events faithfully.

In order to overcome these problems, this work assesses how well GANs reproduce random trajectories by comparing them to traditional stochastic models, particularly Markov chains [21]. Like AI-based approaches, Markov models can replicate observed data without requiring explicit knowledge of the underlying process. However, they offer distinct advantages: the ability to provide continuous-time descriptions, to identify whether a process is stationary or time-continuous [22], and to yield interpretable parameters that offer direct insights into system dynamics—features generally absent in ANN-based methods.

The structure of the manuscript is as follows. The technical implementation of the Markov chains and GAN architectures used is described in Section 2. The datasets employed, which include both synthetic sequences and empirical eye-tracking data, are described in Section 3. GANs and Markov chains are compared across various datasets in Section 4, and the results and their wider implications are discussed in Section 5.

2. Procedures and Algorithms

2.1. Architecture of Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) represent a class of AI algorithms structured as shown in Fig. 1. The generator starts the process by creating a synthetic time-series from a

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

random noise vector z0. This output is then reviewed by the discriminator, which receives either genuine data x or created data z, and attempts to categorize it as real or bogus. In a framework for min-max optimization, both networks are trained concurrently [23]. While the generator is educated to lower this accuracy by generating progressively realistic outputs, the discriminator is tuned to maximize its accuracy in differentiating authentic from created samples. Through this adversarial interplay, the generator progressively learns to replicate the statistical characteristics of the true data. As the discriminator improves its ability to detect fakes, the generator must correspondingly enhance its outputs to continue deceiving it.

In mathematical terms, the behaviour of a GAN can be expressed using the following objective function.

$$L\big(G(z), D(x)\big) = \mathbb{E}_{x\sim\rho x}[\log D(x)] + \mathbb{E}_{z\sim\rho z}[\log(1 - D(G(z)))] \qquad (1)$$

The discriminator $D(x) \in [0, 1]$ assigns to each input x the probability that it corresponds to real data. Accordingly, the expectation $\mathbb{E}_{x\sim\rho x}\ [log\ D(x)]$ quantifies the average accuracy of the discriminator when evaluating genuine data samples drawn from the distribution ρ_x. In parallel, the generator G is defined as a mapping from an input noise vector z to a synthetic series $G(z)$ (see Fig. 1). For such generated samples, the term $1 - D(G(z))$ represents the probability that the discriminator correctly identifies them as artificial. The underlying probability distributions are denoted as ρ_x for the real data series, ρ_z for the input noise, and ρ_G for the generator's output.

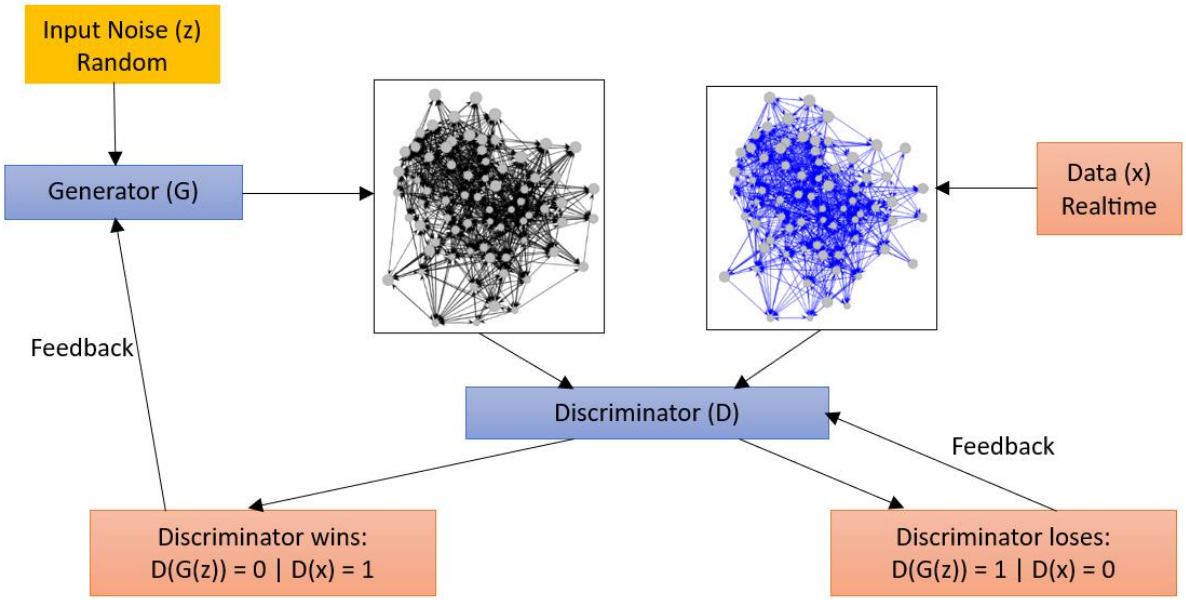

Fig. 1. A GAN framework, the generator aims to replicate real data as convincingly as possible, whereas the discriminator's role is to identify whether a given sample originates from the true dataset or has been generated synthetically.

The generator's objective is to align the probability distribution ρ_G of the synthetic series $G(z)$ as closely as possible with the distribution ρ_x of the real data series x, thereby reducing both terms on the right-hand side of the objective function. In contrast, the discriminator seeks to maximize these expectations—by increasing $D(x)$ for real samples and decreasing $D(G(z))$ for generated ones. Consequently, with the objective function defined in Eq. (1), the training of a

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

GAN is formulated as a min–max optimization problem, where the generator minimizes and the discriminator maximizes the same function.

$$\min_{G} \max_{D} L(G(z), D(x)) \tag{2}$$

Within the broader GAN framework, multiple architectural variations can be constructed by employing different types of artificial neural networks as generators and discriminators. In this study, we examine a set of representative architectures, summarized in Table 1 along with the specific configurations of their generator and discriminator components. Comprehensive training procedures for these GAN models are provided in Appendix B.

2.1.1. Recurrent conditional GAN

The Recurrent Conditional GAN (RCGAN) [24] represents one of the earliest GAN-based approaches tailored for time-series data. It was developed to replicate multi-valued medical time-series, specifically modelling the progression of patients' health states in emergency care. The dual motivation behind this design was to enable outcome prediction while simultaneously generating realistic synthetic data free from privacy constraints. As its name implies, RCGAN employs recurrent neural networks for both the generator and the discriminator, with Long Short-Term Memory (LSTM) cells forming the core of each architecture [25].

2.1.2. TimeGAN

TimeGAN, introduced in 2019, was designed to surpass several state-of-the-art GAN models [26], with applications demonstrated on stock market and energy consumption datasets. The model integrates a classical GAN framework with an autoencoder—an artificial neural network composed of two components: an encoder that maps a time-series into a latent representation, and a decoder that reconstructs the series from this representation [27].

The full TimeGAN architecture comprises four networks. Within the GAN component, a recurrent neural network (RNN) serves as the generator, while a bidirectional LSTM functions as the discriminator. Unlike conventional GANs, the discriminator does not directly receive raw data; instead, it operates on encoded vectors produced by the autoencoder.

Training is guided by three distinct loss functions:

- **Reconstruction loss**, derived from the decoder's output, which updates the encoder and decoder parameters.
- **Unsupervised loss**, based on the discriminator's predictions, analogous to the min–max objective of the original GAN (Eq. 1), and used to update both generator and discriminator.
- **Supervised loss**, computed from the generator's output and the encoder's representation of real data, which informs updates to the generator and encoder.

It is worth noting that some implementations of TimeGAN omit the autoencoder component, relying solely on the GAN structure [28]. In this work, we adopt that streamlined implementation.

2.1.3. SigCWGAN and RCWGAN

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The Conditional Sign-Wasserstein GAN (SigCWGAN) [28] was developed to model temporal dependencies in multi-dimensional time-series data while accurately capturing the tail behaviour of the underlying distribution. To achieve this, the authors introduced a novel metric, the Signature Wasserstein-1 (C-Sig-W1), which functions as the discriminator. This metric is designed to be both more robust and computationally less demanding than conventional ANN-based discriminators. For the mathematical formulation of the discriminator, readers are referred to Ref. [28].

In terms of architecture, the generator is implemented as a three-layer feed-forward neural network with residual connections and parametric ReLU activation functions. This design, termed the Autoregressive Feed-Forward Neural Network (AR-FNN), is specifically intended to capture autoregressive processes and thereby preserve temporal dependencies in the generated series. Further implementation details are provided in the appendix of the original paper [28].

The authors also proposed a variant in which both the generator and discriminator are recurrent neural networks incorporating AR-FNN cells. This extension is referred to as the Recurrent Conditional Wasserstein GAN (RCWGAN), and it is the architecture considered in this study.

Finally, the original PyTorch implementations of SigCWGAN, RCWGAN, and other GAN models discussed can be accessed via the authors' GitHub repository [29].

2.2. Reproducing time-series - Markov-chain models

Markov models, first introduced in 1906 [30], were developed to capture conditional probabilities and thereby describe the temporal evolution of stochastic processes. Their earliest application involved estimating the likelihood of encountering a vowel in a text, given knowledge of the preceding letter [21]. Formally, a time-series X_t is said to follow a Markov process if it satisfies the Markov property: the conditional distribution of the future state depends solely on the present state, and not on the sequence of past states.

$$\begin{aligned}\Pr\left(X_{t=j} = \hat{x}_j \middle| X_{t=j-1} = \hat{x}_{j-1}, \dots, X_{t=0} = \hat{x}_0\right) \\ = \Pr\left(X_{t=j} = \hat{x}_j | X_{t=j-1} = \hat{x}_{j-1}\right)\end{aligned} \tag{3}$$

Capital letters signify stochastic variables at various time steps for all positive integers j, whereas lowercase letters indicate their values. At regular intervals Δt, successive observations are made.

