Distinguishing Dynamical Kinds
An Approach for Automating Scientific Discovery

Colin R. Shea-Blymyer

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science and Applications

Benjamin C. Jantzen, Chair

Bert Huang

Anuj Karpatne

B. Aditya Prakash

May 13, 2019

Blacksburg, Virginia

Keywords: Data Analysis, Dynamical Kinds, Nonlinear Systems, Chaos, Automated Scientific Discovery, Order Identification

Distinguishing Dynamical Kinds
An Approach for Automating Scientific Discovery

Colin R. Shea-Blymyer

(ABSTRACT)

The automation of scientific discovery has been an active research topic for many years. The promise of a formalized approach to developing and testing scientific hypotheses has attracted researchers from the sciences, machine learning, and philosophy alike. Leveraging the concept of dynamical symmetries a new paradigm is proposed for the collection of scientific knowledge, and algorithms are presented for the development of EUGENE – an automated scientific discovery tool-set. These algorithms have direct applications in model validation, time series analysis, and system identification. Further, the EUGENE tool-set provides a novel metric of dynamical similarity that would allow a system to be clustered into its dynamical regimes. This dynamical distance is sensitive to the presence of chaos, effective order, and nonlinearity. I discuss the history and background of these algorithms, provide examples of their behavior, and present their use for exploring system dynamics.

Distinguishing Dynamical Kinds
An Approach for Automating Scientific Discovery

Colin R. Shea-Blymyer

(GENERAL AUDIENCE ABSTRACT)

Determining why a system exhibits a particular behavior can be a difficult task. Some turn to causal analysis to show what particular variables lead to what outcomes, but this can be time-consuming, requires precise knowledge of the system's internals, and often abstracts poorly to salient behaviors. Others attempt to build models from the principles of the system, or try to learn models from observations of the system, but these models can miss important interactions between variables, and often have difficulty recreating high-level behaviors. To help scientists understand systems better, an algorithm has been developed that estimates how similar the causes of one system's behaviors are to the causes of another. This similarity between two systems is called their "dynamical distance" from each other, and can be used to validate models, detect anomalies in a system, and explore how complex systems work.

Dedication

To my parents, for their continuous encouragement and masterful support;

and to the pendragons, for the many roles they've played.

Acknowledgments

The work presented here was supported by the National Science Foundation under Grant No. 1454190 and grant No.CMMI-1708622, and by a Data & Decisions seed grant from Virginia Tech.

The author acknowledges Advanced Research Computing at Virginia Tech for providing computational resources and technical support that have contributed to the results reported within this paper. URL: http://www.arc.vt.edu

I would like to thank my advisor, Ben Jantzen, for bringing me into the world of research, for helping me thrive, for his aid in my work, and for keeping the lab lively. I am also grateful for the support of my committee members; each of whom have enthusiastically supported my research. Further thanks go to Nicole Abaid, for her supervision over my assistantship, her support of my academic career, and for her influence on my research. Thanks also to Subhradeep Roy, Amanda Hashimoto, and the members of the Digital Philosophy Lab.

Contents

List of Figures **viii**

1 Introduction **1**

2 Project History and Personal Contributions **2**

3 A General Metric for the Similarity of Stochastic and Deterministic System Dynamics **5**

4 Differentiation of Collective Behavior Based on Automated Discovery of Dynamical Kinds **37**

5 Summary **45**

Bibliography **47**

Appendix A Distance Characterization **50**

A.1 Lotka-Volterra Equations 50

A.2 Test Framework 51

A.3 Test Results 52

A.3.1 Comparing Deterministic Systems as Data Increases 52

A.3.2 Comparing Stochastic Systems as Data Increases 64

A.3.3 Comparing Stochastic Systems as Number of Replicates Increases . . 64

A.3.4 Comparing Stochastic Systems as Magnitude of Stochastic Noise Increases 72

A.3.5 Comparing Deterministic Systems with Different Levels of Observability, as Sampling Rate Increases 72

A.4 Test Results Discussion 78

List of Figures

3.1 Symmetry Commutation 7

3.2 Distance Between Two Systems 14

3.3 Two Systems with Same Capacity and Different Linearity 16

3.4 Two Systems with Same Capacity and Different Effective Order 17

3.5 Two Systems with Scaled Interaction Term through Chaos 19

3.6 Two Systems with Different Interaction Term through Chaos 20

3.7 Distance between Systems of Varying Degrees of Chaos 21

3.8 Exploration of Chaos in Lotka-Volterra Systems 23

3.9 Exploration of Distance in Lotka-Volterra Systems 24

3.10 Data produced by Kuramoto Model 25

3.11 Dynamic Anomaly Detection 26

3.12 Matrix Profile Anomaly Detection 26

3.13 Distance Between Two Stochastic Systems 28

3.14 Distance Between Two Partially Observable Stochastic Systems 31

3.15 Dynamic Anomaly Detection on Partially Observable System 32

4.1 Mean Values of Steady-State Group Parameters 39

4.2 Snapshots of Group Motion for the Five Behavioral States 40

4.3 Time Series Data for Each Behavioral State 41

4.4 Schematic of EUGENE Algorithm 42

4.5 Distance Between Five Behavioral States as Computed by EUGENE 43

A.1 Dynamical distance between system A and a different system B as sampling rate increases. 52

A.2 Dynamical distance between system A and a similar system B as sampling rate increases. 53

A.3 Dynamical distance between system A and a different system B as sampling rate increases, and measurement noise is included. 54

A.4 Dynamical distance between system A and a similar system B as sampling rate increases, and measurement noise is included. 55

A.5 Dynamical distance between system A and a different system B as sampling rate increases, and clipping is not used. 56

A.6 Dynamical distance between system A and a similar system B as sampling rate increases, and clipping is not used. 57

A.7 Dynamical distance between system A and a different system B as sampling rate increases, measurement noise is included, and clipping is not used. . . . 58

A.8 Dynamical distance between system A and a similar system B as sampling rate increases, measurement noise is included, and clipping is not used. . . . 59

A.9 Dynamical distance between system A and a different system B as sampling rate increases. 60

A.10 Dynamical distance between system A and a similar system B as sampling rate increases. 61

A.11 Dynamical distance between system A and a different system B as sampling rate increases, without clipping. 62

A.12 Dynamical distance between system A and a similar system B as sampling rate increases, without clipping. 63

A.13 Dynamical distance between system A and a different system B as number of replicates increases. 64

A.14 Dynamical distance between system A and a similar system B as number of replicates increases. 65

A.15 Dynamical distance between system A and a different system B as number of replicates increases, without clipping. 66

A.16 Dynamical distance between system A and a similar system B as number of replicates increases, without clipping. 67

A.17 Dynamical distance between system A and a different system B as magnitude of stochastic noise increases. 68

A.18 Dynamical distance between system A and a similar system B as magnitude of stochastic noise increases. 69

A.19 Dynamical distance between system A and a different system B as magnitude of stochastic noise increases, without clipping. 70

A.20 Dynamical distance between system A and a similar system B as magnitude of stochastic noise increases, without clipping. 71

A.21 Dynamical distance between system A and a different system B as sampling rate increases. Red denotes a system with a SSD variable set. Blue denotes a system with a non-SSD variable set. 72

A.22 Dynamical distance between system A and a similar system B as sampling rate increases. Red denotes a system with a SSD variable set. Blue denotes a system with a non-SSD variable set. 73

A.23 Dynamical distance between system A and a different system B as sampling rate increases, without clipping. Red denotes a system with a SSD variable set. Blue denotes a system with a non-SSD variable set. 74

A.24 Dynamical distance between system A and a similar system B as sampling rate increases, without clipping. Red denotes a system with a SSD variable set. Blue denotes a system with a non-SSD variable set. 75

A.25 Dynamical distance between system A and a different system B as sampling rate increases. Data is clipped, and system is rescanned to maximize data resolution. Red denotes a system with a SSD variable set. Blue denotes a system with a non-SSD variable set. 76

A.26 Dynamical distance between system A and a similar system B as sampling rate increases. Data is clipped, and system is rescanned to maximize data resolution. Red denotes a system with a SSD variable set. Blue denotes a system with a non-SSD variable set. 77

Attribution

For *A general metric for the similarity of stochastic and deterministic system dynamics* I performed literature review, algorithmic development, produced, analyzed, and lead discussion on the results. Subhradeep Roy performed literature review, and Benjamin Jantzen performed literature review, communicated theory, and produced and analyzed results.

For *Differentiation of Collective Behavior Based on Automated Discovery of Dynamical Kinds* Amanda Hashimoto developed and executed models, performed literature review, and discussed results. Nicole Abaid discussed background, results, and model design. Subhradeep Roy performed literature review, and produced and discussed results. Benjamin Jantzen performed algorithmic development and discussed results, and I performed algorithmic development, and discussed algorithmic theory.

Chapter 1

Introduction

The automation of scientific discovery has been pursued by researchers for almost as long as computers have been in the hands of scientists. Early attempts include Herbert Simon's efforts to use heuristics to discover chemical laws with BACON [7]. Modern approaches rely on more sophisticated machine learning techniques, and are aided by more powerful machinery [13, 15].

A year before he won an ACM Turing Award, Yann LeCun was asked about the diversity of AI research during a public Reddit interview. He responded, "But really, we are missing at least two things: (1) learning machines that can reason, not just perceive and classify, (2) learning machines that can learn by observing the world, without requiring human-curated training data, and without having to interact with the world too many times."[18] It is in the spirit of that second class of research that EUGENE is developed.

The algorithms presented in this thesis are fundamental parts of a greater endeavor in the automation of scientific discovery. These methods largely serve as a similarity/distance measure of a system's underlying dynamics, and are certainly novel among its peers [8]. Further, they have already been used for causal discovery, and model validation [5, 12]. What likens these methods to LeCun's second class of research, however, is that they are as much the formalization of an experimental process as they are a metric over which one can cluster data.

Chapter 2

Project History and Personal Contributions

The EUGENE project is named after the Hungarian-American scientist Eugene Wigner in the style of Simon's BACON project to honor the inspiration received from his theory of symmetry in physics. For a broad class of systems in fact, it is a subset of the Lie symmetries (which underpin Wigner's symmetry theory, and are named for the Norwegian mathematician Marius Lie) of a system that determine its dynamical kind [2]. While a system's Lie symmetries can be determined analytically, the process to do so is intractable in all but the simplest cases, and requires the system to be expressed mathematically.

During my undergraduate career, I joined Dr. Ben Jantzen's EUGENE project and was instructed to develop a symbolic regression routine in order to circumvent the limitations of solving a system's Lie symmetries by hand [16]. The other advantage of this approach to discovering "dynamical symmetries" was that it could fit mathematical expressions to raw data, allowing further analytic methods to be used. I developed this algorithm with Joseph Mehr, another undergraduate student, and presented our results at the Virginia Tech Undergraduate Research in Computer Science Spring Symposium. This genetic algorithm could successfully find the symmetries of the logistic growth equation, but struggled with more complex systems, and exhibited a troublesome fondness for the identity symmetry.

Between my senior year of my undergraduate program, and the beginning of my graduate

career, I continued to work on the EUGENE project, during which time I helped design EUGENE v1 [3]. My primary contribution to this version is the development of a routine referred to as "range cover" - a process designed to maximize the coincident regions of inputs, so none would have a greater range than the others. Range cover effectively freed us from having to hand tune simulation parameters in order to ensure that EUGENE could draw comparisons between seemingly disparate systems. Version 1 of the EUGENE algorithm is the direct precursor of the modern EUGENE algorithm, and is based upon the same principles. This version of the algorithm showed that comparing the mapping of one system's untransformed and transformed trajectories to the mapping of another system, EUGENE could determine, with a high degree of accuracy, even in the presence of noise, if the two systems were of the same dynamical kind [4].

EUGENE v1 was limited, however, to deterministic, fully observable, fully controllable systems - conditions that are difficult to replicate in the real world. These limitations motivated the development of the current EUGENE v2 set of algorithms. A generalization of the previous algorithm, EUGENE v2 has been designed to estimate two systems' dynamical similarity, even when lacking information, or control over a system. By treating each system's input as a distribution, I suggested that a statistical distance could be used to estimate the dynamical similarity between the two systems.

With a generalized algorithm came a number of practical problems. The most prominent was the computational complexity of the new approach. In calculating the Hellinger distance between the two systems' distributions, EUGENE was attempting to find each distribution's probability density function using kernel density estimation - a time consuming task [1, 17]. This problem was solved by switching, at Ben Jantzen's suggestion, to the energy distance between distributions, as it only required cumulative distribution functions, greatly reducing the time needed to run the algorithm [11].

One problem that gained importance as we dealt with real-world systems was the problem of mis-matched dynamic range. If one system took too long to reach equilibrium, for instance, and another - too short, then comparing the two would result in a distorted dynamical distance. I had originally approached this problem with range cover (which remains part of the EUGENE package), but Ben and I later developed a clipping algorithm (see chapter 3) reminiscent of dynamic time warping [14].

The largest problems that arose from the generalized approach had to do with the lack of control over passively observed systems. To select untransformed and transformed pairs of distributions, Ben developed an algorithm to maximize the difference in the principal components of the new selections (chapter 3). Similarly, when dealing with discrete data, I developed an algorithm that would select near-optimal-length sets of data without overlap, allowing for more accurate determination of kind similarity. Using these algorithms, I developed a tool that allows us to designate untransformed and transformed sets of data when provided with specific experimental treatments.

As the project matured, I remained diligent in testing the bounds of the algorithm - comparing the trade-offs of various data-treatments and parameter selections, and testing performance in the presence of measurement noise and stochasticity (see appendix A). I also helped design a tool for detecting change in dynamical kind, inspired by algorithms like the Matrix Profile [20]. To further the testing of the algorithm on stochastic systems, I developed a stochastic differential equation that incorporated Brownian action into the Lotka-Volterra systems of equations [19]. In fact, developing simulations became something of a specialty. Each new feature of the project demanded a new system be tested on. In this pursuit I developed multiple generalizations of the Kuramoto model [6] and a damped harmonic oscillator for the study of nonlinearity, a multiple-pendulum simulation for the study of chaos, and deterministic Lotka-Volterra systems that allow for varying degrees of nonlinearity or

effective model order.

Most recently, I have been responsible for a wide sweep of the parameter space of the Lotka-Volterra equations. This has included use of Virginia Tech's Advanced Research Computing resources, and the development of a visualization of the similarity of systems in parameter space using multi-dimensional scaling to reduce a distance matrix to color-space.

My contributions to the EUGENE project have indirectly contributed to multiple publications, directly to one published work (chapter 4), and two works in preparation (including chapter 3).

Chapter 3: A general metric for the similarity of stochastic and deterministic system dynamics

Colin Shea-Blymyer c0lin@vt.edu
Department of Computer Science
Virginia Tech
Blacksburg, VA 24060, USA

Subhradeep Roy sdroy@vt.edu
Dept. of Philosophy (0126)
229 Major Williams Hall
Virginia Tech
220 Stanger St.
Blacksburg, VA 24060, USA

Benjamin C. Jantzen bjantzen@vt.edu
Dept. of Philosophy (0126)
229 Major Williams Hall
Virginia Tech
220 Stanger St.
Blacksburg, VA 24060, USA

Editor:

Abstract

abstract
Keywords: keywords

1. Introduction

A suite of problems of importance to those looking to model, predict, or control the behavior of complex dynamical systems can be viewed as aspects of the problem of determining whether two systems – real or simulated – are dynamically similar. For example, for system identification it is often important to know at the outset whether the system in question can be described with a linear model (CITATION). For contexts where acquiring data points in a time series is expensive – such as community ecology – it's important to know the effective order of the dynamical process of interest in order to know how long of a time series will be needed to fit a reliable model (CITATION). Relatedly, the degree of nonlinearity and chaos exhibited by a system is essential for managing error in system control or prediction (CITATION). Each of these problems – identifying effective order, nonlinearity, and chaos – can be seen in terms of dynamical similarity. Whether a system is strongly nonlinear is equivalent to asking whether and to what degree it is dynamically similar to a strongly nonlinear system and mutatis mutandis for effective order and chaos.