According to the Markov condition, forecasts regarding a time series X_t*'s* future are based just on its current state and not on its historical trajectory. For most natural systems, this characteristic—often called "memorylessness"—is a simplification. However, it is quite helpful: the temporal history of the process may be fully described by calculating the conditional probability $\Pr(X_{t=j} = \hat{x}_j \mid X_{t=j-1} = \hat{x}_{j-1})$. To put it another way, two-point statistics capture all the data needed to characterize the system. Appendix C contains more information on creating Markov processes in this way.

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3. Data and evaluation metrics

3.1. Synthetic data

This study's initial dataset is a Vector Auto-Regressive *(VAR)* process that was created artificially. A *VAR(p)* model in the one-dimensional case asserts that, subject to a tiny random noise term, the current observable at time *t* is determined by its values over the previous *p* time steps. The observable in our application is the eye-gaze velocity, which is expressed as spatial increments in both the *x* and *y* axes. The *VAR(p)* model can be written as follows for these increments (ΔX):

$$\Delta X_t = \sum_{n=1}^{p} \phi_n \Delta X_{t-n} + \xi_t(\sigma) \tag{4}$$

In this case, a normally distributed random variable with zero mean and standard deviation σ is denoted by ξ. Future increments in a *VAR(p)* process are decided by a linear mix of Gaussian fluctuations and the previous *p* increments. Units with constant intervals Δt are used to index time. In this investigation, we start with the initial condition $\Delta X_0 = 0$ and set $\Delta t = 1$ to generate about 85,000 points, which matches the size of the empirical eye-tracking dataset.

In the analysis that follows, we concentrate on a *VAR(1)* process, in which every increment is dependent upon both the previous increment and a random term. Extending this to two dimensions, we consider increments in both *X* and *Y*, with σ_{XY} capturing the correlation between ΔX and ΔY. The system of equations that follows defines the final process:

$$\Delta X_t = \phi_X \Delta X_{t-1} + \xi_t^{(X)}(\sigma_X, \sigma_{XY}), \Delta Y_t = \phi_Y \Delta Y_{t-1} + \xi_t^{(X)}(\sigma_Y, \sigma_{YX}) \tag{5}$$

In this case, the stochastic fluctuations in the *X-dimension* with zero mean, standard deviation σ_X, and correlation σ_{XY} with the fluctuations in the *Y-dimension* are represented by $\xi_t^{(X)}(\sigma_X, \sigma_{XY})$. To keep things simple, we consider an isotropic process, where $\sigma_X = \sigma_Y \equiv \sigma$ and $\phi_X = \phi_Y \equiv \phi$.

One of the most basic synthetic time-series models with temporal correlation is the *VAR(1)* process. It is a suitable benchmark for assessing the accuracy of our implementation because it satisfies the Markov property by construction. Additionally, for$\varphi>0$, it offers an appropriate test case to investigate the well-known shortcomings of GAN algorithms and neural networks in modelling distributional tails.

A two-dimensional *VAR(1)* process with parameters$\sigma= 1,\varphi= 0.8$, and $\sigma_{XY} = 0.8$ is shown in Figure 2 (top). The equivalent scatter plot of increments in both *X* and *Y* directions is displayed in the inset. As can be seen, there is a positive correlation between ΔX and ΔY, and the values tend to line up along the main diagonal. Additionally, extreme values in ΔX (corr. ΔY) are frequently followed by extreme values in the other coordinate, indicating the positive autocorrelation caused by$\varphi= 0.8$. Three different *VAR* process types, corresponding to orders *p* = *1, 2, and 3*, are taken into consideration in this study.

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

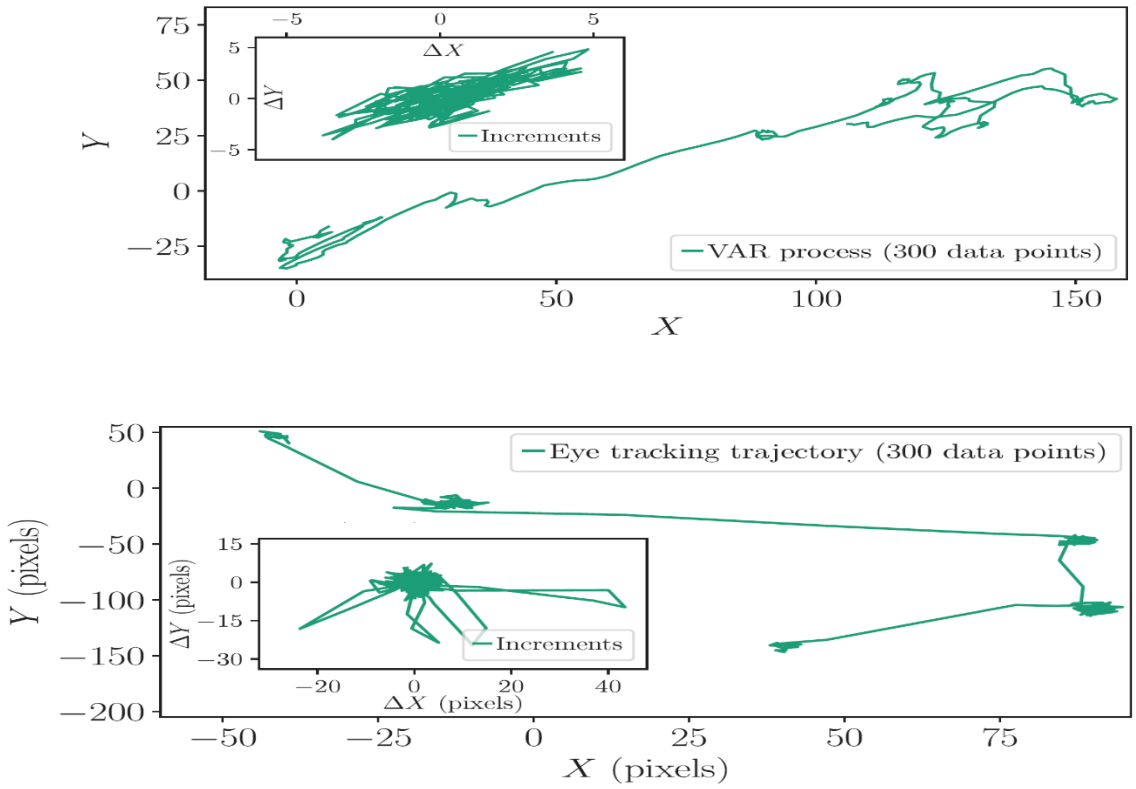

Figure 2. Top: A two-dimensional *VAR(1)* process's trajectory is depicted in the left panel, and its corresponding increments are shown in the right. Positive autocorrelation *(φ = 0.8)* and positive cross-feature correlation *(σ_{XY} = 0.8)* were used to create this artificial time series. Consequently, the values fill the first and third quadrants and are primarily oriented along the diagonal. Bottom: An empirically estimated gaze trajectory captured by a contemporary eye-tracker is shown in the left panel, and its increments are shown in the right. Gaze positions in these kinds of data usually cluster inside a limited spatial area, with quick relocation movements that correspond to changes in visual attention.

3.2. Eye-tracking data

The Eye-link Duo, a cutting-edge instrument with a maximum sampling frequency of 2000 Hz and a precision of 0.1° of viewing angle, was used at Oslo Metropolitan University to gather the eye-tracking dataset. For the present analysis, the data was downsampled to 200 Hz, and blinks were removed, resulting in approximately 85,000 data points. The gaze coordinates *(X,Y)* are expressed in screen pixels. Participants were instructed to locate pre-selected targets within images taken from Where's Wally?. Eight images were presented, each for a duration of two minutes. Since two minutes is insufficient to identify all targets in such complex images, this design ensured sustained engagement throughout the experiment. Under Reference Number 176347, data collection complied with all ethical guidelines authorized by the Norwegian Centre for Research Data (NSD).

Figure 2 (bottom) illustrates a representative gaze trajectory, with its increments shown in the inset. Consistent with established literature, the trajectory alternates between fixations (periods of relative stability around a point) and saccades (rapid relocations). Notably, the velocity time series exhibits extreme events, with an excess kurtosis of $\kappa \sim 50$. Such heavy-tailed behaviour

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

is particularly valuable for evaluating the performance of artificial neural network (ANN)-based models, as these methods are known to struggle with capturing extremes. Some models explicitly claim improved ability to replicate distributional tails. By contrast, real-world datasets typically exhibit lower kurtosis values $5 < \kappa < 15$, containing far fewer extreme events.

Two forms of non-Markovian behaviour are evident in gaze trajectories:

- Inhibition of return (IoR): a mechanism whereby the gaze avoids revisiting a recently inspected area for a certain period.
- Screen confinement: the restriction of gaze trajectories to the screen boundaries, which induces long-range dependencies in velocity sequences.

To characterize gaze dynamics, three quantities are central:

- The velocity magnitude (‖v‖).
- The velocity angle (θ) measured relative to the horizontal axis.
- The turning angle (ϕ) defined between consecutive velocity vectors.

Figure 3 shows these quantities. The capacity of the model to reproduce the distributions of these measures and the temporal evolution of the velocity magnitude ‖v‖ will be used to evaluate its performance in the analysis that follows.

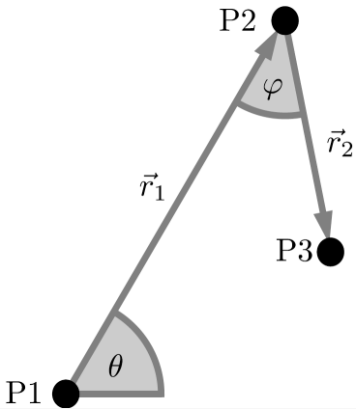

Fig. 3. Illustrated Key parameters utilized to describe gaze trajectories. These comprise the amount of the glance velocity, the turning angle *(ϕ)* between two successive gaze relocations, and the gaze direction *(θ)* with respect to the horizontal axis. The displacement vectors $\vec{r}_1$ and $\vec{r}_2$ occur over equal time intervals *Δt* in the case. As a result, the velocity magnitude is defined as $\|\vec{v}_1\| = \|\vec{r}_1\|/\Delta t$.

3.3. Evaluation metrics

Most of the current research focuses on image-based applications, and the selection of criteria for assessing the performance of generative adversarial networks (GANs) is still up for debate [32–34].

The authors of Ref. [24] use maximum mean discrepancy (MMD) to evaluate the degree of resemblance between artificially generated samples and actual data. The average fluctuations within each dataset are compared to the differences between them using this metric.

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Additionally, they suggest the train-on-synthetic-test-on-real (TSTR) method, which calls for tagged synthetic data and a supervised classifier that can correctly learn from the actual dataset. Using this method, the classifier is evaluated on actual data after being trained on synthetic data. Successful categorization suggests that models with real prediction potential can be constructed using synthetic, completely anonymised data.

In Ref. [26], the authors employ t-distributed stochastic neighbour embedding (t-SNE) to visualize high-dimensional data in two dimensions, thereby enabling inspection of distributional properties. To evaluate whether correlations across dimensions are preserved, they apply principal component analysis (PCA). Although auto-correlation profiles are not directly examined, conditional probability distributions are compared as a proxy for assessing time dependencies and predictive capacity.

Lastly, L^1-distance applied to probability density functions and auto-correlation functions is used in Ref. [28] to assess GAN performance. This metric is also expanded to quantify variations in feature correlations when dealing with multidimensional data. The TSTR scheme is also used by the authors to verify the prediction value of artificial datasets.

In general, two key aspects are of interest when evaluating generated data:

1. Does the synthetic data's distribution resemble the original dataset's?
2. How well the temporal relationships are replicated, including any possible long-range correlations.

We look at the distributions of the angular measurements, θ and ϕ, as well as the velocity magnitude $\|v\|$ to determine how well a method reproduces the distributional characteristics of the original data (see Fig. 3).

We use the Jensen–Shannon (JS) divergence, a symmetrized version of the Kullback–Leibler (KL) divergence, to measure distributional similarity. The JS divergence is especially useful for comparing probability distributions because, in contrast to the KL divergence, it satisfies the characteristics of a true distance metric. The definition of the KL divergence itself is:

$$D_{KL}(\rho_{emp}||\rho_{syn}) = \int \rho_{emp} \log\left(\frac{\rho_{emp}}{\rho_{syn}}\right). \tag{6}$$

Hence, divergence of JS is defined as

$$D_{JS}(\rho_{emp}||\rho_{syn}) = \frac{1}{2}\left(D_{KL}(\rho_{emp}||\bar{\rho}) + D_{KL}(\rho_{syn}||\bar{\rho})\right) \tag{7}$$

where $\overline{\rho} = \frac{1}{2}(\rho_{emp} + \rho_{syn})$.

The Kullback–Leibler (KL) divergence is one of the most widely used measures for quantifying the similarity between probability distributions, as minimizing KL divergence corresponds to

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

maximum likelihood estimation. The Jensen–Shannon (JS) divergence preserves this desirable property while also introducing symmetry, namely: $D_{JS}(\rho_{emp}||\rho_{syn}) = D_{JS}(\rho_{syn}||\rho_{emp})$.

This symmetry is particularly intuitive in the context of generative adversarial networks: the generator seeks to align synthetic distributions with empirical ones, whereas the discriminator attempts to distinguish between real and generated distributions.

Table 2

Moments of the probability distribution function of $|v|$ are shown for each dataset alongside the corresponding Markov and GAN-generated data. The results indicate that the Markov model performs relatively well in reproducing the distribution of $|v|$. In contrast, GAN-based models consistently underestimate the distributional moments. A key limitation of GANs lies in their inability to capture the heavy tails of the distribution, reflected in the reduced kurtosis values.

An exception arises with TimeGAN applied to the *VAR(2)* dataset, which achieves a closer match in kurtosis. However, this comes at the cost of a substantial underestimation of the standard deviation. The challenge of replicating the moments of the $|v|$-distribution becomes even more pronounced in the case of empirical gaze trajectories, whose complexity far exceeds that of synthetic *VAR* processes.

		Mean	Standard deviation	Skewness	Excess of kurtosis
VAR(1)	Data	2.0	1.27	1.1	1.39
	Markov	2.00 ± 10^{-2}	1.30 ± 10^{-2}	$1.10 \pm 3 \cdot 10^{-2}$	1.4 ± 10^{-1}
	RC	$1.00 \pm 4 \cdot 10^{-2}$	$6.8 \cdot 10^{-1} \pm 5 \cdot 10^{-2}$	$1.2 \pm 2 \cdot 10^{-1}$	2 ± 1
	Time	$1.10 \pm 2 \cdot 10^{-2}$	$7.1 \cdot 10^{-1} \pm 1 \cdot 10^{-2}$	$7.8 \cdot 10^{-1} \pm 6 \cdot 10^{-2}$	$3 \cdot 10^{-1} \pm 2 \cdot 10^{-1}$
	SIGCW	$1.20 \pm 2 \cdot 10^{-2}$	$7.2 \cdot 10^{-1} \pm 2 \cdot 10^{-2}$	$1.00 \pm 7 \cdot 10^{-2}$	$1.0 \pm 3 \cdot 10^{-1}$
	RCW	$1.40 \pm 3 \cdot 10^{-2}$	$8.7 \cdot 10^{-1} \pm 2 \cdot 10^{-2}$	$9.3 \cdot 10^{-1} \pm 7 \cdot 10^{-2}$	$8 \cdot 10^{-1} \pm 3 \cdot 10^{-1}$
VAR(2)	Data	1.5	1.0	1.0	1.2
	Markov	$1.500 \pm 4 \cdot 10^{-3}$	$9.80 \cdot 10^{-1} \pm 3 \cdot 10^{-3}$	1.10 ± 10^{-2}	$1.20 \pm 5 \cdot 10^{-2}$
	RC	$8.9 \cdot 10^{-1} \pm 2 \cdot 10^{-2}$	$5.1 \cdot 10^{-1} \pm 1 \cdot 10^{-2}$	$7.2 \cdot 10^{-1} \pm 6 \cdot 10^{-2}$	$6 \cdot 10^{-1} \pm 2 \cdot 10^{-1}$
	Time	$2.80 \pm 4 \cdot 10^{-2}$	$1.8 \cdot 10^{-1} \pm 10^{-2}$	5.0 ± 10^{-1}	$3.00 \pm 9 \cdot 10^{-2}$
	SIGCW	$1.100 \pm 7 \cdot 10^{-3}$	$8.40 \cdot 10^{-1} \pm 6 \cdot 10^{-3}$	$1.00 \pm 2 \cdot 10^{-2}$	$8.7 \cdot 10^{-1} \pm 7 \cdot 10^{-2}$
	RCW	(Diverges)	(Diverges)	(Diverges)	(Diverges)
VAR(3)	Data	2.1	1.4	1.1	1.2
	Markov	$2.200 \pm 9 \cdot 10^{-3}$	$1.400 \pm 8 \cdot 10^{-3}$	$1.00 \pm 1 \cdot 10^{-2}$	$1.20 \pm 7 \cdot 10^{-2}$
	RC	1.80 ± 10^{-2}	$4.4o \cdot 10^{-1} \pm 9 \cdot 10^{-3}$	$-5 \cdot 10^{-1} \pm 5$	$5 \cdot 10^{-1} \pm 10^{-1}$
	Time	(Diverges)	(Diverges)	(Diverges)	(Diverges)
	SIGCW	1.00 ± 10^{-2}	$7.10 \cdot 10^{-1} \pm 8 \cdot 10^{-3}$	$1.10 \pm 3 \cdot 10^{-2}$	1 ± 10^{-1}
	RCW	(Diverges)	(Diverges)	(Diverges)	(Diverges)
Eye-Gaze	Data	3.2	6.2	5.8	$4.1 \cdot 10^{+1}$
	Markov	3.3 ± 10^{-1}	$6.2 \pm 3 \cdot 10^{-1}$	$5.8 \pm 2 \cdot 10^{-1}$	$4.1 \cdot 10^{+1} \pm 3$
	RC	$4.20 \cdot 10^{-1} \pm 5 \cdot 10^{-3}$	$2.90 \cdot 10^{-1} \pm 5 \cdot 10^{-3}$	$1.50 \pm 8 \cdot 10^{-2}$	$2.9 \pm 5 \cdot 10^{-1}$
	Time	$3.70 \cdot 10^{-1} \pm 4 \cdot 10^{-3}$	$2.20 \cdot 10^{-1} \pm 3 \cdot 10^{-3}$	$1.20 \pm 8 \cdot 10^{-2}$	$2.4 \pm 6 \cdot 10^{-1}$
	SIGCW	$1.200 \pm 7 \cdot 10^{-3}$	$7.70 \cdot 10^{-1} \pm 7 \cdot 10^{-3}$	$1.30 \pm 4 \cdot 10^{-2}$	$2.6 \pm 3 \cdot 10^{-1}$
	RCW	$4.50 \cdot 10^{-1} \pm 9 \cdot 10^{-3}$	$3.7 \cdot 10^{-1} \pm 10^{-2}$	$2.0 \pm 2 \cdot 10^{-1}$	6 ± 1