Another prominent problem in the literatures of a number of fields (CITATIONS) is that of change detection. It is often of interest to know when the causal structure of a

dynamical system has changed, as for instance when an ecosystem (or climate) has been perturbed by external forcing, or when a mechanical component has begun to fail. This too can be seen as a problem of dynamical similarity, in this case determining whether one and the same system at a later time is dynamically dissimilar to itself at an earlier time. Importantly, this is not the same as asking whether, say, the statistics of the later time evolution of the system differ from those of an earlier portion. Such differences may simply reflect the fact that complex system has moved into a new region of its phase space. While such superficial proxies may be useful in spotting changes, what is at issue is the causal structure determining the time evolution. In general, we want to know whether the system will continue to evolve and respond to perturbations in the same way as before. In other words, we want to know whether the dynamics have changed. All of these problems of dynamical similarity are exacerbated by sampling noise in deterministic systems, and become significantly more challenging to address for stochastic systems.

We here describe a general metric of dynamical similarity that is well-posed for both stochastic and deterministic dynamical systems and which is sensitive to the effective order of dynamics, the degree of nonlinearity, and the presence of chaos. Importantly, this metric can be informative of these dynamical features even when only partial information about the dynamical state of a system is tracked, or a lossy function of the dynamical variables is observed. We introduce a variety of algorithms to show that this metric can be learned in a range of contexts, from situations in which one has full control of the dynamical system and complete dynamical information to situations in which only partial information is available for passively observed systems. We also demonstrate how this metric can be applied to the problem of change detection in this range of circumstances, and how it can be deployed to rapidly map out the varieties of dynamical behavior as a function of parameter space for a given dynamical model.

2. Related work

Our algorithm's determination of dynamical similarity is sensitive to nonlinearity, chaos, and order of a system, among other features. These three structural qualities have rich histories of research in identification.

2.1 Detecting nonlinearity

Model free approaches like transfer entropy [28] and mutual information, built on information theoretic concept, have been proposed as effective methods to detect the nature of coupling in dynamical systems. In [19], a transfer entropy based metric is proposed to assess the degree of nonlinearity present in a systems dynamics. The metric is further compared with time-delayed mutual information. The transfer entropy is found to be a more effective indicator to detect the nonlinearity when compared to the mutual information for both simulated and experimental data. A test for nonlinearity for multivariate time series data has been proposed in [21], which combines an information-theoretic (redundancy) approach with the surrogate data technique. The univariate implementation of the method is described in details in [20]. The basic idea associated with the surrogate-data-based nonlinearity test is to compute a nonlinear statistic from the given data and from an ensemble of realizations of a linear stochastic process, which captures the "linear properties" of the

data. If the comparison with the computed statistics reflect a significant difference, the inference is drawn that the data are not generated by a linear process; otherwise the null hypothesis is accepted, which means a linear model fully explains the data. In [13] the use of a different test statistic, called the number of missing ordinal patterns, is used to test nonlinearity in both model and experimental data. The idea is built on the previously introduced concept of ordinal patterns in [2]. The challenge of detecting nonlinearity from short and noisy data has been addressed in a recent work in [34], which uses the concept of permutation entropy that contains information about the temporal structure associated with the underlying dynamics of a time series.

2.2 Determining effective order

System order identification techniques that can be applied to the more challenging classes of systems that our algorithm can be applied to (e.g. non-linear, stochastic systems) are well documented [11]. One of the earliest approaches to determining the order of nonlinear, autoregressive systems relied on the Akaike information criterion (AIC) of a candidate model [30]. These methods regress toward an optimal model order, requiring further parameter estimation of the model itself. The method of false nearest neighbors (FNN) embeds the available data in a low-dimensional subspace, then compares the distance between neighbors in the subspace to their distance in prediction-space. If the distance in prediction-space is high, then they are considered false neighbors, and if there are many false neighbors, then the dimension of the subspace is increased until nearly no false neighbors are present [23]. This method, and similar methods using Lipschitz numbers [3], are model-free, but still require an iterative process of regressing to an optimal order. Subspace methods test the smallest singular value of the crucial matrix of a system of data, increasing the rank of the matrix until the test is passed [3]. Again, these methods are model free, but rely on a sequence of tests. Further, while many of the above methods have been developed for classes of models used to represent stochastic systems, their performance in the presence of noise has been questioned [11].

2.3 Determining the degree of chaos in dynamical systems

In the literature, a variety of approaches have been proposed to study chaotic time-series data such as correlation dimension analysis [22, 7], estimation of the largest Lyapunov exponent [9], reconstruction of the dynamical system [26], and controlling chaos [10, 27]. The correlation dimension analysis is based on the phase space reconstruction and quantification of the strangeness of the attractors. The main limitation of the method is that it is a data-hungry process and sensitive to the presence of measurement noise in the data. An alternative approach has been proposed in [29] to analyze short time series data called nonlinear forecasting, and utilizes a nearest-neighbor technique to predict the future state of a dynamical system using the history of that time-series. A comparison between these two methods in [33] demonstrate that nonlinear forecasting is more efficient and robust than the correlation dimension for chaotic time series analysis in a noisy system.

2.4 Dynamical similarity and change

Time series similarity measures are important in classification and change detection, for example to characterize ecosystem dynamics, or explore genomes. An overview and quantitative comparison of the similarity measures has been provided in [16]. The most common measures used in literature include a number of distance measures (e.g. Manhattan, Euclidean Minkowski and Mahalanobis, correlation), and transformation approaches (Principal Component Analysis, and Fourier based similarities). Manhattan [4] and Euclidean distance [6] measures quantify the absolute magnitude and the Euclidean distance of the difference between time series, respectively, whereas Mahalanobis distance [5] measure additionally accounts for nonstationarity of variance and temporal cross-correlation. Correlation distance [6] detects the degree of linear relationship between time series. Distance measures based on principal component analysis [18] quantifies the difference between times principal components. Additionally, Fourier based similarities [8, 15] detect the shape similarity based on Fourier transform derived amplitude and phase differences. Finally, these metrics can be used for finding patterns in data [1], and the detection of anomalies [32]. These similarity measures find applications in several time series clustering and change detection.

2.5 Dynamical kinds

Effective order, nonlinearity, and chaos are each aspects of the causal structure of a dynamical system. The existing tests contrived to assess these aspects consider a target system in isolation; the details of a particular system's behavior are used to determine, e.g., the effective order of its dynamics, and only after the fact is it recognized that two systems share causal features in common. The theory of dynamical kinds offers an alternative approach: by determining that two systems are of the same or different dynamical kinds, we learn whether they share any of these dynamical properties. Information about any one can then be transferred to the class.

Dynamical kinds were first defined in CITATION. The dynamical kinds theory partitions the space of dynamical systems into equivalence classes – dynamical kinds – on the basis of dynamical symmetry. A dynamical symmetry is an intervention – an externally induced change in the values of the some of the dynamical variables of a system – that commutes with any intervention on a chosen index variable. For dynamical systems for which we are explicitly concerned with temporal dynamics, time is generally treated as the index variable. In that case, a dynamical symmetry (with respect to time) is defined as follows (CITATION):

Definition 1 (Dynamical symmetry with respect to time) *Let t be the variable representing time, and let V be a set of additional dynamical variables such that $t \notin V$ and Ω is the space of states that can be jointly realized by the variables in V. Let $\sigma : \Omega \to \Omega$ be an intervention on the variables in $Int \subseteq V$, and Λ_{t_0,t_1} the time-evolution operator that advances the state of the system from t_0 to t_1. The transformation σ is a dynamical symmetry with respect to time if and only if for all intervals Δt and initial states $\omega_i \in \Omega$, the final state of the system $\tilde{\omega_f} \in \Omega$ is the same whether σ is applied at some time t_0 and the*

system evolved until $t_0 + \Delta t$, or the system first allowed to evolve from t_0 to $t_0 + \Delta t$ and then σ is applied. This property is represented by the following commutation diagram:

$$\begin{array}{ccc} \omega_i & \xrightarrow{\sigma} & \tilde{\omega_i} \\ {\scriptstyle \Lambda_{t_0,t_0+\Delta}}\downarrow & & \downarrow{\scriptstyle \Lambda_{t_0,t_0+\Delta}} \\ \omega_f & \xrightarrow{\sigma} & \tilde{\omega_f} \end{array} \tag{1}$$

To reiterate a trivial and oft-cited example, for an exponentially growing population of bacteria, for which the population x changes according to $dx/dt = rx$, any transformation that scales the population by a positive constant k is a dynamical symmetry of the system – scaling by k and then allowing the bacteria to grow for Δt versus growing for Δt and then scaling the resulting population by k results in the same final population.

The composition of any two dynamical symmetries (by successive intervention on the variables of a system) is itself a dynamical symmetry, and for any given system, its dynamical symmetries typically exhibit non-trivial algebraic structure under composition. It is the collection of symmetries and their structure under composition that characterizes a dynamical kind (CITATION).

CITATION provided a method for determining whether or not two physical systems with continuous deterministic dynamics belong to the same dynamical kind directly from time series data without first constructing dynamical models of either system. The method exploits two facts: (i) that a necessary condition for two systems to belong to the same kind is that they share all of their dynamical symmetries, and (ii) that for every state-determined system in the same dynamical kind, there exactly one symmetry maps the unique trajectory passing through one point in phase space, $\vec{x}$, to the trajectory passing through another point $\tilde{\vec{x}}$. That is, even if Systems A and B exhibit different trajectories given initial conditions $\vec{x}$ or $\tilde{\vec{x}}$, the symmetry connecting these two trajectories for System A must be the same as that for System B if they belong to the same kind. Numerically estimating and then comparing these symmetries from time series data thus provides a sensitive test for the sameness of dynamical kind that is robust under significant sampling noise (CITATION).

3. Dynamical distance

3.1 Stochastic dynamical kinds

I was suggested by CITATION that that the definition of dynamical similarity should be generalized to accommodate stochastic dynamics as follows:

Definition 2 (Dynamical symmetry) *Let V be a set of random variables, Ω the set of states that can be jointly realized by the variables in V, and Γ the space of probability distributions over Ω. Let $\sigma : \Gamma \to \Gamma$ be an intervention on the variables in $Int \subset V$. The transformation σ is a dynamical symmetry with respect to some index variable $X \in V - Int$ if and only if σ has the following property: for all initial joint distributions $\gamma_i \in \Gamma$ and marginal probability distributions f and g, the final joint probability distribution over V, $\tilde{\gamma_f} \in \Gamma$, is the same whether σ is applied when the marginal distribution over X is given by $p_x(x) = f(x)$ and then an intervention $\Lambda_{f(x),g(x)} : \Gamma \to \Gamma$ on X makes it such that $p_x(x) = g(x)$, or the intervention on X is applied first, changing its marginal distribution*

from $f(x)$ to $g(x)$, and then σ is applied. This property is represented in the following commutation diagram:

$$\begin{array}{ccc} \gamma_i & \xrightarrow{\sigma} & \tilde{\gamma}_i \\ {\scriptstyle \Lambda_{f(x),g(x)}}\downarrow & & \downarrow{\scriptstyle \Lambda_{f(x),g(x)}} \\ \gamma_f & \xrightarrow{\sigma} & \tilde{\gamma}_f \end{array} \tag{2}$$

We accept this definition, and use it to develop a natural metric over dynamical systems, as we develop below.

3.2 Constructing a metric of dynamical similarity

Consider a dynamical system described by n variables, some of which may be derivatives or time-lagged versions of other variables. The states of such a system can be represented by n-dimensional vectors $\vec{x} \in \mathbf{X}$. Let $\mathbf{F}$ be the space of probability density functions over $\mathbf{X}$. We say that such a system is *stochastically state-determined* (SSD) if and only if the (possibly stochastic) dynamics of the system is such that the probability density $\rho_i \in \mathbf{F}$ over possible states at time $t_i > t_1$ is completely determined by the density $\rho_1 \in \mathbf{F}$ over states at t_1. In other words, for an SSD system there exists a map, $\Lambda_{t_1,t_i} : \mathbf{F} \to \mathbf{F}$ that, for any $t_i > t_1$ advances the probability density over states of the system from $\rho_1(\vec{x})$ to $\rho_i(\vec{x})$. Any function, $\sigma : \mathbf{F} \to \mathbf{F}$ that commutes with this map is a dynamical symmetry of the system. More precisely, σ is a dynamical symmetry of a system if, for every $t_i > t_1$

$$\sigma \circ \Lambda_{t_1,t_i} \circ \rho_1(\vec{x}) = \Lambda_{t_1,t_i} \circ \sigma \circ \rho_1(\vec{x}) \tag{3}$$

where $\sigma \circ \Lambda_{t_1,t_i} \circ \rho_1$ is the probability distribution that results from successive application of the maps Λ and then σ to the original distribution, ρ_1 (and similarly for $\Lambda_{t_1,t_i} \circ \sigma \circ \rho_1$). This commutation property is represented in Figure 1.

Figure 1: caption

$$\begin{array}{ccc} \tilde{\rho}_1(\vec{x}) = \sigma \circ \rho_1(\vec{x}) & \xrightarrow{\Lambda_{t_1,t_i}} & \tilde{\rho}_i(\vec{x}) = \sigma \circ \Lambda_{t_1,t_i} \circ \rho_1(\vec{x}) = \Lambda_{t_1,t_i} \circ \sigma \circ \rho_1(\vec{x}) \\ \uparrow{\scriptstyle \sigma} & & \uparrow{\scriptstyle \sigma} \\ \rho_1(\vec{x}) & \xrightarrow{\Lambda_{t_1,t_i}} & \rho_i(\vec{x}) = \Lambda_{t_1,t_i} \circ \rho(\vec{x}) \end{array}$$

Consider two probability densities connected by one such dynamical symmetry at time t_1: ρ_1 and $\sigma \circ \rho_1$. We focus on a particular probability density function that relates the time evolution of these two initial densities. Specifically, if one of n distinct times, $t_1, \ldots, t_n$, in the evolution of the system is selected at random from a uniform distribution over the n possibilities, we seek the probability density that the time-evolution beginning with an initial state dictated by ρ_1 presents a system state $\vec{x}$ and the time-evolution beginning with

an initial density $\sigma \circ \rho_1$ at t_i presents a system state $\tilde{\vec{x}}$. This joint density is given by:

$$\begin{aligned}\rho^*(\vec{x}, \tilde{\vec{x}}) &= \sum_i (\text{prob. that } \rho = \rho_i)(\text{prob. of } \vec{x} \text{ given } \rho = \rho_i)(\text{prob. of } \tilde{\vec{x}} \text{ given } \rho = \sigma \circ \rho_i) \\ &= \sum_i \frac{1}{n} \rho_i(\vec{x}) \sigma \circ \rho_i(\tilde{\vec{x}}). \end{aligned} \tag{4}$$

Denote the corresponding cumulative distribution by cdf^*. This distribution for any given dynamical system is shaped by its dynamical structure. In order to compare dynamical structure, our approach is to compare cdf^*. A suitable metric for doing so is the energy distance [citations]. The energy distance for two cumulative distributions, cdf^A and cdf^B is defined as follows:

$$D_E^2(cdf^A, cdf^B) = 2\mathbb{E}\left[||X^A - X^B||\right] + \mathbb{E}\left[||X^A - X^{A'}||\right] - \mathbb{E}\left[||X^B - X^{B'}||\right]. \tag{5}$$

The square root of the right hand side of Equation 5, D_E, is a proper metric, and is zero if and only if $cdf^A = cdf^B$. For two systems A and B, if at each time index i, $\rho_i^A(\vec{x}) = \rho_i^B(\vec{x})$, then the difference between $\rho^{A*}(\vec{x}, \tilde{\vec{x}})$ and $\rho^{B*}(\vec{x}, \tilde{\vec{x}})$ and thus any difference between cdf^A and cdf^B, must be due entirely to the action of the respective symmetries, σ^A and σ^B. In that case, measuring the energy distance $D_E(cdf^A, cdf^B)$ provides a quantitative comparison of the symmetries of Systems A and B, and thus of the underlying casual structure that gives rise to them. We call $D_E(cdf^A, cdf^B)$ the *dynamical distance* between A and B.