To assess how effectively the time-dependency of the data is reproduced, we examine the autocorrelation of the velocity magnitude ‖v‖. Autocorrelation is computed as the Pearson correlation coefficient for each spatial coordinate. For the X-coordinate, this is expressed as:

$$\gamma X(t_{lag}) = \frac{E[(X(t) - \bar{X})(X(t - t_{lag}) - \bar{X})]}{E[(X(t) - \bar{X})(X(t) - \bar{X})]} \tag{8}$$

Here, X denotes the mean of the spatial coordinate, t_{lag} represents the time lag at which the autocorrelation is computed, and $E[x]$ indicates the expected value of the stochastic variable x. It is important to note that empirical gaze trajectories are not stationary stochastic processes. However, both the AI-based models and the Markov model generate time-homogeneous

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

trajectories. In this context, γ_X (and analogously γ_Y) serves as a measure of the model's ability to reproduce the average dynamical behaviour of the time series.

To evaluate how accurately an algorithm reproduces the autocorrelation function of each of the three quantities, we employ the L^2-distance as a measure of similarity. The L^2-distance between the empirical function $f_{emp}(x)$ and the synthetic function $f_{syn}(x)$ is defined as:

$$d(f_{emp}, f_{syn}) = \left(\int_{-\infty}^{\infty} (f_{emp}(x) - f_{syn}(x))^2 \, dx \right)^{\frac{1}{2}}. \tag{9}$$

4. Comparative analysis

The results of this study are presented in Figure 4, with additional simulations provided in Appendix A (Figures 6 and 7), and in Figure 5, which focuses on the reproduction of eye-gaze trajectories. Table 2 reports the expected values and associated uncertainties for the first four moments of the $|v|$-distribution—namely the mean, variance, skewness, and kurtosis—original datasets (synthetic, *VAR* processes, and empirical eye-gaze trajectories) and their matching Markov and GAN-generated equivalents in each scenario.

The closeness between simulated and empirical data is summarized in Table 3. We give the numerical values of the Jensen–Shannon divergence (cf. Eq. (7)) between empirical and synthetic distributions for each of the three parameters defining eye-gaze trajectories (cf. Fig. 3). Additionally, the L^2-distance is used to quantify the similarity of the velocity magnitude autocorrelation function (see Eq. (9)).

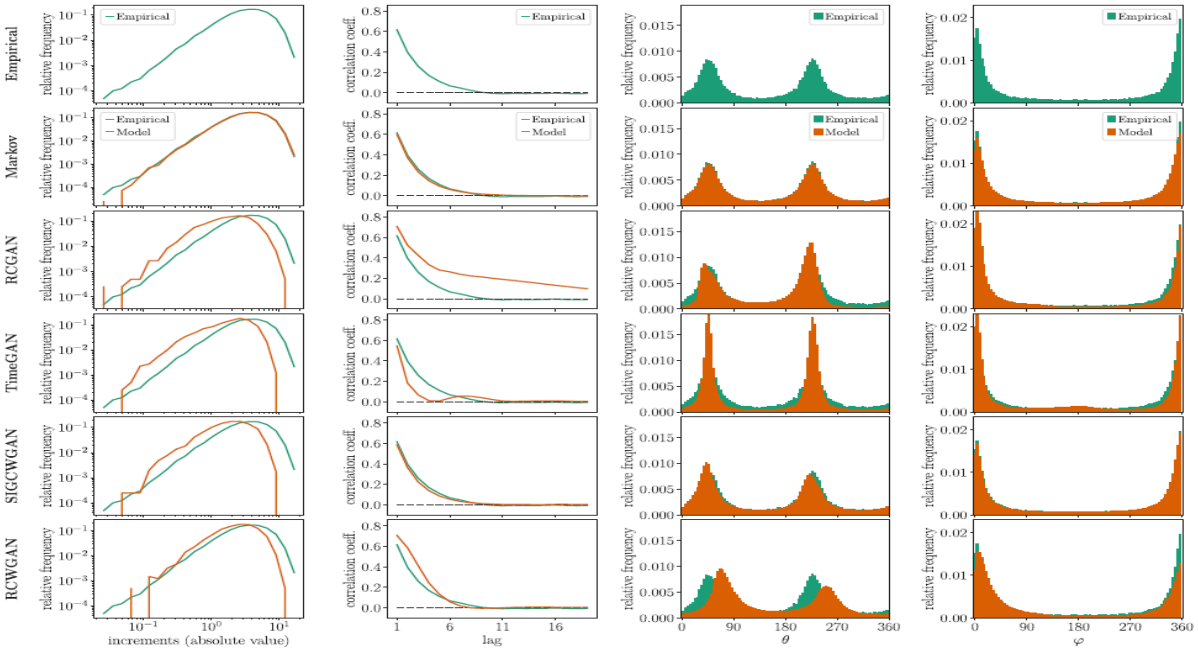

Fig. 4. shows the outcomes of employing time-series GANs and the Markov model to replicate the *VAR(1)* process. The distribution and autocorrelation function of the velocity magnitude ‖v‖ are shown in the first two rows. The distributions of the angular quantities—the turning angle ϕ and the gaze direction θ—are shown in the final two rows.

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 3

The metrics outlined in Eqs. (7) and (9) are used to assess how well algorithms replicate a *VAR* process (top) and gaze trajectories (bottom). The accuracy of recreating the velocity magnitude ‖v‖ distribution is reported in the first column, and the associated autocorrelation function is displayed in the second column. The distances between the synthetic and empirical distributions of the angular measurements θ and ϕ are shown in the third and fourth columns, respectively.

The results reveal that AI-based models consistently fail to replicate the distributions of ‖v‖, θ, and ϕ. In terms of autocorrelation of ‖v‖, SigCWGAN achieves performance comparable to the Markov model. However, it is challenging to draw the conclusion that SigCWGAN accurately depicts the time-evolution dynamics of the process due to the poor replication of the angular distributions.