3.3 Clipping

When the dynamic properties of a system are unknown, it can be difficult to predict when one system reaches equilibrium. If a system reaches equilibrium before one it's being compared to, the vestigial data can lead to a decrease in the accuracy of the dynamical distance. In order to mitigate this effect, we clip either $p^A(\vec{x})$ or $p^B(\vec{x})$ to a length that minimizes the energy distance between the two. This method is expressed in algorithm 1.

This algorithm can also be used to resample the system that produced the data to be clipped in order to maintain a higher sample density, provided we have sufficient control over the system.

The objective of clipping (to select a subsequence of one time-series that minimizes its energy distance from another) is similar to the objective of dynamic time warping (to select a mapping of the indices of one time-series to those of another that minimizes the sum of differences between the index pairs) [25]. Further, dynamic time warping (DTW) preserves the initial conditions of a time-series - an important caveat for some of our methods. While we could adapt DTW as a replacement clipping algorithm, we're uncertain how the warping itself would impact the computation of dynamical distance, and so have elected to use the presented algorithm.

4. Properties of dynamical distance

In order to better understand how our algorithm performed in detecting changes to features of dynamical kinds, we performed a battery of tests using the Lotka-Volterra and Kuramoto systems of equations.

Algorithm 1 Calculate clipping location of one curve minimizing energy distance to another.

Require: $p^A(\vec{x}), p^B(\vec{x}), l_{min}, S$

```
Δ ← (len(p^A(x⃗)) − l_min)/S
distance_min ← inf
i_min ← l_min
for i ← l_min to len(p^A(x⃗)) step Δ do
    δ ← len(p^B(x⃗))/i + 1
    distances ← ∅
    for j ← 0 to i do
        k ← ⌊j * δ⌋
        distances_j ← EnergyDistance(p^A(x⃗)_j, p^B(x⃗)_k)
    end for
    distance ← mean(distances)
    if distance < distance_min then
        distance_min ← distance
        i_min ← i
    end if
end for
return distance_min, i_min
```

4.1 Difference in dynamical kind

Difference in dynamical kind can be caused by features of the system like nonlinearity, order, and chaos (as discussed below), but can also be present in subtler transformations through a system's parameter space. In the Lotka-Volterra predator-prey equations [31], a change in the carrying capacity leads to a change in dynamical kind, while uniformly scaling all species' growth rates does not [12]. These equations are generalized as:

$$\frac{dx_i}{dt} = r_i x_i \Big(1 - \frac{\sum_{j=0}^{n} a_{ij} x_j}{k_i}\Big), i = 0...n \tag{6}$$

where x_i denotes the population size of the ith species, k_i denotes its carrying capacity, r_i denotes the growth rate of the species, and a_{ij} is the interaction coefficient of species j on species i.

To show that our metric reflects this variety of dynamical change created a test that compares the change in distance between two systems caused by a multiplicative increase of the carrying capacities of one system being compared to the change caused by the same increase to the growth rates of another system. In this test, a two-species system is simulated from $t = 0$ to 15 in order to focus on the transition periods of the systems. One system remains unchanged in all comparisons, and is referred to as system A. The other two systems begin the test identical to system A. System B's carrying capacities are scaled from $1 * k^A$ to $10 * k^A$, while system C's growth rates are scaled from $1 * r^A$ to $10 * r^A$. If our metric does capture the difference in kind caused by changing a system's k values, then we expect to see the distances between systems A and C to be much larger than those between systems A and B.

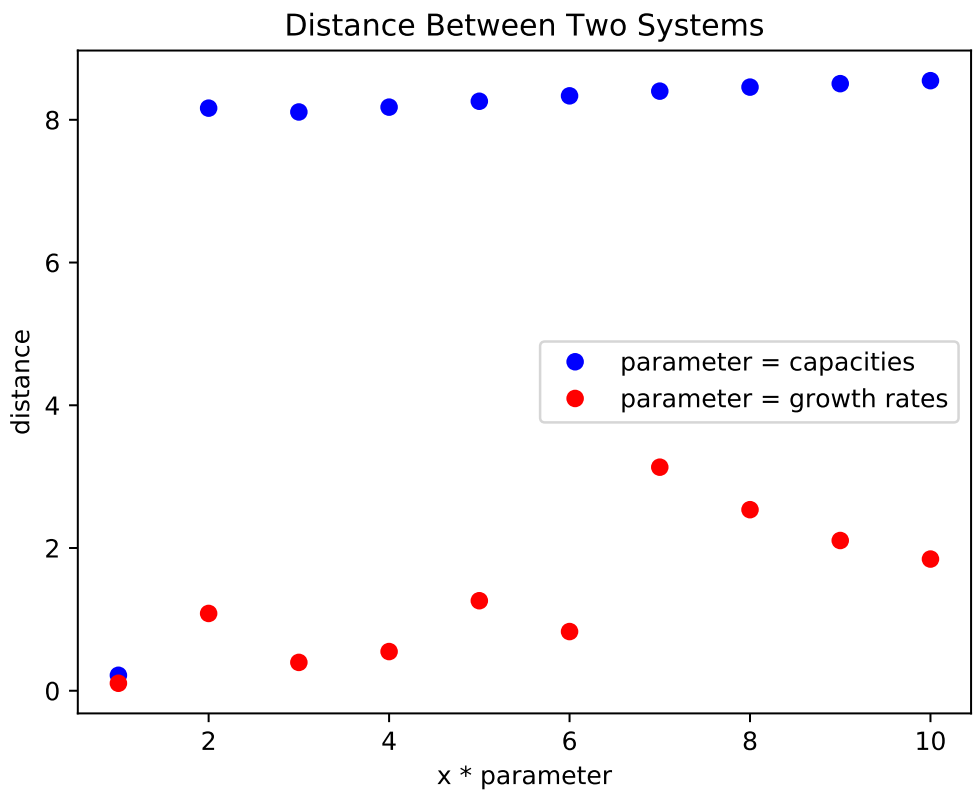

Figure 2: Dynamical distance between systems A and B (blue), and between systems A and C (red) as B's carrying capacities and C's growth rates are scaled up.

As seen in figure 2, the distances between systems A and B quickly grow to be more than 4 times the distances between systems A and C. This suggests that our metric is properly reflecting the impact of changing a system's carrying capacity on that system's dynamical kind.

4.2 Nonlinearity

Our distance metric can be said to be sensitive to changes in a system's degree of linearity if the distance grows as the change in linearity grows. To demonstrate this, we contrived a modification of the Lotka-Volterra equations (equation 6) that allows us to modulate the system's degree of linearity:

$$\frac{dx_i}{dt} = r_i x_i \Big(1 - \frac{\sum_{j=0}^{n} a_{ij} l x_j}{k_i}\Big), i = 0...n \tag{7}$$

where l scales the linearity of the system. For this test, a two-species system is simulated, and all species are provided with identical carrying capacities (k). System B received growth rates (r) 1.5 times larger than those of System A - a difference that does not affect dynamical kind. System A's value of l was set to 1, and was compared to system B, whose value of l ranged from 0 to 1.8. These differential equations were solved from $t = 0$ to 15 in order to capture the behavior of the linear system, while remaining largely in the transitional phases of the nonlinear systems. Each system's untransformed initial population is 5 members, and each transformed initial population is 8.

The results of this test, as seen in figure 3, show a minimum in dynamical distance where the linearity factor is 1, and consequently, both systems are of the same kind. The distance between the two systems seems to increase to a plateau as the linearity factor approaches 0, increasing slightly at zero. This increase at zero may be symptomatic of the complete uncoupling of the populations in the system. As the value of l increases to the right of 1, we see similarly sharp increase in the distances between system A and system B. This intimates the fact that a large l value overwhelms the effects of the natural growth rate of each species, greatly changing the dynamics of the system.

4.3 Effective order

To show that our distance metric is sensitive to changes in the effective order of a system, we constructed a test wherein a second-order version of the Lotka-Volterra equations (equation 6) is outfitted with a scaling variable (ω) on its second-order term:

$$\frac{dx_i}{dt} = y_i \tag{8}$$

$$\frac{dy_i}{dt} = \omega\Big(r_i x_i \Big(1 - \frac{\sum_{j=0}^{n} a_{ij} x_j}{k_i}\Big) - y_i\Big), i = 0...n \tag{9}$$

As ω approaches infinity, the system approaches first-order dynamics. In this test, the parameters of systems A and B are the same as the systems in 4.2, but for the absence of l and the presence of ω. System A, whose ω value set to 1, was compared to system B as its ω value grew exponentially. These systems were also run from $t = 0$ to 15 for our tests.

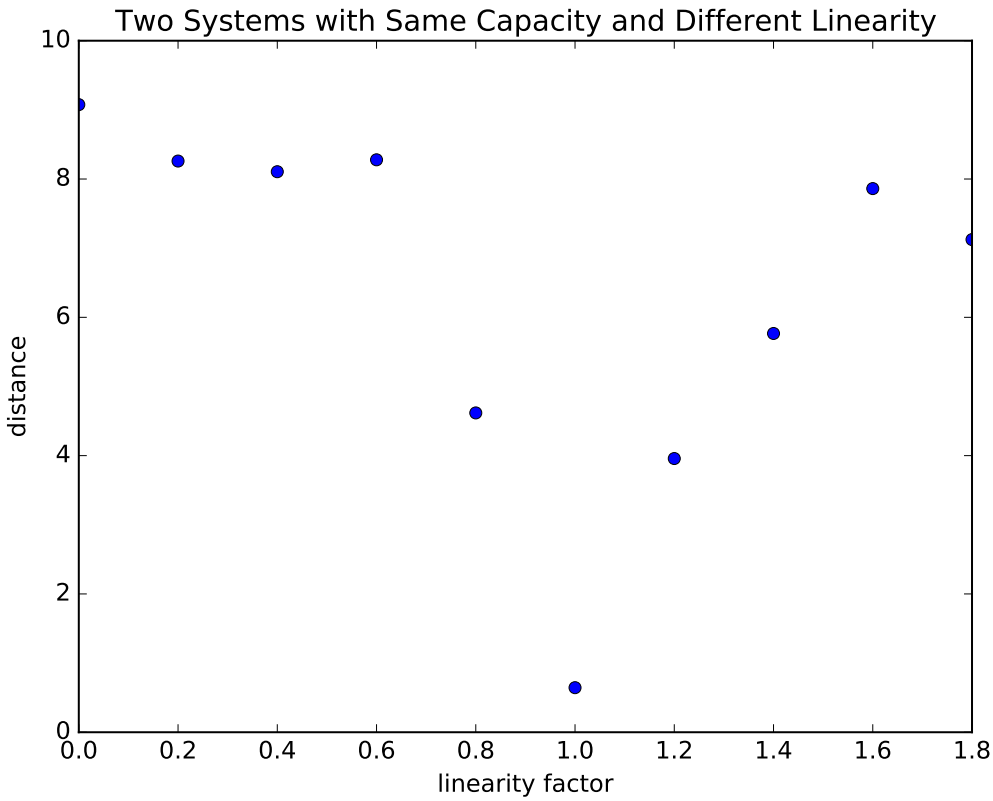

Figure 3: Dynamical distance of system A (linearity factor = 1.00) from system B as linearity factor of system B is changed.

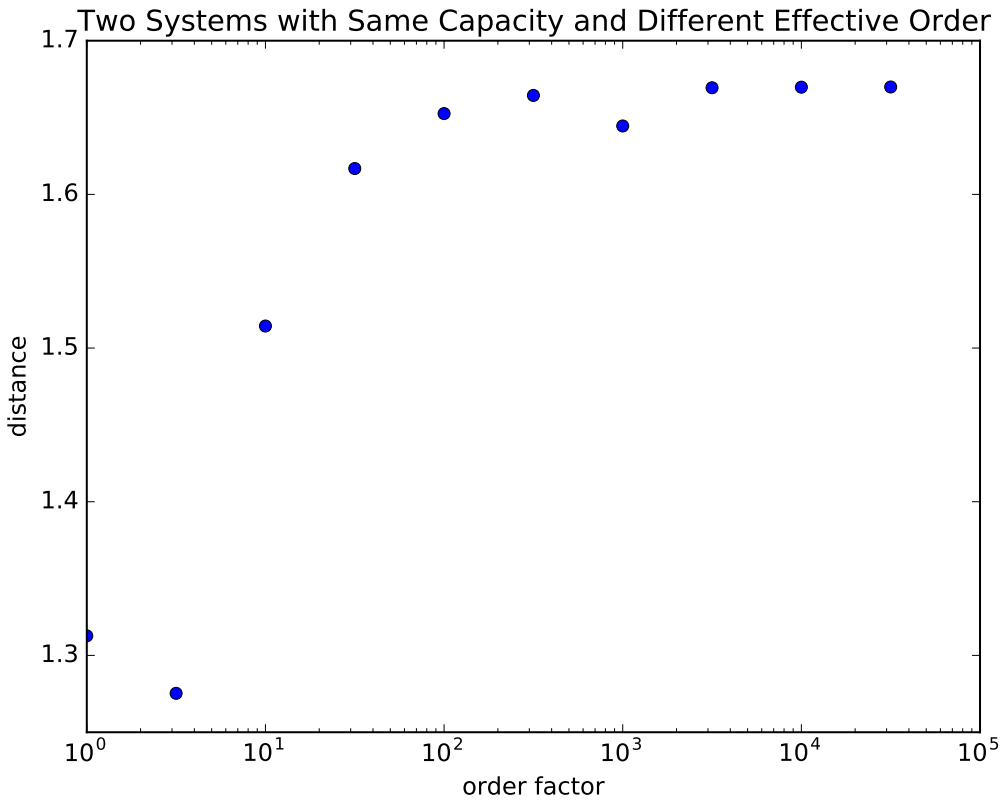

Figure 4: Dynamical distance of system A (order factor = 1.00) from system B as order factor of system B is changed.

We expect to see that systems with similar effective order are close together. This is shown in figure 4 - the distance between the two systems is smallest when the order factors are close. Similarly, we expect that systems of different order will be farther apart. This is evidenced in the figure - the distance between the two systems grows with the order factor, leveling out as the large ω value causes system B to approximate first-order dynamics.

4.4 Chaos

Testing the sensitivity of our distance metric to the presence of chaos requires us to compare a known chaotic system to similarly parameterized systems while changing only one of the parameters. We again expect two systems with similar parameters to be similar in dynamics, but in chaotic regimes, we expect the systems to diverge more quickly.

We use the chaotic four-species Lotka-Volterra system referenced as (R_1, A_1) in [24] as system A for this test, with carrying capacities of 100. System B differs from system A in that the $a_{0,1}$ term varies from 2.319 to 2.519, while the corresponding value for system A is kept at 2.419. To provide context for the behavior of the distance metric in this space of systems, we performed the same test on a non-chaotic system for which the terms of the interaction matrix differ from those in the chaotic system by less than 0.3. These systems were run for longer (to $t = 100$) than previous tests in order to capture more of the chaotic behavior of the system. More comparisons were performed, as well, to allow us to see the complex structure of their similarities.

The plot in figure 5 shows that, as expected, the smallest difference for both the chaotic and the non-chaotic sets of systems is when there is no difference in their interaction terms. The behavior of the distance between chaotic systems as compared to non-chaotic systems is very interesting. Not only does the distance between a chaotic system and its neighbors in parameter space increase more rapidly than those between non-chaotic systems, the character of the two behaviors is patently different. While the behavior in dynamical distances observed in this test on the non-chaotic systems can be considered a smooth, symmetric curve towards greater distances as their parameters grow farther apart, the corresponding behavior for a chaotic system is neither smooth, nor symmetric. Indeed, the curve produced when one system is chaotic seems to be non-monotonic, and non-differentiable. Beyond showing that our distance metric is sensitive to the presence of chaos, the rich structure of how these distances grow across parameter space suggests that the metric retains significant information about the structure of the chaotic behavior.

In section 2.3, it is mentioned that some alternative methods for detecting chaos struggle to perform well in the presence of noise. To address this, we performed the above test with the addition of measurement noise ($\mathcal{N}(\mu = 0, \sigma = 2)$). The results are recorded in figure 6, and show that, even in the presence of significant noise, much of the structure of the distance's behavior remains in tact.

4.4.1 Chaos and the l_2 Norm

We further demonstrate our algorithm's sensitivity to chaos by selecting a system between two chaotic points in the Lotka-Volterra parameter space corresponding to (R_1, A_1) and (R_2, A_2) in [24]. That system becomes system A, and is compared to systems that approach (R_1, A_1) from system A's position in parameter space. We also compare the performance

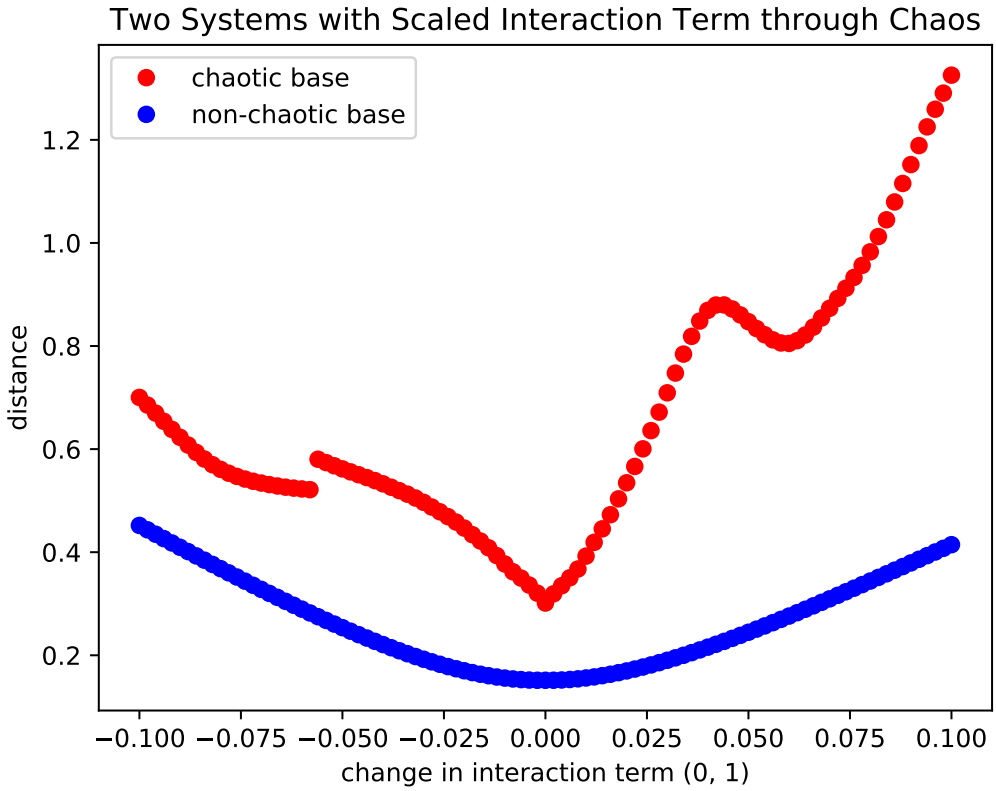

Figure 5: Dynamical distance of system A (change of interaction term = 0) from system B as one interaction term of system B is changed. The case where system A is chaotic is presented in red, and where system A is not chaotic in blue.

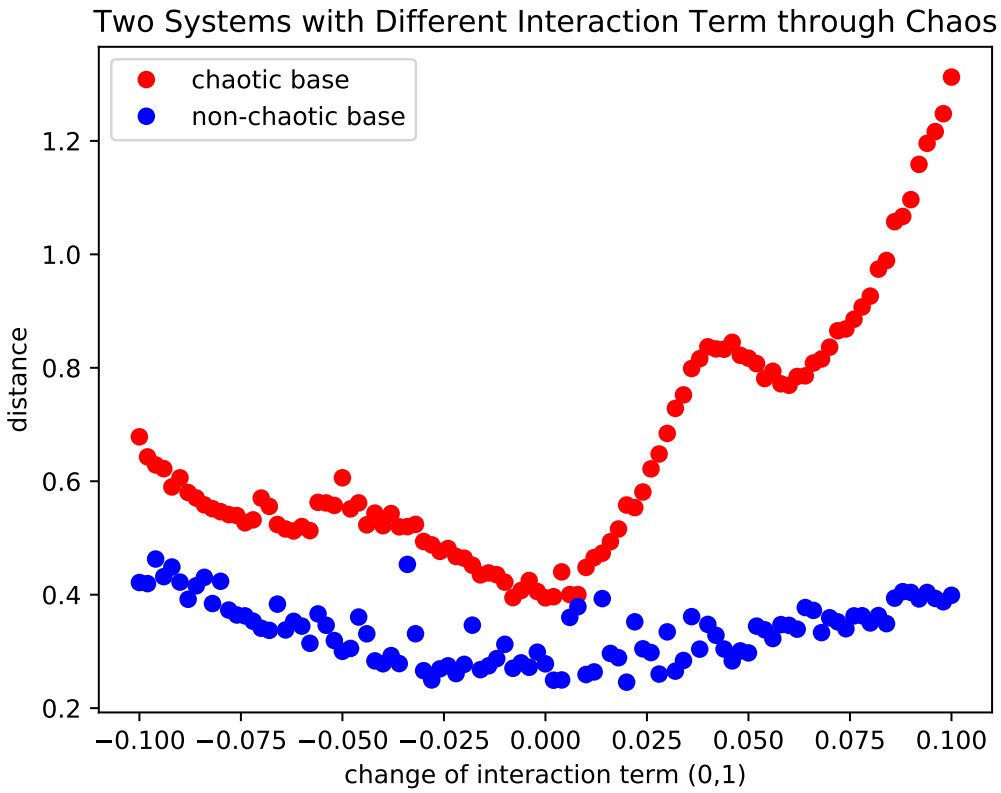

Figure 6: Dynamical distance of system A from system B with the introduction of measurement noise.

of our algorithm here to the performance of the l_2 norm of the difference between the untransformed trajectories of system A and the other systems.

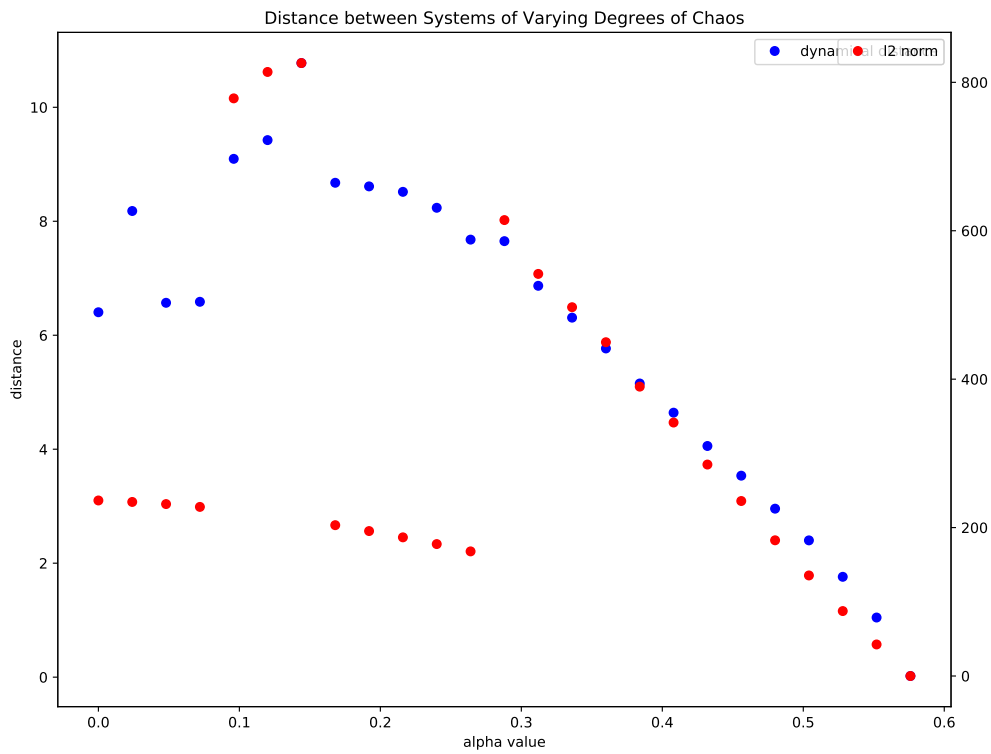

Figure 7: Dynamical distance of system A from system B (left y-axis; blue) and l_2 norm of the difference between the untransformed trajectories of systems A and B (right y-axis; red).

This result shows that both distance metrics pick up on the change of kind due to shifts in the system's interaction matrix, however, when the non-chaotic system A is compared to the chaotic system B at $\alpha = 0$, the distance given by our algorithm is relative to twice the difference determined by the simple l_2 method. It's notable, though, that the simpler l_2 method does capture many of the more linear dynamics of the system.

4.5 Model Mapping: Lotka-Volterra

To explore how distances produced by our algorithm behave outside of isolation to a single type of change in dynamical kind, we reproduced the parameter space researched in [24]. This two-dimensional slice of the 24-dimensional parameter space of a four-species

Lotka-Volterra system is constructed such that its corners lie in regions that exhibit chaotic behavior.

For our experiments, we use the same parameter space that the [24] paper used:

$$(R, A) = (R_1, A_1) + \alpha(R_2 - R_1, A_2 - A_1) + \beta(R_3 - R_1, A_3 - A_1)$$

where R_x corresponds to the growth vector (containing r_i), and A_x corresponds to the interaction matrices (containing a_{ij}). The linear combination of (R_x, A_x) is a mixture of (R_1, A_1), (R_2, A_2), and (R_3, A_3).

Samples of the system are collected from regularly arranged points in the selected region. At each such arrangement of system parameters, the simulation is run for two different sets of initial species populations, forming two time-series per set of system parameters - one untransformed representative, and one transformed.

We calculate the (R, A) values for α from 0 to 1.2 and for β from -0.2 to 1.2. For our study we do a sweep over this α and β range with uniform step sizes.

For all systems we sample, we set the carrying capacity of each species to 100, set the growth rates of each species to 5 for the untransformed systems, and to 8 for the transformed systems. Each system is simulated for 4000 time-steps, with a Δt of 0.5. An untransformed system and a transformed system is sampled for each coordinate of 50 $\alpha[0, 1.2]$ values and 50 $\beta[-0.2, 1.2]$ values. We then compute pairwise comparisons for each system, and a distance matrix is produced. For visualization purposes, this distance matrix is reduced to three dimensions using Multi-Dimensional Scaling (MDS), and displayed in color space, where the x and y axes correspond to the α and β values of the systems presented, and the similarity in color of each system is representative of the dynamic similarity between them.

The results from [24] can be seen in figure 8, where the color of each pixel represents the maximum number of coexisting species at the end of a large number of simulations to 4000 time-steps. This demonstrates an attractive separation of the system's parameter space, while maintaining the presence of chaotic behavior. The results from our scan of the parameter space are shown in figure 9. The similarity in color between two pixels in our figure represents the closeness of the three-dimensional embedding of the corresponding systems' distance matrix entries.

While our figure displays a significant amount of structure in the space, few features of its representation reflect features displayed in figure 8. Most noticeably, the chaotic region at (0, 1) is nearly absent, save for some pips of outlying color. Curiously, though we've shown that our distance metric is sensitive to the presence of chaos, and that moving through a region of chaos presents swift changes in distance, the chaotic regions in our figure are solidly colored. This might indicate that the distances incurred by moving through a chaotic regime are smaller than those due to moving out of a chaotic regime, i.e. chaotic systems are more similar to other chaotic systems (though still noticeably different) than to non-chaotic systems. Similarly, while our algorithm catches the upper boundary of figure 8's large 2-species region, no other boundaries present there are reflected in figure 9. Instead, whole, new regions are identified; some with surprisingly sharp boundaries.

Using a low-dimensional embedding, constructed without normalization, we have discovered well-structured regions of similar dynamical structure, demonstrating the application of our algorithms as a novel method for exploring a system's behavior.