		Dist. $\lvert v\rvert$ $[D_{\mathrm{JS}}]$	Aut.corr. $\lvert v\rvert$ $[d(f_{\mathrm{emp}}, f_{\mathrm{syn}})]$	Dist. θ $[D_{\mathrm{JS}}]$	Dist. φ $[D_{\mathrm{JS}}]$
VAR(1)	Markov	$\mathbf{6 \cdot 10^{-3}} \pm 2 \cdot 10^{-3}$	$\mathbf{7 \cdot 10^{-2}} \pm 1 \cdot 10^{-2}$	$\mathbf{6 \cdot 10^{-2}} \pm 10^{-2}$	$\mathbf{1.4 \cdot 10^{-1}} \pm 2 \cdot 10^{-2}$
	RCGAN	$3.4 \pm 4 \cdot 10^{-1}$	$7 \cdot 10^{-1} \pm 3 \cdot 10^{-1}$	$3.1 \pm 6 \cdot 10^{-1}$	1.2 ± 10^{-1}
	TimeGAN	2.8 ± 10^{-1}	$3.7 \cdot 10^{-1} \pm 4 \cdot 10^{-2}$	$4.7 \pm 3 \cdot 10^{-1}$	1.5 ± 10^{-1}
	SigCWGAN	2.5 ± 10^{-1}	$1.3 \cdot 10^{-1} \pm 4 \cdot 10^{-2}$	$6 \cdot 10^{-1} \pm 10^{-1}$	$3.2 \cdot 10^{-1} \pm 4 \cdot 10^{-2}$
	RCWGAN	1.0 ± 10^{-1}	$3.0 \cdot 10^{-1} \pm 3 \cdot 10^{-2}$	$6.8 \pm 3 \cdot 10^{-1}$	$2.1 \pm 2 \cdot 10^{-1}$
VAR(2)	Markov	$\mathbf{3.5 \cdot 10^{-3}} \pm 8 \cdot 10^{-4}$	$2.20 \cdot 10^{-1} \pm 6 \cdot 10^{-3}$	$\mathbf{2.8 \cdot 10^{-2}} \pm 4 \cdot 10^{-3}$	$\mathbf{4.5 \cdot 10^{-2}} \pm 5 \cdot 10^{-3}$
	RCGAN	$2.7 \pm 2 \cdot 10^{-1}$	1.6 ± 10^{-1}	$9.8 \pm 3 \cdot 10^{-1}$	$1.7 \pm 2 \cdot 10^{-1}$
	TimeGAN	$2.80 \pm 4 \cdot 10^{-2}$	$1.8 \cdot 10^{-1} \pm 10^{-2}$	$5.00 \pm 2 \cdot 10^{-2}$	$3.00 \pm 8 \cdot 10^{-2}$
	SigCWGAN	$9.3 \cdot 10^{-1} \pm 4 \cdot 10^{-2}$	$\mathbf{4 \cdot 10^{-2}} \pm 10^{-2}$	$10^{1} \pm 10^{-1}$	9.2 ± 10^{-1}
	RCWGAN	(Diverges)	(Diverges)	(Diverges)	(Diverges)
VAR(3)	Markov	$\mathbf{3.1 \cdot 10^{-3}} \pm 8 \cdot 10^{-4}$	$9.2 \cdot 10^{-1} \pm 10^{-2}$	$\mathbf{2.9 \cdot 10^{-2}} \pm 4 \cdot 10^{-3}$	$\mathbf{8.9 \cdot 10^{-2}} \pm 8 \cdot 10^{-3}$
	RCGAN	5.2 ± 10^{-1}	$1.30 \pm 7 \cdot 10^{-2}$	$4 \cdot 10^{1} \pm 1$	$1.20 \cdot 10^{1} \pm 3 \cdot 10^{-1}$
	TimeGAN	(Diverges)	(Diverges)	(Diverges)	(Diverges)
	SigCWGAN	4.2 ± 10^{-1}	$\mathbf{1.8 \cdot 10^{-1}} \pm 5 \cdot 10^{-2}$	$1.60 \cdot 10^{1} \pm 2 \cdot 10^{-1}$	$3.0 \pm 2 \cdot 10^{-1}$
	RCWGAN	(Diverges)	(Diverges)	(Diverges)	(Diverges)
Eye-Gaze	Markov	$\mathbf{1.6 \cdot 10^{-2}} \pm 3 \cdot 10^{-3}$	$\mathbf{3.2 \cdot 10^{-1}} \pm 7 \cdot 10^{-2}$	$\mathbf{3.3 \cdot 10^{-1}} \pm 2 \cdot 10^{-2}$	$\mathbf{4.6 \cdot 10^{-2}} \pm 6 \cdot 10^{-3}$
	RCGAN	$2.600 \pm 3 \cdot 10^{-3}$	$6.7 \cdot 10^{-1} \pm 10^{-2}$	1.1 ± 10^{-1}	$\mathbf{5 \cdot 10^{-2}} \pm 6 \cdot 10^{-2}$
	TimeGAN	$2.600 \pm 3 \cdot 10^{-3}$	$1.100 \pm 5 \cdot 10^{-3}$	1.2 ± 10^{-1}	$4.2 \cdot 10^{-1} \pm 5 \cdot 10^{-2}$
	SigCWGAN	$1.60 \pm 2 \cdot 10^{-2}$	$4.3 \cdot 10^{-1} \pm 2 \cdot 10^{-2}$	$9.3 \cdot 10^{-1} \pm 4 \cdot 10^{-2}$	$1.20 \cdot 10^{1} \pm 10^{-1}$
	RCWGAN	$2.6 \pm 4 \cdot 10^{-1}$	$8.90 \cdot 10^{-1} \pm 8 \cdot 10^{-3}$	$2.2 \pm 2 \cdot 10^{-1}$	$5.1 \pm 2 \cdot 10^{-1}$

100 artificial time series, each with roughly 85,000 data points and the same duration as the real time series, were created for each method.

4.1. Replicating synthetic data

VAR(1) process results (Fig. 4). There is no heavy-tailed behaviour in the distribution of the absolute velocity increments (first column). However, GAN-based models perform consistently across architectures and are unable to replicate the bigger values of the distribution. Their accuracy is two to three orders of magnitude worse than that of the Markov chain model (see Fig. 3). However, the RCWGAN design performs somewhat better than the other GAN variations.

Results of autocorrelation (second column). Once more, the Markov model performs better than any GAN architecture. SigCWGAN outperforms all other GANs, with RCWGAN coming in second. The generator used in these systems, an AR-FNN created especially to capture autocorrelations, is responsible for their relative success [28].

Nevertheless, both GAN models are unable to replicate the high values in the velocity increment distribution, even though they can approximate the positively valued autocorrelation function. Even though the additional kurtosis is still very small, the AI models evaluated here

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

consistently misrepresent the tails of the *VAR* increment distributions, as Figure 4 illustrates. This draws attention to a significant drawback: although GANs may mimic autocorrelation patterns, they are still unable to represent distributional extremes.

The distribution of θ is most accurately represented by the Markov model. SigCWGAN performs substantially worse than the Markov model, which is followed by the RCGAN architecture, even though it produces results that are very similar. While TimeGAN generates a highly concentrated distribution of θ about 45° and 135°, RCWGAN clearly shows an overestimation bias.

Every algorithm performs quite well for the distribution of θ. Although the differences for SigCWGAN are within the margin of error, the Markov model performs somewhat better than SigCWGAN. These results are in line with those found in Ref. [28] for the same dataset, where the other GAN designs demonstrated somewhat better performance while SigCWGAN performed marginally worse.

It is not unexpected that the Markov model accurately reproduces *VAR(1)* processes as they are intrinsically Markovian. Higher-dimensional *VAR* processes provide similar results. The Markov model consistently outperforms all studied GAN structures, as seen in Appendix A (*VAR(2)* and *VAR(3)* examples). Additionally, the results for empirical eye-gaze trajectories show similar trends, with the Markov technique once more outperforming GAN-based models, as will be covered in the next part.

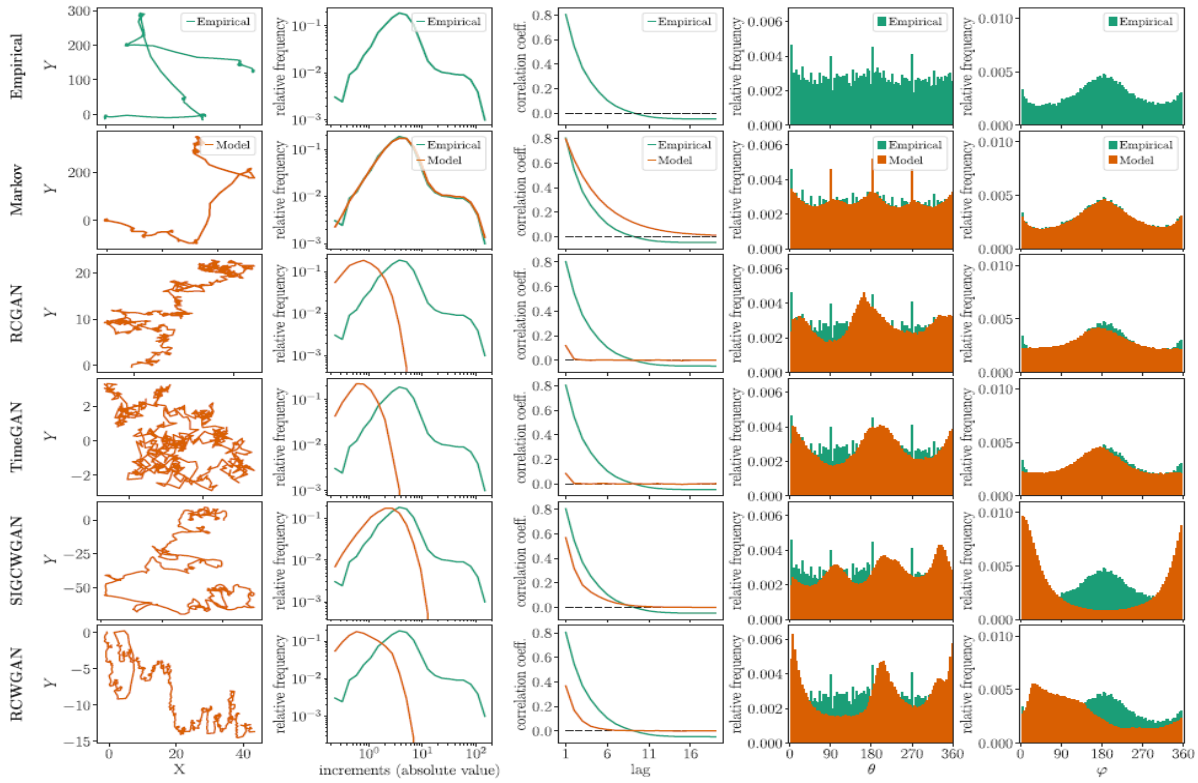

Fig. 5. Eye-gaze trajectory reproduction using time-series GANs and the Markov model. The first 600 points of a typical trajectory are shown in the first column. The distribution and

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

autocorrelation function of the velocity magnitude ‖v‖ are shown in the second and third columns. The distributions of the angular variables θ and ϕ are displayed in the final two rows.

4.2. Replicating empirical data

Empirical eye-gaze trajectories (Fig. 5). The first column presents examples of empirical trajectories alongside their modelled counterparts generated by different algorithms. The distribution of velocity increments (second column) again exhibits no heavy-tailed behaviour. While the Markov model aligns closely with the empirical distribution, the inability of GAN-based models to capture extreme values is even more pronounced than in the case of VAR processes. Among the GANs, SigCWGAN performs slightly better than the other architectures.