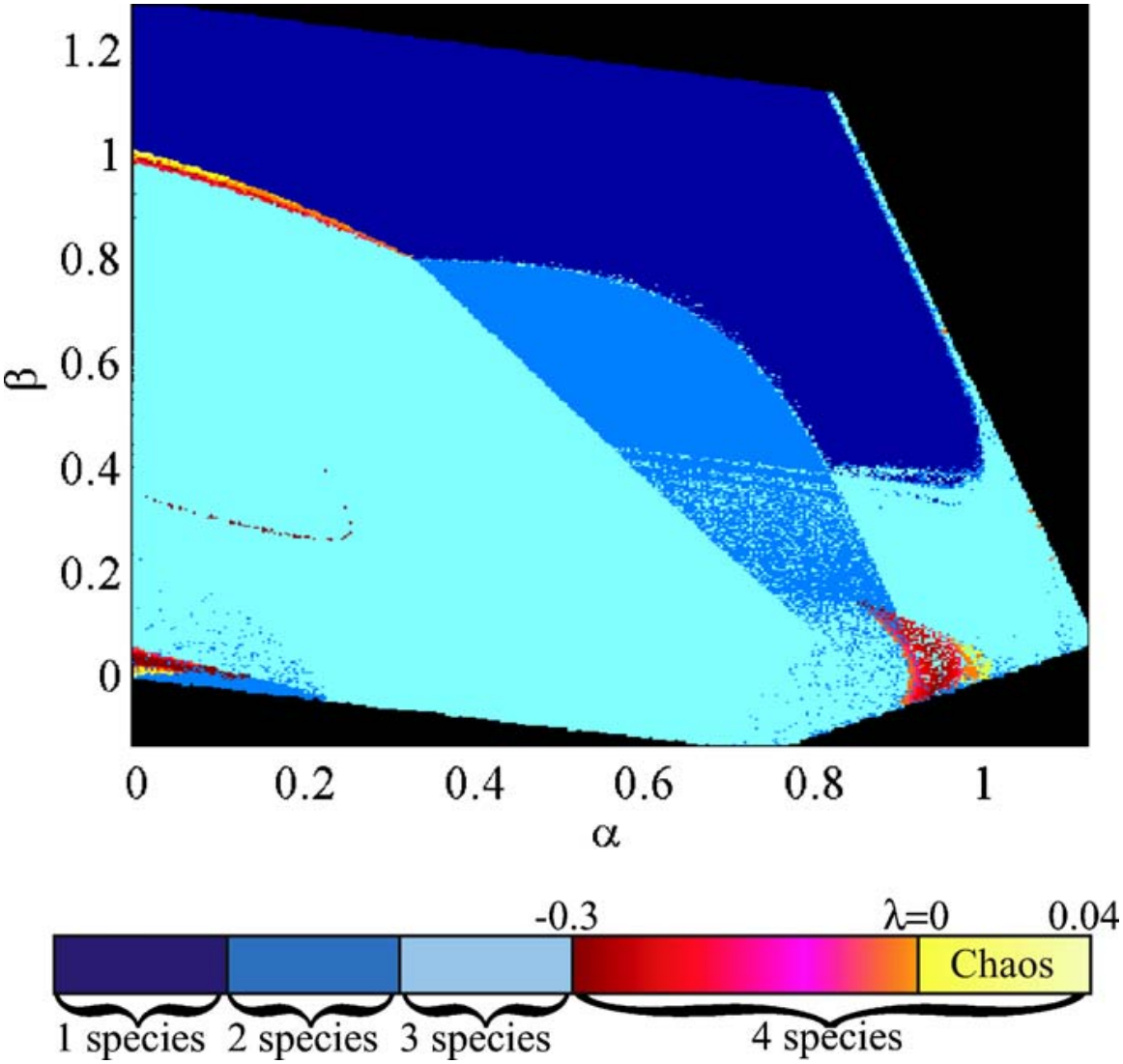

Figure 8: Exploration of chaos and species survival rates in a section of parameter space, as presented in [24]. The coordinates (0,0), (1,0), and (0,1) correspond to identified regions of chaos. The color of each pixel represents the maximum number of surviving species (or maximum Lyapunov exponent, in case of chaos) of the system in that pixel. Black regions correspond to non-competitive systems (i.e. negative values are present in A).

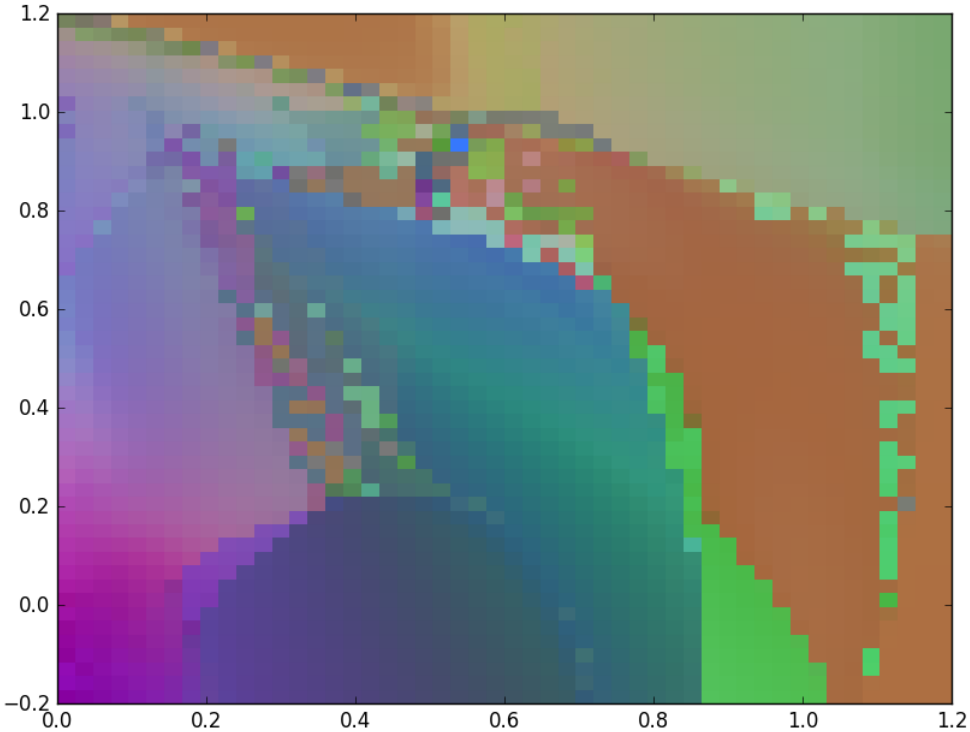

Figure 9: Exploration of the same parameter space as figure 8. The color of each pixel represents a three-dimensional embedding of that system's position in the pairwise dynamical distance matrix of the systems in the space. The x-axis is equivalent to figure 8's α axis, and the y-axis equivalent to the β axis.

4.6 Change detection

A natural extension of a distance metric is its use for anomaly detection. By computing the dynamical distance between each subsequence and its neighbors in a time-series, we can expect to find peaks of dissimilarity when the target subsequence is of one dynamical kind, but its neighbor is of a different kind. To test this straightforward approach, we simulated a three-member Kuramoto system [14], generalized as:

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{N}\sum_{j=0}^{N} sin(\theta_j - \theta_i), i = 0...N \tag{10}$$

where $theta_i$ is the angle of the ith oscillator, ω_i is that oscillator's natural frequency, N is number of oscillators in the system, and K, the coupling coefficient, determines how much the difference in angle between oscillators affects an oscillator's future state.

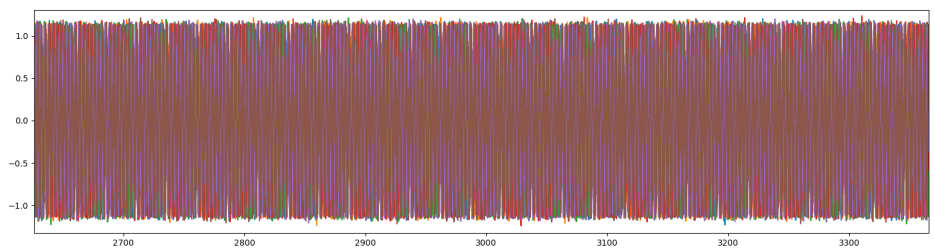

Figure 10: Data produced by Kuramoto model.

In this test, our three-member Kuramoto system began with a K value of zero; allowing each member of the system to act independently until a time $t = .5t_{max}$ when the K value was increased. Figure 10 displays the data we set our change detction algorithm on. This algorithm takes a parameter that defines the size of each of its subsequences called "width", and another that determines how much space is kept between neighbor subsequences called "lag". Figure 11 shows the performance of this algorithm, while figure 12 shows the performance of the matrix profile algorithm for anomaly detection [32].

Though the matrix profile algorithm does succeed in detecting the change in the coupling coefficient, the rest of its values fail to reflect that the system is behaving under the same (and quite simple, for the first half) regime on either side of the spike. Our algorithm, which is admittedly quite data-hungry (as discussed in section 5), not only discovers the change in dynamics, but respects the self-similarity of either half of the time-series. Furthermore, it is expected that a change to the system that would not change its dynamical kind (e.g. a change to ω values) would be detected as anomalous by the matrix profile, but would cause little change in the moving dynamical distance.

5. Partial information and degrees of control

To make use of the dynamical similarity metric defined above to compare physical systems, one needs an appropriate collection of time series. Ideally, these would consist of multiple

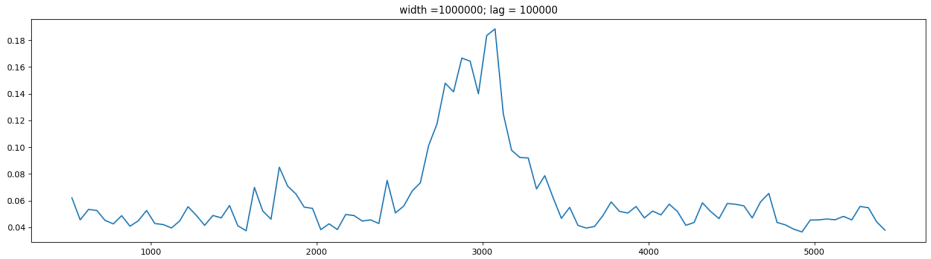

Figure 11: Anomaly detection performed by dynamic change detection. Width = 1000000; lag = 100000, for 12000000 data points.

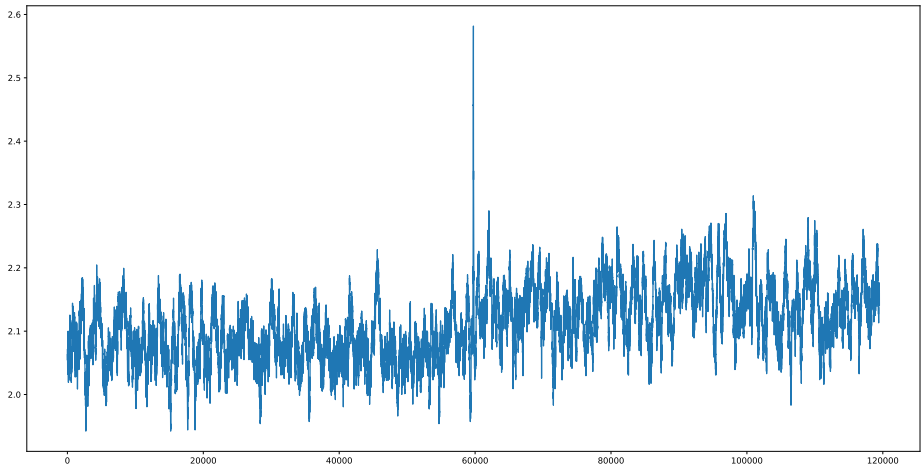

Figure 12: Anomaly detection performed by matrix profile on the same data.

replicates for each system beginning from the same initial conditions $\vec{x}$ and $\tilde{\vec{x}}$, where the each system is sampled over a length of time such that $\rho_i^A(\vec{x}) \approxeq \rho_i^B(\vec{x})$ at each time index i. If one has control over the initial conditions (the ability to intervene on the system), this is manageable. But in many cases of interest, this is impossible. Furthermore, it is often the case the variables in terms of which systems are described fail to satisfy the SSD condition, typically because the set is incomplete or amounts to a non-invertible function of an SSD set. We describe methods for applying our dynamical symmetry metric for every combination of these conditions.

5.1 SSD variable set and full control of the initial distribution

In this case, we are provided a sufficiently observable, and fully controllable system. To take samples of this system in order to satisfy $\rho_i^A(\vec{x}) \approxeq \rho_i^B(\vec{x}) \forall t_i$, we simply need to reset our system to our preferred untransformed and transformed initial conditions until we are satisfied with the number of replicates we have collected for each system. These collections can then be clipped and, as we have full control over the system, resampled, to minimize the difference between $\rho_i^A(\vec{x})$ and $\rho_i^B(\vec{x}$.

To test this method, we constructed a stochastic version of the Lotka-Volterra equations based on [17]:

$$\Delta x_i(t) = x_i(t) r_i(t) \Big(1 - \sum_{j=0}^{n} a_{ij} x_j(t)\Big) \Delta t + \sigma_i x_i(t) \xi_i(t) \sqrt{\Delta t} + \frac{\sigma_i^2}{2} x_i(t) \Big((\xi_i(t))^2 - 1\Big) \Delta t \quad (11)$$

where σ_i is the intensity of the noise for species i, and ξ_i is a Gaussian random variable of the form $\mathcal{N}(0, 1)$.

In this test, three two-species systems were compared. The parameters of systems A and B are identical, but for a scaled growth rate given to system B, and system C is given a scaled growth rate, and an increased carrying capacity. These systems of stochastic differential equations are then solved at 50 values up to $t = 15$ for ten replicates. Systems B and C are then compared to system A as σ is increased from 0.1 to 1 for all species.

As seen in figure 13, the metric remains accurate for stochastic systems with moderate values of σ, and full control makes the collection of replicates a simple task.

5.2 SSD variable set and no control of the initial distribution

This case mirrors passive sampling of a highly observable system. The method for calculating distance in this case is used in section 4.6 to determine the distance between subsequences. In this situation, we can not collect replicates by resetting the system, so we must use algorithm 2, detailed in section 5.2.1 to select replicates from the previously observed data.

5.2.1 Selecting initial conditions

When faced with partially observable, uncontrollable systems, the selection of initial conditions becomes our surrogate for control. While we can not set the initial conditions of a passively observed system the way we might with a simulation or experimental environment, we can imagine that subsequences within a time-series of observations are each an

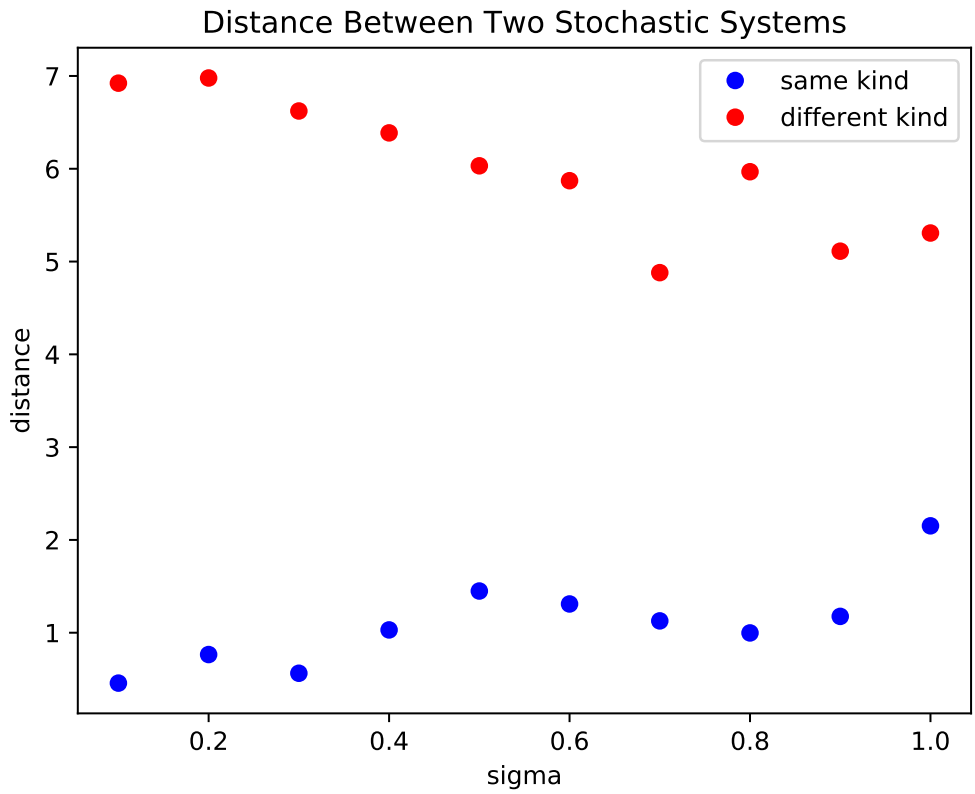

Figure 13: Dynamical distance between system A and a similar system B (blue), and between system A and a dissimilar system C (red), as stochastic magnitude increases.

initialization of that system. By carefully selecting such subsequences as instances of a system's untransformed and transformed states, we can study the system's symmetries, and compare them to those of another system.

In order to select subsequences from a time-series of observations we first arrange the initial conditions of all candidate subsequence, and find the group's first principal component. We then construct a normal distribution in the direction of the first principal component to represent our desired distribution of untransformed initial conditions, and construct a second normal distribution in the reverse direction for transformed initial conditions. A desired number of initial conditions is sampled from each distribution, and comprise the untransformed and transformed distributions necessary for the distance metric calculation.

Candidate subsequences are expected not to overlap each other, but, in cases where data is sparse, the above process can be optimized in order to maximize the number of subsequences investigated. Here, however, we present the case where sufficient data allows the selection of many candidates at regular intervals.

Once replicates have been selected from the available data, we can set about calculating the distance between systems as we would in section 5.1. As demonstrated in figure 11, our algorithm can still detect changes in dynamical kind. However, as mentioned in section 4.6, this method assumes that sufficient data is available, since every serial subsection of satisfactory length becomes a candidate replicate, but, even then, not all candidate replicates are used.

5.3 non-SSD variable set and full control of the initial distribution

The conditions of this case are similar to those in section 5.1, but the system is now only partially observable. While, again, satisfying $\rho_i^A(\vec{x}) \approxeq \rho_i^B(\vec{x}) \forall t_i$ is trivial, we do not know if the variables we can observe carry sufficient information about the system's symmetry transformation to calculate a dynamical distance with any accuracy. Even if the variables carry very little useful information, we can attempt to amplify that information by collecting more replicates. Furthermore, many non-SSD variable sets seem to retain significant information in this regard. As we'll see in section 5.4, only one rectangular coordinate is necessary to obtain good estimates of dynamical similarity for polar systems. Similarly, even the average of a system's output variables can be sufficient.

We test this latter claim similarly to the test presented in 5.1: three two-species stochastic Lotka-Volterra systems are instantiated with the same parameters as before. This time, however, we are only able to observe the average of the two species' populations.

Figure 14 shows that, while this non-SSD variable set does provide less accurate results than a fully SSD set, there is still a discernible difference between two systems of different kinds.

5.4 non-SSD variable set and no control of the initial distribution

Representing the most difficult case our algorithm can be applied to, these systems are only partially observable, and uncontrollable. In response, our method for gleaming information about dynamical kind in such situations reflects both the method used in section 5.2, and the attitude of section 5.3. In essence, we must perform the same procedure as in the case

Algorithm 2 Choose untransformed and transformed representatives from data

Require: $data, r, \alpha, \beta$

$\mu_0 \leftarrow mean(x_0 \in \vec{x} \in P \in data)$ #average of initial conditions
$cov_0 \leftarrow cov(x_0 \in \vec{x} \in P \in data)$ #covariance matrix of initial conditions
$w, v \leftarrow eig(cov_0)$ #compute eigenvalues and right eigenvectors
$w_{max} \leftarrow \sqrt{max(w)}$ #select largest eigenvalue
$\vec{e} \leftarrow v[index(w_{max} \in w)]$ #select eigenvector paired with w_{max}
$\mu \leftarrow \mu_0 - \alpha * w_{max} * \vec{e}$
$\tilde{\mu} \leftarrow \mu_0 + \alpha * w_{max} * \vec{e}$
$u, s, v \leftarrow svd(cov_0)$ #singular value decomposition of cov_0
$s \leftarrow \beta * s$
$cov_t = v \cdot (u \cdot diag(s))$
$p \leftarrow \emptyset$
$\tilde{p} \leftarrow \emptyset$
for all $P \in data$ **do**
  $q \leftarrow \emptyset$
  for all $\vec{x} \in P$ **do**
    append $\mathcal{N}(x_0|\mu, cov_t)$ to q #value of $\mathcal{N}$ at x_0
  end for
  $idx = index(sort(q))[-r:]$ #select indices of r points with densest value in $\mathcal{N}$
  $p \leftarrow [data_i | i \in idx]$
  $del(data_i | i \in idx)$ #remove already selected sequences from candidates
end for
for all $\tilde{P} \in data$ **do**
  $\tilde{q} \leftarrow \emptyset$
  for all $\vec{x} \in \tilde{P}$ **do**
    append $\mathcal{N}(x_0|\tilde{\mu}, cov_t)$ to $\tilde{q}$
  end for
  $\tilde{idx} = index(sort(\tilde{q}))[-r:]$
  $\tilde{p} \leftarrow [data_i | i \in \tilde{idx}]$
end for
return $p, \tilde{p}$

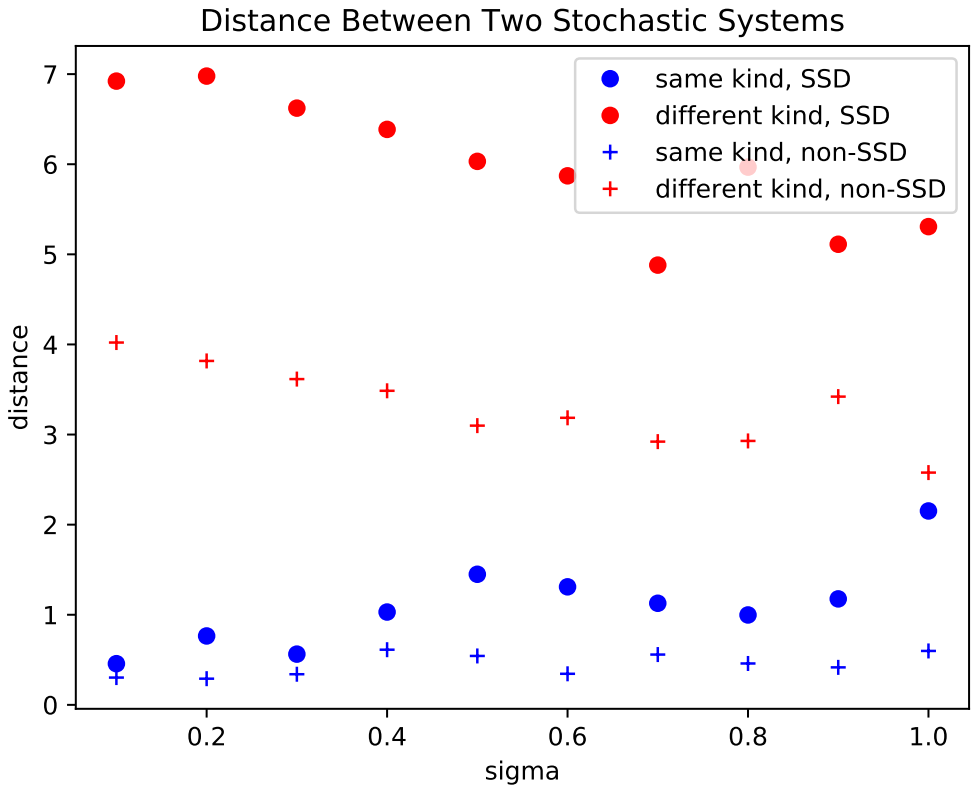

Figure 14: Dynamical distance between system A and a similar system B (blue), and between system A and a dissimilar system C (red), as stochastic magnitude increases. Comparisons between fully observable systems are denoted by filled circles, while comparisons between partially observable systems are denoted by crosses.

of SSD variables and no control and hope that the variables provide sufficient information, or else try to extract more replicates from the observed data.

We demonstrate our method in this most restricted case by revisiting the Kuramoto system that exhibits dynamical change part-way through its observation. For this test, however, we only have access to one of the oscillator's coordinates.

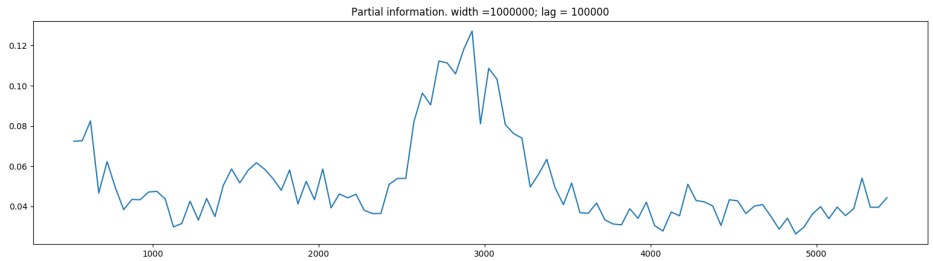

Figure 15: Anomaly detection performed by dynamic change detection on a system whose variables are non-SSD. Width = 1000000; lag = 100000, for 12000000 data points.

Though many systems of interest in the real world have even less informative, observable variables than the example given here, the high performance evidenced above suggests significant robustness in the face of information loss. The results in figure 15 show a high sensitivity to dynamical change, even when we have no control over the system, and lack significant amounts of information.

6. Conclusion

In this paper, we have presented a theory and framework for calculating a similarity metric for dynamical kinds. Further, we have shown that this metric is sensitive to simple changes in dynamical kind, differences among systems' linearity and effective order, and the presence of chaos. When applied to chaotic systems, our method reveals striking dynamical structure in the evolution of distances as comparisons are made throughout parameter space. To that end, we demonstrate the use of our metric as a tool to explore regions of a system's parameterizations. We also compare our approach to the performance of the l_2 norm for detecting the difference in between chaotic and non-chaotic systems, proving our metric to be more accurate in chaotic regimes. We also compare our algorithm's capability to detect changes in a system's dynamical kind to the anomaly detecting power of the matrix profile approach, showing the validity of our method for detecting salient changes, and suggesting that matrix profile is more sensitive to changes that don't affect kind. An exploration of our metric's performance as our knowledge and control of a system of interest changes is also given; showing significant retention of accuracy in even the most trying cases.

Armed with this tool, we can determine if a model and the system it is meant to reflect share the same degree of nonlinearity, or effective order without having to test candidate solutions. With this tool, we can decide if two chaotic systems share the same parameters,

or tell if a system's causal structure has changed. This tool allows us to explore a system's parameter space with expectations about how similar certain regions that lie close together may behave very differently, even in the absence of chaos.

However, this tool is no panacea for understanding the underlying structure of system dynamics. Prominently, the method's requirement for transformed and untransformed trajectories (and, in some cases, multiples of each) means that an abundance of data is required for most real-world explorations. Similarly, the process is computationally complex relative to a simple norm, and seems to work just a well in cases with low degrees of nonlinearity. This metric can also be confused by long observations of a system's equilibrium, requiring some fore-knowledge of simpler systems in order to avoid such behavior.

These challenges are not insurmountable, and there are many areas of application this general metric and find use. In order to reduce the data requirements of this method, optimizations can be performed on the clipping and initial condition selection routines - trading computational complexity for data efficiency. To reduce overall complexity, it may be possible to replace certain facets of the distance calculation with statistical estimations. In the space of applications, the concept of control by dynamical kind seems promising as a fault-tolerant approach to pursuing high-order or emergent behaviors. This tool is also positioned well to act as a quick validation of a model of an observed phenomenon. Finally, we suggest that the rich structure captured in our brief exploration of the parameter space of the Lotka-Volterra system indicates the possibility of using this metric as a powerful tool for explaining behavioral regimes in complex systems.

References

[1] T. L. Bailey, M. Boden, F. A. Buske, M. Frith, C. E. Grant, L. Clementi, J. Ren, W. W. Li, and W. S. Noble. Meme suite: tools for motif discovery and searching. *Nucleic acids research*, 37(suppl_2):W202–W208, 2009.

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

[3] D. Bauer. Order estimation for subspace methods. *Automatica*, 37(10):1561 – 1573, 2001. ISSN 0005-1098. doi: https://doi.org/10.1016/S0005-1098(01)00118-2. URL `http://www.sciencedirect.com/science/article/pii/S0005109801001182`.

[4] Y. Bayarjargal, A. Karnieli, M. Bayasgalan, S. Khudulmur, C. Gandush, and C. Tucker. A comparative study of noaa–avhrr derived drought indices using change vector analysis. *Remote Sensing of Environment*, 105(1):9–22, 2006.

[5] K. M. Bergen, D. G. Brown, J. F. Rutherford, and E. J. Gustafson. Change detection with heterogeneous data using ecoregional stratification, statistical summaries and a land allocation algorithm. *Remote Sensing of Environment*, 97(4):434–446, 2005.

[6] J. C. Brown, W. E. Jepson, J. H. Kastens, B. D. Wardlow, J. M. Lomas, and K. Price. Multitemporal, moderate-spatial-resolution remote sensing of modern agricultural production and land modification in the brazilian amazon. *GIScience & Remote Sensing*, 44(2):117–148, 2007.

[7] A. Celletti and A. E. Villa. Low-dimensional chaotic attractors in the rat brain. *Biological Cybernetics*, 74(5):387–393, 1996.

[8] J. Evans and R. Geerken. Classifying rangeland vegetation type and coverage using a fourier component based similarity measure. *Remote Sensing of Environment*, 105(1): 1–8, 2006.

[9] J. Fell, J. Röschke, and P. Beckmann. Deterministic chaos and the first positive lyapunov exponent: a nonlinear analysis of the human electroencephalogram during sleep. *Biological cybernetics*, 69(2):139–146, 1993.

[10] A. Garfinkel, M. L. Spano, W. L. Ditto, and J. N. Weiss. Controlling cardiac chaos. *Science*, 257(5074):1230–1235, 1992.

[11] X. Hong, R. J. Mitchell, S. Chen, C. J. Harris, K. Li, and G. W. Irwin. Model selection approaches for non-linear system identification: a review. *International journal of systems science*, 39(10):925–946, 2008.

[12] B. C. Jantzen. Dynamical Kinds and their Discovery. *Proceedings of the UAI 2016 Workshop on Causation: Foundation to Application*, 2017. URL `http://ceur-ws.org/Vol-1792/paper2.pdf`. arXiv: 1612.04933.

[13] C. W. Kulp, L. Zunino, T. Osborne, and B. Zawadzki. Using missing ordinal patterns to detect nonlinearity in time series data. *Physical Review E*, 96(2):022218, 2017.

[14] Y. Kuramoto. International symposium on mathematical problems in theoretical physics. *Lecture notes in Physics*, 30:420, 1975.

[15] S. Lhermitte, J. Verbesselt, I. Jonckheere, K. Nackaerts, J. A. van Aardt, W. W. Verstraeten, and P. Coppin. Hierarchical image segmentation based on similarity of ndvi time series. *Remote Sensing of Environment*, 112(2):506–521, 2008.

[16] S. Lhermitte, J. Verbesselt, W. W. Verstraeten, and P. Coppin. A comparison of time series similarity measures for classification and change detection of ecosystem dynamics. *Remote Sensing of Environment*, 115(12):3129–3152, 2011.

[17] M. Liu and M. Fan. Permanence of stochastic lotka–volterra systems. *Journal of Nonlinear Science*, 27(2):425–452, 2017.

[18] A. Lobo and P. Maisongrande. Searching for trends of change through exploratory data analysis of time series of remotely sensed images of sw europe and nw africa. *International Journal of Remote Sensing*, 29(17-18):5237–5245, 2008.