Regarding the autocorrelation function, deviations from the Markov condition are expected, as gaze trajectories are not inherently Markovian. Even so, SigCWGAN approximates the autocorrelation function more closely to the Markov model, whereas the remaining architectures only partially reproduce the positive autocorrelations of the process.

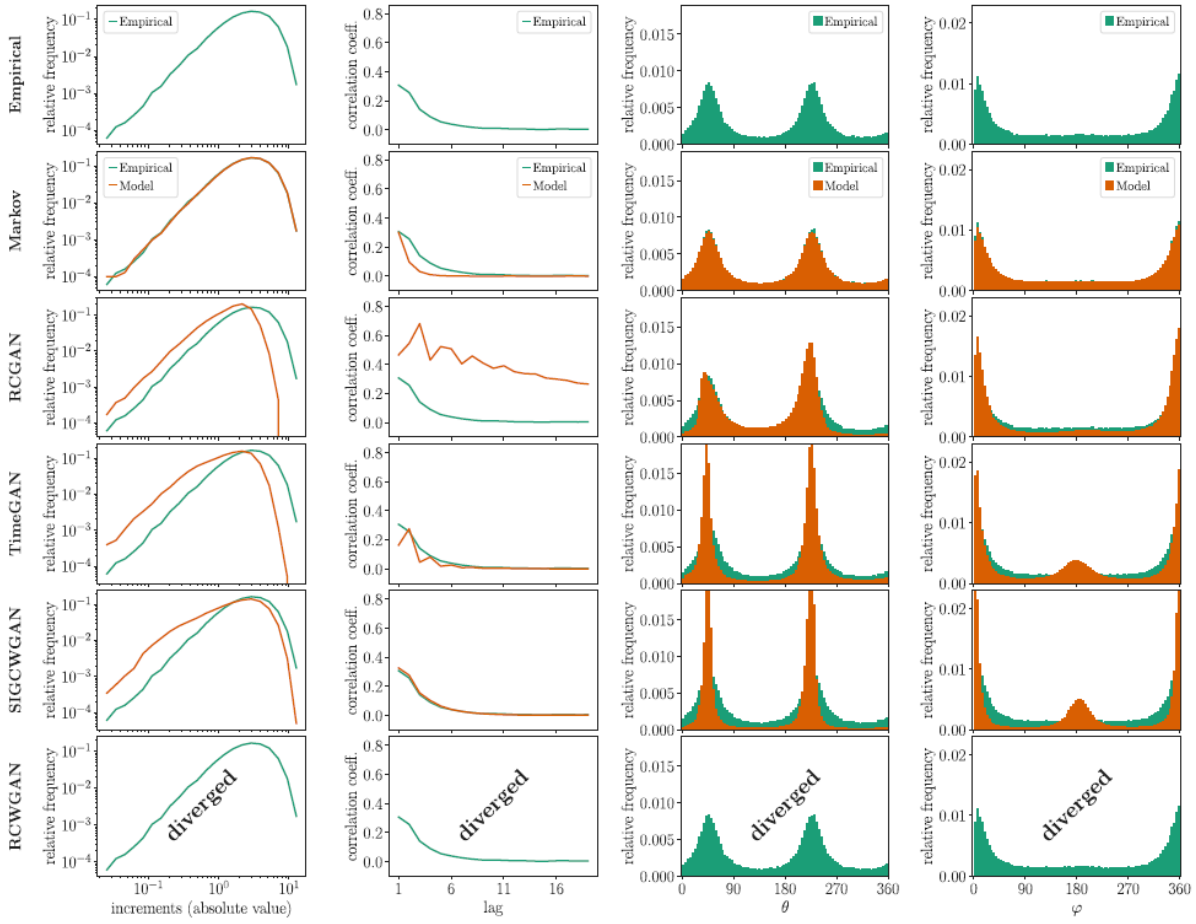

Fig. 6. shows the outcomes of utilizing time-series GANs with the Markov model to replicate the *VAR(2)* process. These results are displayed for direct comparison with the *VAR(1)* scenario depicted in Fig. 4, emphasizing variations in model architectural performance.

The Markov model provides the most accurate representation of the distribution of θ, successfully capturing the preferential directions around 0◦, 90◦, 180◦ and 270◦. However, it

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

fails to reproduce finer details, smoothing out secondary peaks in the distribution. Improved resolution might be achieved with larger trajectory samples or alternative kernel choices when estimating the numerical distribution.

GAN-based models, on the other hand, have trouble capturing this dispersion. They fail to retrieve the preferential angles and generate large biases even though they replicate certain oscillations. Once more, the Markov model performs better than all GANs for the distribution of θ. However, about 180°, TimeGAN and RCWGAN can capture some structure. It's interesting to see that SigCWGAN does not duplicate the distribution of ϕ, despite partially reproducing the autocorrelation function. This example shows an uncommon situation in which a time series maintains a positive autocorrelation despite displaying increments with angles grouped around 180°. One possible explanation is that the algorithm prefers to produce trajectories that continue in the same direction to maintain positive autocorrelation without reproducing the higher values of the distribution.

5. Discussion and Conclusions

The comparison research shows that the Markov model consistently performs better than every GAN design this study looked at. SigCWGAN is the GAN variation that performs the best, matching the Markov implementation in approximating the distribution of ϕ, although it is not quite as good at modeling the distribution of θ. Particularly with regard to the distribution of θ, the remaining GAN architectures perform quite poorly and fall short of the Markov process's statistical efficacy.

While it is encouraging that a particular GAN architecture can replicate certain statistical properties at a level comparable to a Markov process—suggesting potential for application to more complex phenomena—our findings demonstrate that simple Markov models remain substantially more effective in modelling the overall distribution of the process. Unlike GANs, which typically require parameter counts two orders of magnitude larger and operate as opaque, black-box systems, Markov models are comparatively transparent and computationally efficient. Additionally, calculating a Markov transition matrix not only makes distributional modelling easier, but it also makes it possible to determine basic properties of the underlying natural process, including whether it is stationary or time-continuous [22].

Both processes with very high excess kurtosis, such 200 Hz free-viewing eye-gaze trajectories, and those with insignificant excess kurtosis, like *VAR(p)* models, exhibit GAN restrictions. Although we do not claim that non-parametric Markov models are always better than AI-based methods, our findings show that when non-parametric modelling is needed, it is not reasonable to assume that AI methods are the best option. We demonstrate that conventional mathematical models can outperform their AI equivalents in both a highly complicated empirical dataset and a straightforward synthetic time-series. As a result, we recommend against using AI algorithms for time-series prediction without first comparing them to more straightforward, comprehensible approaches.

This paper's evidence can be expanded in several ways. First, future research could investigate different GAN architectures and datasets, even if the GAN models studied here reflect some of

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

the most popular AI techniques and the processes taken into consideration include both Gaussian and non-Gaussian properties. Models like SigCWGAN and TimeGAN that are intended to capture the extreme values of a distribution should get special consideration. In addition, integrating datasets that reflect other kinds of processes—such as those driven by jump-diffusion noise [35] or intermittent dynamics [36]—would permit a more thorough exploration into the bounds of our findings.

Conversely, potential extensions of the Markov framework warrant exploration—for instance, models with a Markov length exceeding a single time lag. A promising hybrid strategy could also involve employing GANs to model the residuals of a Markov process or, alternatively, training GANs to generate Markov transition matrices, potentially in higher-dimensional settings.