[19] J. Nichols, M. Seaver, S. Trickey, M. Todd, C. Olson, and L. Overbey. Detecting nonlinearity in structural systems using the transfer entropy. *Physical Review E*, 72 (4):046217, 2005.

[20] M. Paluš. Testing for nonlinearity using redundancies: Quantitative and qualitative aspects. *Physica D: Nonlinear Phenomena*, 80(1-2):186–205, 1995.

[21] M. Paluš. Detecting nonlinearity in multivariate time series. *Physics Letters A*, 213 (3-4):138–147, 1996.

[22] B. Pilgram, W. Schappacher, W. Löscher, and G. Pfurtscheller. Application of the correlation integral to respiratory data of infants during rem sleep. *Biological cybernetics*, 72(6):543–551, 1995.

[23] C. Rhodes and M. Morari. Determining the model order of nonlinear input/output systems directly from data. *Proceedings of 1995 American Control Conference - ACC'95, American Control Conference, Proceedings of the 1995*, page 2190, 1995. ISSN 0-7803-2445-5. URL `http://login.ezproxy.lib.vt.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.531288&site=eds-live&scope=site`.

[24] L. Roques and M. D. Chekroun. Probing chaos and biodiversity in a simple competition model. *Ecological Complexity*, 8(1):98–104, 2011.

[25] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26 (1):43–49, February 1978. ISSN 0096-3518. doi: 10.1109/TASSP.1978.1163055.

[26] T. Sauer. Reconstruction of dynamical systems from interspike intervals. *Physical Review Letters*, 72(24):3811, 1994.

[27] S. J. Schiff, K. Jerger, D. H. Duong, T. Chang, M. L. Spano, and W. L. Ditto. Controlling chaos in the brain. *Nature*, 370(6491):615, 1994.

[28] T. Schreiber. Measuring information transfer. *Physical review letters*, 85(2):461, 2000.

[29] G. Sugihara and R. M. May. Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature*, 344(6268):734, 1990.

[30] P. Vieu. Order choice in nonlinear autoregressive models. *Statistics*, 26(4):307–328, 1995. doi: 10.1080/02331889508802499. URL `https://doi.org/10.1080/02331889508802499`.

[31] V. Volterra. Fluctuations in the abundance of a species considered mathematically, 1926.

[32] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh. Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 1317–1322. IEEE, 2016.

[33] G. Yunfan, X. Jianxue, R. Wei, H. Sanjue, and W. Fuzhou. Determining the degree of chaos from analysis of isi time series in the nervous system: a comparison between correlation dimension and nonlinear forecasting methods. *Biological cybernetics*, 78(2): 159–165, 1998.

[34] L. Zunino and C. W. Kulp. Detecting nonlinearity in short and noisy time series using the permutation entropy. *Physics Letters A*, 381(42):3627–3635, 2017.

Proceedings of the ASME 2018 Dynamic Systems and Control Conference
DSCC2018
September 30-October 3, 2018, Atlanta, Georgia, USA

DSCC2018-9139

CHAPTER 4: DIFFERENTIATION OF COLLECTIVE BEHAVIOR BASED ON AUTOMATED DISCOVERY OF DYNAMICAL KINDS

Amanda Hashimoto, Nicole Abaid*
Engineering Mechanics Program
Virginia Tech
Blacksburg, Virginia, 24061
Email: ahashimo, nabaid@vt.edu

Subhradeep Roy, Benjamin Jantzen
Department of Philosophy
Virginia Tech
Blacksburg, Virginia, 24061
Email: sdroy, bjantzen@vt.edu

Colin Shea-Blymyer
Department of Computer Science
Virginia Tech
Blacksburg, Virginia, 24061
Email: c0lin@vt.edu

ABSTRACT

In this paper, we explore a model of collective behavior using EUGENE, an algorithm for automated discovery of so-called "dynamical kinds". Two systems are of the same dynamical kind if their underlying causal dynamics are similar, as defined using dynamical symmetry. We apply EUGENE to simulation data from a model capable of generating a range of qualitatively different collective behaviors, from aligned motion to circular milling. These behaviors are measured using both global and local order parameters, and this data is analyzed with EUGENE. We find that EUGENE is capable of differentiating between these systems when global order parameters are used, and can only identify more coarse characteristics when local order parameters are considered.

INTRODUCTION

Collective behavior refers to group-level complexity that can emerge from local interactions among multiple individuals. Besides being studied extensively in natural animal groups [1], such as bird flocks and fish schools, collective behavior is of interest to the larger scientific community for its ability to generate desired behaviors without directly controlling each individual in a group. This work finds application in engineering in particular, for example, in the design and control of robotic swarms, granular media, and engineered organisms [2–4].

Collective behavior can be generated in models for multi-agent systems that apply agent-based rules [5] or view the group as a continuum [6]. These models have been shown capable of capturing a variety of behaviors that either replicate observations of nature [7] or are relevant to an engineering application [8]. As an example, the model in [9] defines interactions among a group of self-propelled particles through potential functions and includes an external source to diversify the types of collective behaviors the model can produce. These behaviors include aligned motion, unaligned swarming, and circular milling patterns, and are distinguished by selecting model parameters.

For the model in [9] and in the literature in general, collective behavior is quantified by defining order parameters, whose values capture the level of coordination in the group. Examples of order parameters include linear momentum [10], angular momentum [7], and a measure of closeness in space [9]. A detractor of such order parameters is that they are defined irrespective of the system dynamics and take a human-centered, global perspective on the group's behavior. Not only does this ubiquitous view neglect the perspective of an individual (which would be relevant to an animal group, for example), there has not been a rigorous demonstration of the parameters' optimality for measuring order or complexity as far as we know. This open question on what order parameters are appropriate to measure collective behavior may be answered by using model-free data-driven methods to directly study collective dynamics without imposing any model on the system.

Data-driven methods have been used to study collective behavior, at least in terms of pairwise relationships that can be used to build an interaction network. For example, information flow has been measured to detect animal group interactions among

*Address all correspondence to this author.

fish [11,12] and insects [13] moving in groups, and time-delayed embeddings have been used for studying causality in economic systems [14] and neural networks [15]. These studies use a variety of methods, such as entropy (based in information theory) and manifold embedding (taken from dynamical systems). To a lesser extent, tools from the field of causal discovery have been used to learn about the causal structure of collectives in terms of both individual-level relationships and connections between high-level features of the collective [16, 17]. Recently, work on the automated discovery of scientifically relevant kinds has provided a set of tools for indirectly assessing the similarity of underlying dynamics in complex systems [18]. However, these tools have yet to be applied to the empirical investigation of collective behavior.

To gain a broad understanding of how data-driven analytical methods can be used to quantify collective behavior, it is necessary to test a system for which the dynamics can be precisely defined. To be useful for investigating collective behavior in real systems where it is difficult or impossible to obtain accurate information about the behavior of all members, we ideally require an analytical method that is sensitive to detailed causal structure and insensitive to possible confounding factors. Such factors include the choice of variables used to describe the collective, measurement noise, and stochasticity in the underlying dynamics. A new algorithm in the EUGENE collection of automated discovery tools [19] satisfies all of these desiderata.

In this paper, we seek to understand the potential of EUGENE to study collective behavior using an established model. We demonstrate the application of EUGENE to measure different behaviors generated by the self-propelled particle (SPP) model in [9]. We show that this algorithm is capable of distinguishing between different behaviors using low-dimensional parameters as input, rather than the high-dimensional full state of the system. In addition, we present a local version of such conventional order parameters, to which EUGENE can be analogously applied, and we find that results are less sensitive to behavioral differences. We find that EUGENE can differentiate behaviors in model data, even when the full state of the system is not given. This work supports EUGENE's use in future work to analyze field data from animal groups for which the full dynamics may be very difficult to observe and the rules governing interactions cannot be known.

SELF-PROPELLED PARTICLE MODEL

SPP model with external leader particle

The self-propelled particle model used in this study investigates the collective behavior of a multi-agent system interacting with an external leader particle (ELP). This model is developed and studied in [9] for the wide range of collective behaviors it produces. The agents in the group are N identical, self-propelled particles of mass m moving in two dimensions. The position vector of the i-th particle is denoted as $\mathbf{x}_i(t)$, $i = 1,\ldots,N$, and its velocity as $\mathbf{v}_i(t) = \dot{\mathbf{x}}_i(t)$, where $t \in \mathbb{R}$ is the time variable. Interactions between particles are governed by attractive and repulsive forces, and all agents also interact with a fixed ELP. The dynamics of the i-th agent, $i = 1,\ldots,N$, are given by

$$m\dot{\mathbf{v}}_i = (\alpha - \beta\|\mathbf{v}_i\|^2)\mathbf{v}_i + \mathbf{F}_i + \mathbf{F}_i^0 \tag{1}$$

with $\alpha, \beta > 0$, $\alpha\mathbf{v}_i$ is a self-acceleration term, and $-\beta\|\mathbf{v}_i\|^2\mathbf{v}_i$ is a friction term. The force between agents is

$$\mathbf{F}_i = -\sum_{\substack{j=1\\ j\neq i}}^{N} \nabla_{\mathbf{x}_i}\Phi(\|\mathbf{x}_{ij}\|) \tag{2}$$

where $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ and Φ is a generalized Morse potential given by $\Phi(z) = -c_a\mathbf{e}^{-z/l_a} + c_r\mathbf{e}^{-z/l_r}$ for a positive scalar z. The constants c_a, c_r, l_a, and l_r give the strength and cut-off length for the attractive and repulsive forces, labeled with a and r, respectively. The force between the ELP and agent i is

$$\mathbf{F}_i^0 = -\gamma\nabla_{\mathbf{x}_i}\Phi(\|\mathbf{x}_{i0}\|) \tag{3}$$

where $\mathbf{x}_{i0} = \mathbf{x}_i - \mathbf{x}_0$ is the relative position with respect to the ELP, and γ is the strength of interaction between particles ($\gamma \geq 0$). By varying γ and the self-acceleration term α in simulation, we analyze the generated behaviors with EUGENE.

Global order parameters

In line with the analysis in [9], we compute three physics-based order parameters using the state of the whole system of particles. These parameters capture linear and angular momenta, and hence alignment and rotation of the group with respect to the ELP, and cohesion with respect to the ELP's fixed position. At time t, the normalized group linear momentum (polarization) $P(t)$, the normalized group angular momentum $M(t)$, and a measure of cohesiveness $C(t)$ with respect to the ELP are defined as

$$P(t) = \frac{\|\sum_{i=1}^{N}\mathbf{v}_i(t)\|}{\sum_{i=1}^{N}\|\mathbf{v}_i(t)\|} \tag{4}$$

$$M(t) = \frac{\|\sum_{i=1}^{N}\mathbf{x}_{i0}(t)\times\mathbf{v}_i(t)\|}{\sum_{i=1}^{N}\|\mathbf{x}_{i0}(t)\|\|\mathbf{v}_i(t)\|} \tag{5}$$

$$C(t) = \frac{1}{N}\sum_{i=1}^{N}\exp\left[-\frac{\|\mathbf{x}_{i0}(t)\|}{4l_a}\right] \tag{6}$$

We comment that, while the polarization is independent of the ELP, the angular momentum and cohesion are defined based on

relative velocities or positions with respect to the ELP. In [9], the authors found that these parameters allowed for distinguishing between distinct states better than considering analogous order parameters defined with respect to the system's center of mass, which would be an intuitive alternate form. As a note, M and C in this work correspond to M^0 and C^0 in [9].

Local order parameters

While we observe in [9] that the group's state is qualitatively well-described by these order parameters, it is often extremely difficult to obtain such detailed information from a natural system, particularly with very large groups. To complement these global parameters, we also define local order parameters from the perspective of an individual agent, with motivation that this data may be more easily attainable and may in fact be useful to recover the group-level behavior.

We define local order parameters with respect to a randomly-chosen focal agent f, which is selected independently and used for all time steps in a single simulation. Once this focal agent is selected, we define a range R and consider all agents in the ball of radius R centered at the focal agent's position, $\mathbf{x}_f(t)$. We call this set of agents $N^L(t)$. The local linear momentum (local polarization) with this limited perspective is

$$P^L(t) = \frac{\|\sum_{i\in N^L(t)} \mathbf{v}_i(t)\|}{\sum_{i\in N^L(t)} \|\mathbf{v}_i(t)\|} \tag{7}$$

The local angular momentum is defined using the relative position of agents in $N^L(t)$ with respect to $\mathbf{x}_f$. Specifically,

$$M^L(t) = \frac{\|\sum_{i\in N^L(t)} \mathbf{x}_{if}(t) \times \mathbf{v}_i(t)\|}{\sum_{i\in N^L(t)} \|\mathbf{x}_{if}(t)\| \|\mathbf{v}_i(t)\|} \tag{8}$$

where $\mathbf{x}_{if} = \mathbf{x}_i - \mathbf{x}_f$ is the relative position of the agents with respect to the selected focal agent. The cohesion is similarly defined with respect to the focal agent, as

$$C^L(t) = \frac{1}{|N^L(t)|} \sum_{i\in N^L(t)} \exp\left[-\frac{\|\mathbf{x}_{if}(t)\|}{4l_a}\right] \tag{9}$$

where $|\bullet|$ gives the cardinality of a set.

Five distinct behavioral states

Since the goal of this paper is generating simulation data evidencing different behaviors in a system of particles, we select model parameters based on the parameter study in [9] that show a range of states. For these simulations, we take $N = 50$, $m = 1$, $\beta = 0.5$, $l_r = 0.5$, $l_a = 2$, $c_a = 0.5$, and $c_r = 1$. The parameters α and γ are varied to obtain different group behaviors. We use two different sets of initial conditions to generate random position and velocity vectors for the agents. The agents are initially dispersed uniformly within a square of side length $2l_a$ that is centered around the origin. For all simulations, the same initial conditions are used to allow for the comparison that EUGENE requires.

The simulation is run for the time interval $[0, 1000]$ and the entire run (including the initial transient) is used for analysis. This mostly transient data was selected since EUGENE relies on creating maps between systems at a variety of states. In contrast to most analyses for models of collective behavior, analysis with EUGENE is much less informative when the system is in steady-state.

We generate data for five different system behaviors (denoted A, B, C, D, and E) by varying α and γ, summarized in

TABLE 1. SYSTEM BEHAVIORS AND THEIR ASSOCIATED α AND γ. THE TRIPLE $[P, M, C]$ CORRESPOND TO GENERALLY ORDERED (1) AND DISORDERED (0) STEADY STATES, AND INCLUDE A SHORT DESCRIPTION OF THE BEHAVIOR.

State	α	γ	$[P, M, C]$	Behavior
A	0.2	0.0001	$[0,0,0]$	No organized motion
B	0.02	0.3	$[0,0,1]$	Cohesive unaligned motion
C	0.8	0.8	$[0,1,1]$	Milling on a circle
D	0.5	4.0	$[1,0,0]$	Cluster on a linear path
E	0.2	6.0	$[1,1,1]$	Cluster on a circular path

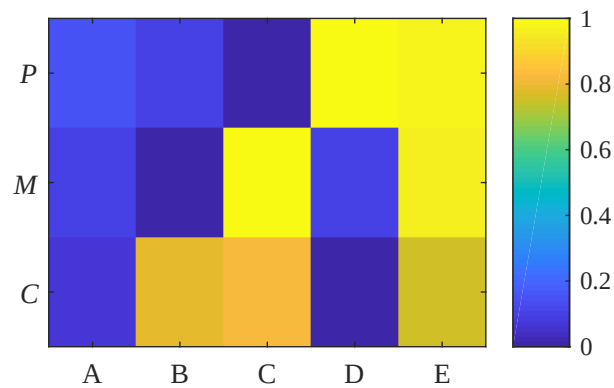

FIGURE 1. MEAN VALUES OF STEADY-STATE GROUP MOMENTUM (P), ANGULAR MOMENTUM (M), AND COHESIVENESS (C) WITHIN THE 5 BEHAVIORAL STATES.