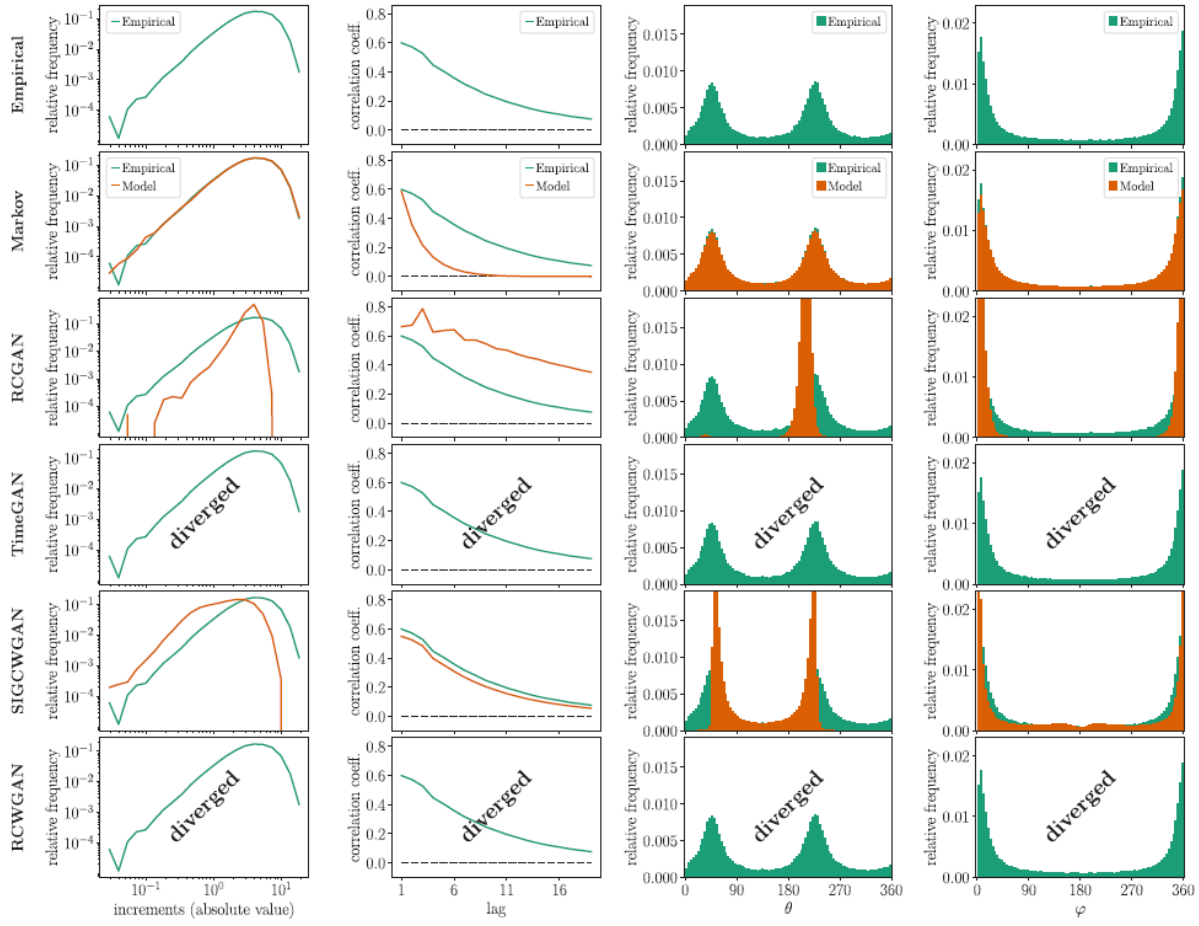

Fig. 7. shows the outcomes of using a Markov model with time-series GANs to replicate the *VAR(3)* procedure. for contrast with Figures 4 and 6.

When we compared our training process to that described in earlier research on the same dataset [28], we found that training statistics and results were generally quite consistent.

As conditional models, the GANs assessed in this study produce subsequent points based on earlier ones. In this case, the number of previous points—often represented as p in the literature—that were utilized to forecast the subsequent set was methodically optimized. The best option, according to the search, was $p=3$. To guarantee optimal performance, additional

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

hyperparameters, such as network size and batch size, were also improved through comparative trials.

In conclusion, even though GANs are frequently said to be able to capture the time-evolution of stochastic processes, complex time-series do not support this assertion. Even when some architectures are said to be able to replicate extreme values, their shortcomings are still apparent, especially when it comes to their inability to fully capture the range of the auto-correlation function. At first appearance, it seems surprising that AI techniques cannot outperform a theoretically straightforward mathematical framework. However, a large portion of the literature focuses solely on their achievements, ignoring situations in which they fail. The central limit theorem essentially limits GANs since the data they produce usually converges to a normal distribution. Redesigning conventional GAN designs to include noise inputs derived from α-stable distributions could be one way to overcome this restriction.

Appendix A. Replicating VAR(2) and VAR(3) processes – Results

Applying our methodology to synthetic processes that are non-Markov is as enlightening, even though gaze trajectories show non-Markov behaviour. These include the *VAR(p=2)* and *VAR(p=3)* processes as defined in Eq. (4), in contrast to the *VAR(1)* process.

We utilized $\phi_1=0.5$ and $\phi_2=0.4$ for the *VAR(2)* process and $\phi_1=0.3$, $\phi_2=0.3$, and $\phi_3=0.3$ for the *VAR(3)* process. The correlation coefficient between the two spatial dimensions was set at 0.8 in both situations. As previously, we produced an 85,000-point time series with a sample period of $\Delta t = 1$. Table 3 displays the corresponding results, which are depicted in Figs. 6 and 7.

We found that several GAN-generated time-series in this research deviated to infinity. In particular, RCWGAN showed divergence in the *VAR(2)* scenario, but TimeGAN and RCWGAN both showed divergence in the *VAR(3)* procedure. Under different parameter selections, similar behavior was also seen, with RCGAN occasionally diverging as well. In contrast, when SigCWGAN was used with convergent time-series, it did not diverge. Since some GANs continued to generate divergent sequences, modifications to training epochs or learning rates did not significantly alleviate these problems. Interestingly, the auto-correlation function of *VAR(2)* and *VAR(3)* processes was reliably reproduced by SigCWGAN with an accuracy like that attained in the *VAR(1)* instance. The Markov process did not exhibit this degree of replication, as was to be expected.

Even while the ANN used in SigCWGAN (the AR-FNN) is specifically made to duplicate the auto-correlation structure of the data, its incapacity to precisely capture the distribution of ϕ indicates that the algorithm is unable to replicate several crucial features of the temporal dynamics of the process.

In examining the distribution of process increments, we find that GANs, much like in the *VAR(1)* case and gaze trajectories, tend to underestimate the statistical moments—particularly the standard deviation. Furthermore, GANs are unable to replicate the θ distribution, suggesting that the link between the two spatial dimensions is not sufficiently captured by them.

In conclusion, SigCWGAN shows a relative advantage in reproducing the auto-correlation function for the *VAR(2)* and *VAR(3)* processes. Nevertheless, its performance in modelling the

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

distribution of ϕ remains markedly inferior to that of the Markov model. In general, GANs are unable to accurately capture the dependency between the two spatial dimensions, as demonstrated by the distribution of θ, or replicate the distribution of increments. Therefore, the advantages of GANs are limited even in the case of simple synthetic data where the Markov assumption is violated.

Appendix B. GANs Training

Training GANs presents significant challenges, as the process is often unstable and does not guarantee progressive improvement across epochs [8]. Two common pitfalls are vanishing gradients and mode collapse. In the former, the discriminator becomes so effective that both networks cease to learn over successive epochs. In the latter, the generator deceives the discriminator by producing only a limited set of modes, neglecting others and yielding highly similar time-series. To mitigate these issues, careful tuning of the learning rates for both networks is typically recommended.

In our implementation, extending GAN training over longer periods did not consistently yield improved outcomes. We also noticed instances of mode collapse when simulating real time-series, demanding careful calibration of learning rates. Empirically, training the GANs for approximately 200 epochs reduced both training and testing errors while mitigating the risk of mode collapse. The Adam optimizer was used to update the neural networks' weights [37].

Appendix C. Application of the Markov-chain model

By calculating the conditional probability $Pr(X_{t=n+1} = x_{n+1} \mid X_{t=n} = x_n)$ in a Markov model, a time-series can be produced. We use a Gaussian kernel K to empirically estimate this probability.

$$\Pr(X_{t=n+1} = x_{n+1} \mid X_{t=n} = x_n) = \frac{\Pr(X_{t=n+1} = x_{n+1}, X_{t=n} = x_n)}{\Pr(X_{t=n} = x_n)} \tag{C.10}$$

with

$$\Pr(X_{t=n+1} = x_{n+1}, X_{t=n} = x_n) = \frac{1}{(\widehat{N}-1)^2 h^2} \sum_{i=1}^{\widehat{N}-1} K\left(\frac{x_{n+1} - \hat{x}_{i+1}}{h}\right) K\left(\frac{x_n - \hat{x}_i}{h}\right) \tag{C.11}$$

$$\Pr(X_{t=n} = x_n) = \frac{1}{h(\widehat{N}-1)} \sum_{i=1}^{\widehat{N}-1} K\left(\frac{x_n - \hat{x}_i}{h}\right) \tag{C.12}$$

where

Received: August 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$K\left(\frac{x_n - \hat{x}_i}{h}\right) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{x_n - \hat{x}_i}{h}\right)^2\right) \quad \text{(C.13)}$$

Here, h stands for the Gaussian estimate kernel *K's* bandwidth, which is established using Silverman's rule [38].

$$h = \left(\frac{4\hat{\sigma}^5}{3\widehat{N} - 3}\right)^{\frac{1}{5}} \approx 1.06\hat{\sigma}(\widehat{N} - 1)^{-1/5} \quad \text{(C.14)}$$

Here, $\hat{\sigma}$ denotes the standard deviation of $\hat{x}_1 \ldots\ldots \hat{x}_n$ and $\widehat{N}$ represents the number of data points in the sample.

When empirical data analysed, the conditional probability $Pr(X_{t=n+1} \mid X_{t=n})$ can be expressed as a T matrix of dimension $N_s \times N_s$, and entries given by:

$$T_{i,j} = Pr(X_{t=n+1} \subset [k_i, k_{i+1}) | X_{t=n} \subset [k_j, k_{j+1})) \quad \text{(C.15)}$$

here $i, j \in N$, as $i, j \subset [0, N_s]$ and $k_m > k_n \Leftrightarrow m > n$, the state $X_{t=n} \subset [k_j, k_{j+1})$ allows us to compute the observing probability $X_{t=n+1} \subset [k_i, k_{i+1})$. When creating a new time-series, if $X_{t=n+1} \subset [k_i, k_{i+1})$, the value will be assigned by sampling uniformly in the interval $[k_i, k_{i+1})$.

Three important elements affect this method's accuracy. Firstly, the Markov condition validity will affect (3). Secondly, the state numbers N_s will influence, which is limited by computational cost and scales about as N_s^4 for a process of two-dimensions. Thirdly, sample size $\widehat{N}$ has an impact on accuracy, since this directly impacts the bandwidth h: larger samples yield smaller values of h, thereby enhancing the model's precision. A Python and NumPy implementation of this algorithm is available on GitHub [39].

References

[1] A. Flores, H. Tito-Chura, V. Yana-Mamani, Wind speed time series prediction with deep learning and data augmentation, in: Proceedings of SAI Intelligent Systems Conference, Springer, 2021, pp. 330–343, http://dx.doi.org/10.1007/978-3-030-82193-7_22.

[2] P.G. Lind, L. Vera-Tudela, M. Wächter, M. Kühn, J. Peinke, Normal behaviour models for wind turbine vibrations: Comparison of neural networks and a stochastic approach, Energies 10 (12) (2017) 1944, http://dx.doi.org/10.3390/en10121944.

[3] A. Russo, F. Raischel, P. Lind, Air quality prediction using optimal neural networks with stochastic variables, Atmos. Environ. 79 (2013) 822–830, http://dx.doi.org/10.1016/j.atmosenv.2013.07.072.

Received: August 18, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[4] J. Cao, Z. Li, J. Li, Financial time series forecasting model based on CEEMDAN and LSTM, Phys. A 519 (2019) 127–139, http://dx.doi.org/10.1016/j.physa.2018.11.061.

[5] P. Wang, X. Zheng, G. Ai, D. Liu, B. Zhu, Time series prediction for the epidemic trends of COVID-19 using the improved LSTM deep learning method: Case studies in Russia, Peru and Iran, Chaos Solitons Fractals 140 (2020) 110214, http://dx.doi.org/10.1016/j.chaos.2020.110214.

[6] F. Harrou, A. Dairi, F. Kadri, Y. Sun, Forecasting emergency department overcrowding: A deep learning framework, Chaos Solitons Fractals 139 (2020) 110247, http://dx.doi.org/10.1016/j.chaos.2020.110247.

[7] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S.Ozair, A. Courville, Y. Bengio, Generative adversarial networks, 2014, http://dx.doi.org/10.48550/arXiv.1406.2661, arXiv preprint arXiv:1406.2661.

[8] E. Brophy, Z. Wang, Q. She, T. Ward, Generative adversarial networks in time series: A survey and taxonomy, 2021, http://dx.doi.org/10.48550/arXiv.2107.11098, arXiv preprint arXiv:2107.11098.

[9] D. Hazra, Y.-C. Byun, SynSigGAN: Generative adversarial networks for synthetic biomedical signal generation, Biology 9 (12) (2020) 441, http://dx.doi.org/10.3390/biology9120441.

[10] L. Yu, W. Zhang, J. Wang, Y. Yu, Seqgan: Sequence generative adversarial nets with policy gradient, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, 2017, p. 1, URL https://www.aaai.org/Conferences/AAAI/2017/PreliminaryPapers/12-Yu-L-14344.pdf.

[11] O. Mogren, C-RNN-GAN: Continuous recurrent neural networks with adversarial training, 2016, http://dx.doi.org/10.48550/arXiv.1611.09904, arXivpreprint arXiv:1611.09904.

[12] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, Y.-H. Yang, Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, p. 1, URL https://salu133445.github.io/musegan/pdf/musegan-aaai2018-paper.pdf.

[13] Z. Lv, X. Huang, W. Cao, An improved GAN with transformers for pedestrian trajectory prediction models, Int. J. Intell. Syst. 37 (8) (2021) 4417–4436, http://dx.doi.org/10.1002/int.22724.

[14] M. Wiese, R. Knobloch, R. Korn, P. Kretschmer, Quant GANs: Deep generation of financial time series, Quant. Finance 20 (9) (2020) 1419–1440, http://dx.doi.org/10.1080/14697688.2020.1730426.

[15] S. Berkovsky, R. Taib, I. Koprinska, E. Wang, Y. Zeng, J. Li, S. Kleitman, Detecting personality traits using eye-tracking data, in: Proceedings of the 2019 CHI Conference

Received: August 18, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

on Human Factors in Computing Systems, 2019, pp. 1–12, http://dx.doi.org/10.1145/3290605.3300451.

[16] M. Steffens, B. Becker, C. Neumann, A. Kasparbauer, I. Meyhöfer, B. Weber, M. Mehta, R. Hurlemann, U. Ettinger, Effects of ketamine on brain function during smooth pursuit eye movements, Hum. Brain Mapp. 37 (11) (2016) 4047–4060, http://dx.doi.org/10.1002/hbm.23294.

[17] P.M. Grace, T. Stanford, M. Gentgall, P.E. Rolan, Utility of saccadic eye movement analysis as an objective biomarker to detect the sedative interaction between opioids and sleep deprivation in opioid-naïve and opioid-tolerant populations, J. Psychopharmacol. 24 (11) (2010) 1631–1640, http://dx.doi.org/10.1177/0269881109352704.

[18] H. Chauhan, A. Prasad, J. Shukla, Engagement analysis of ADHD students using visual cues from eye tracker, in: Companion Publication of the 2020 International Conference on Multimodal Interaction, 2020, pp. 27–31, http://dx.doi.org/10.1145/3395035.3425256.

[19] A. Lev, Y. Braw, T. Elbaum, M. Wagner, Y. Rassovsky, Eye tracking during a continuous performance test: Utility for assessing ADHD patients, J. Atten. Disord. 26 (2) (2022) 245–255, http://dx.doi.org/10.1177/1087054720972786.

[20] T. Wadhera, D. Kakkar, Eye tracker: An assistive tool in diagnosis of autism spectrum disorder, in: Emerging Trends in the Diagnosis and Intervention of Neurodevelopmental Disorders, IGI Global, 2019, pp. 125–152, http://dx.doi.org/10.4018/978-1-5225-7004-2.ch007.

[21] B. Hayes, et al., First links in the Markov chain, Am. Sci. 101 (2) (2013) 252, http://dx.doi.org/10.1511/2013.101.92.

[22] P. Lencastre, F. Raischel, T. Rogers, P. Lind, From empirical data to timeinhomogeneous continuous Markov processes, Phys. Rev. E 93 (3) (2016) 032135, http://dx.doi.org/10.1103/PhysRevE.93.032135.

[23] R. Huang, B. Xu, D. Schuurmans, C. Szepesvári, Learning with a strong adversary, 2015, http://dx.doi.org/10.48550/arXiv.1511.03034.

[24] S.L. Hyland, C. Esteban, G. Rätsch, Real-valued (medical) time series generation with recurrent conditional GANs, 2017, http://dx.doi.org/10.48550/arXiv.1706.02633, arXiv preprint arXiv:1706.02633.

[25] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[26] J. Yoon, D. Jarrett, M. Van der Schaar, Time-series generative adversarial networks, Adv. Neural Inf. Process. Syst. 32 (2019) URL http://papers.neurips.cc/paper/8789-time-series-generative-adversarial-networks.pdf.

Received: August 18, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[27] D.P. Kingma, M. Welling, et al., An introduction to variational autoencoders, Found. Trends Mach. Learn. 12 (4) (2019) 307–392, http://dx.doi.org/10.1561/2200000056.

[28] H. Ni, L. Szpruch, M. Wiese, S. Liao, B. Xiao, Conditional sig-Wasserstein GANs for time series generation, 2020, http://dx.doi.org/10.48550/arXiv.2006.05421, arXiv preprint arXiv:2006.05421.

[29] H. Ni, L. Szpruch, M. Wiese, S. Liao, B. Xiao, Conditional-Sig-Wasserstein-GANs, 2020, URL https://github.com/SigCGANs/Conditional-Sig-Wasserstein-GANs.

[30] A.A. Markov, Extension of the law of large numbers to dependent quantities, Izv. Fiz.-Matem. Obsch. Kazan Univ. (2nd Ser.) 15 (1) (1906) 135–156.

[31] R.M. Klein, W.J. MacInnes, Inhibition of return is a foraging facilitator in visual search, Psychol. Sci. 10 (4) (1999) 346–352, http://dx.doi.org/10.1111/1467-9280.00166.

[32] A. Borji, Pros and cons of GAN evaluation measures, Comput. Vis. Image Underst. 179 (2019) 41–65, http://dx.doi.org/10.1016/j.cviu.2018.10.009.

[33] A. Borji, Pros and cons of GAN evaluation measures: New developments, Comput. Vis. Image Underst. 215 (2022) 103329,

http://dx.doi.org/10.1016/j.cviu.2021.103329.

[34] K. Shmelkov, C. Schmid, K. Alahari, How good is my GAN? in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 213–229, http://dx.doi.org/10.1007/978-3-030-01216-8_14.

[35] L. Rydin Gorjão, D. Witthaut, P.G. Lind, JumpDiff: Non-parametric numerical estimation of jump-diffusion processes, J. Stat. Softw. 15 (2023) 1–22, http://dx.doi.org/10.18637/jss.v105.i04.

[36] P. Lencastre, S. Denysov, A. Yazidi, P.G. Lind, Uncovering Lévy flights and intermittent processes from empirical time series: a theoretical framework to classify, 2023, in prepartion.

[37] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, http://dx.doi.org/10.48550/arXiv.1412.6980, arXiv preprint arXiv:1412.6980.

[38] B.W. Silverman, Density Estimation for Statistics and Data Analysis, first ed., Routledge, New York, 1998, http://dx.doi.org/10.1201/9781315140919.

[39] P. Lencastre, Markov model, 2023, URL https://github.com/134f/Physica-DMarkov-model.

Received: August 18, 2025