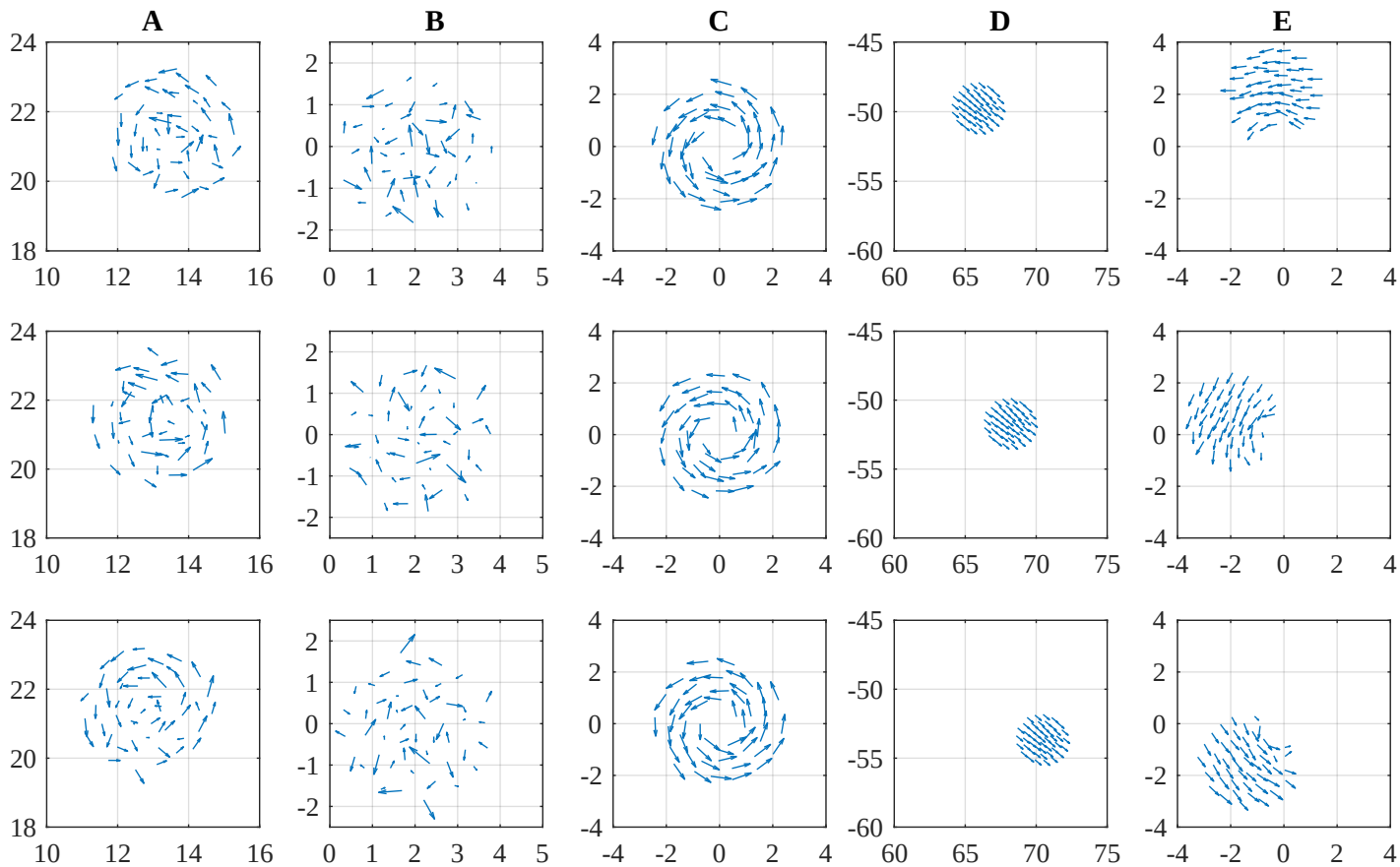

FIGURE 2. SNAPSHOTS OF GROUP MOTION FOR THE FIVE BEHAVIORAL STATES (LABELED FOR EACH COLUMN). FIGURES IN THE FIRST, SECOND, AND THIRD ROW ARE AT $t = 950$, 960, AND 970, RESPECTIVELY. ELP IS LOCATED AT THE ORIGIN.

table 1. These simulations capture a range of behaviors. State A acts as a control condition, since the particles show no strong alignment, rotation, or cohesion about the ELP. State B shows cohesive behavior, but with no alignment or organized milling. State C shows particles moving on an approximately circular trajectory, with agents distributed all along the circumference of the circle similarly to motion on the rim of a wheel. State D shows highly aligned motion of the group as a whole along a linear path. State E shows motion again on an approximately circular trajectory, but in contrast to State C, the particles are concentrated in a cohesive group and move together in an aligned fashion. To ensure that these model parameters generate the desired behaviors, we simulated the model for the time interval [0,1000] and computed the order parameters over the last 750 seconds. These values are given in figure 1 and summarized generally as "ordered" (1) or "disordered" (0) in table 1.

Snapshots of representative steady-state group motion over time for these behaviors are shown in figure 2. Nevertheless, the analysis detailed below is performed on the transient data for each simulation. Time series showing this data for each combination of order parameter and behavior are shown in figure 3, with both the global and local order parameters for polarization, angular momentum, and cohesion plotted over the time interval [0,1000]. These data are used as input for the analysis with EUGENE.

EUGENE ALGORITHM FOR AUTOMATED DISCOVERY

EUGENE is an algorithm for determining the similarity between two systems' underlying causal dynamics [18]. Unlike examples in the existing literature which rely on low dimensional [20] or time delay [21] embeddings, it leverages dynamical symmetries - commutations between an evolution and transformation of a system - to verify that two systems are of different kinds. The original algorithm, however, is not designed for stochastic systems or cases of partial information, so we implement an extension of EUGENE that compares samples from carefully selected distributions over states of one system with the

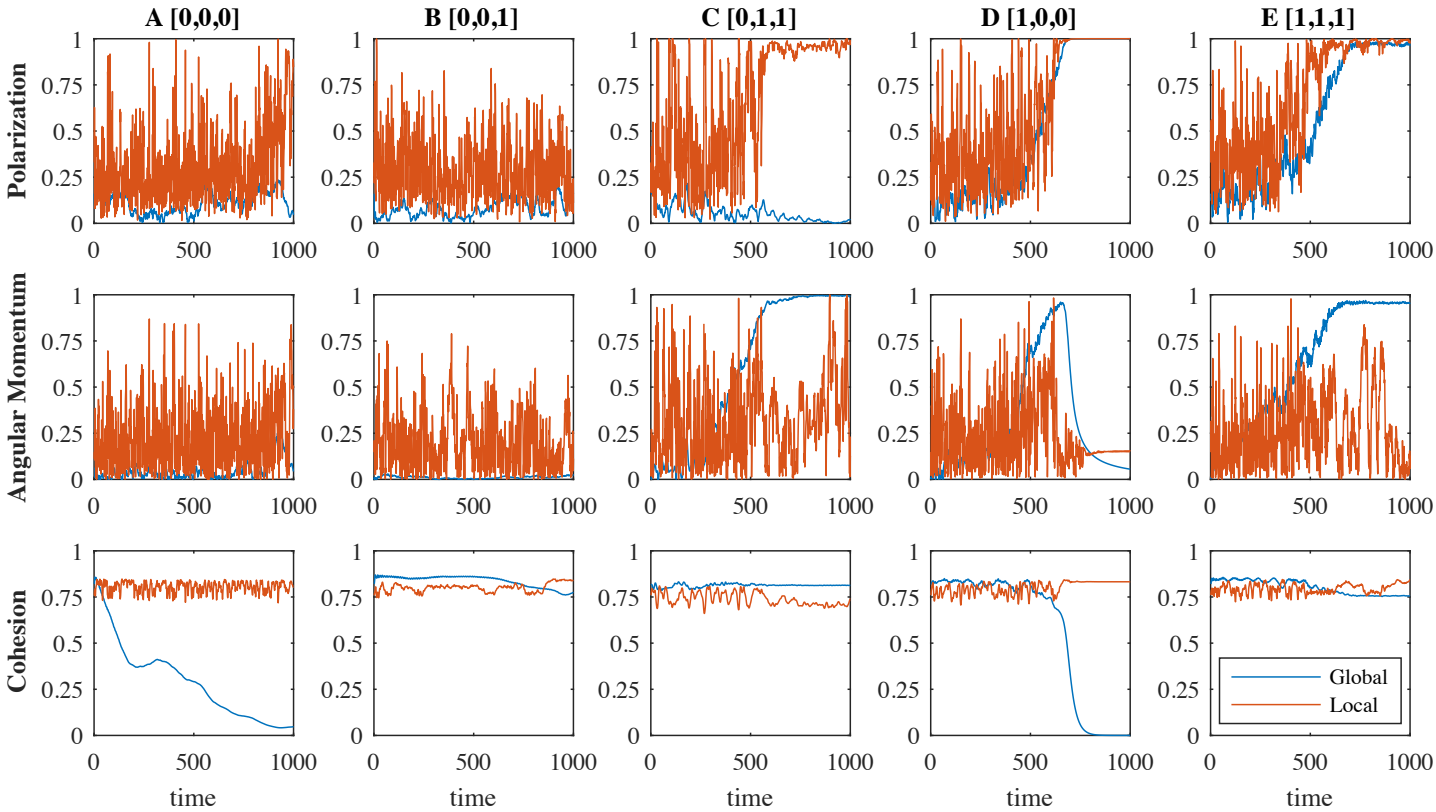

FIGURE 3. TIME SERIES DATA FOR EACH BEHAVIORAL STATE (LABELED FOR EACH COLUMN). THE FIGURES IN THE FIRST, SECOND, AND THIRD ROW CORRESPOND TO POLARIZATION, ANGULAR MOMENTUM, AND COHESION, RESPECTIVELY. THE TWO LINES ARE THE GLOBAL AND LOCAL ORDER PARAMETERS CALCULATED FOR EACH TIME SERIES.

corresponding distribution from another system. These distributions will be identical if and only if the two systems share the same dynamical symmetry, and the degree to which the distributions diverge provides an informative measure of how much the dynamics of one system diverges from that of the other.

Dynamical symmetries and kinds

As given in [22], a *dynamical symmetry* is a transformation σ on the variables of a system such that the final state of the system is identical regardless of whether σ is applied to the system before or after an intervention on the system's index variable (e.g. transformation over time). This definition was generalized in [18] to accommodate stochastic dynamics, which we adopt in the following form:

Definition 1 (Dynamical symmetry). *Let V be a set of random variables. Let σ be an intervention on the variables in $Int \subset V$. The transformation σ is a dynamical symmetry with respect to some index variable $X \in V - Int$ if and only if σ has the following property: for all probability distributions f and g, the final probability distribution over V is the same whether σ is applied when the distribution of X is given by $p_x(x) = f(x)$ and then an intervention on X makes it such that $p_x(x) = g(x)$, or the intervention on X is applied first, changing its distribution from $f(x)$ to $g(x)$, and then σ is applied.*

Dynamical symmetries are sensitive to both causal structure (i.e., the set of relations indicating which variables directly influence which others) and the functional form of causal influence. Thus, if the (possibly stochastic) dynamics underlying one system is described by, say, a differential equation of slightly different form than that of another system, the two will generally possess distinct yet similar dynamical symmetries. Thus, if one can directly compare the dynamical symmetries between systems with unknown dynamics, one can quantify their degree of dynamical similarity in a manner that is data-driven and model-free.

Current approach

The EUGENE algorithm that has been adapted from [19] can detect differences in dynamical symmetries in this general

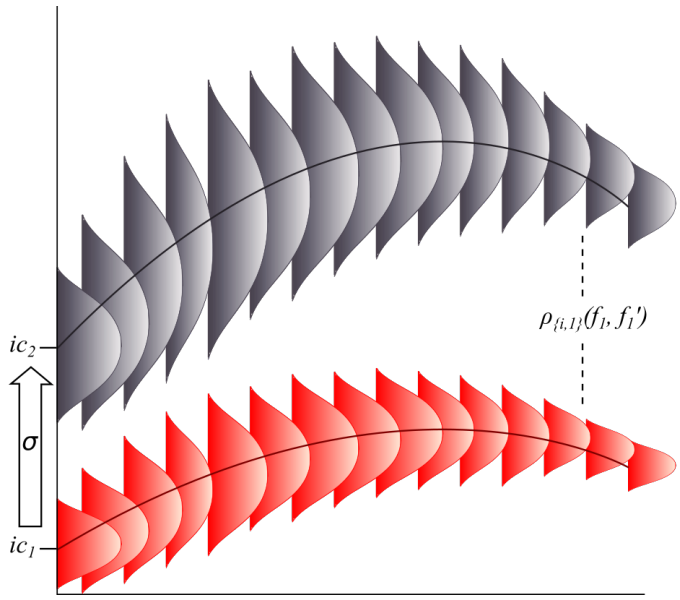

FIGURE 4. SCHEMATIC OF EUGENE ALGORITHM. f_1 AND f_1' ARE COMPOSED OF MULTIPLE REALIZATIONS STARTING FROM INITIAL CONDITIONS ic_1 AND ic_2, RESPECTIVELY, AND EVOLVING IN TIME HORIZONTALLY. RESULTING TIME SERIES ARE VIEWED AS THE RED AND GRAY DYNAMIC DISTRIBUTIONS (SNAPSHOTS SHOWN CHANGING OVER TIME). EUGENE TAKES THE POINTWISE CUMULATIVE JOINT DISTRIBUTION BETWEEN THESE TWO FUNCTIONS TO CHARACTERIZE SYSTEM A. THE SAME PROCESS IS PERFORMED FOR SYSTEM B, AND THE ENERGY DISTANCE BETWEEN THE TWO DISTRIBUTIONS DEFINES DIFFERENCE IN DYNAMICAL KIND.

sense. To do so, the algorithm requires two sets of time series for each of the systems A and B to be compared. Each set of time series contains a realization of the evolution of the system starting from the same initial condition. If the system is stochastic (or if the variables in terms of which it is described are incomplete), each realized time evolution in a set will be distinct. However, the distribution over states at any given time is fixed for each system (assuming the dynamics are unchanging). The function that maps this distribution for one initial condition (ic_1) to the distribution for another initial condition (ic_2) in the same system is a dynamical symmetry. By comparing the function for system A with that for system B, we can assess the similarity of their dynamics. Rather than estimate these functions directly, the EUGENE algorithm compares the joint distribution over states for system A at a time t after ic_1 and at a time t after ic_2 with the corresponding distribution from system B. Cumulative density functions are compared by computing the energy distance between them [23].

For analyzing data from the SPP model, the EUGENE algorithm requires two realizations of the simulation for each set of parameter values of interest, where each realization is a multi-dimensional time-series spanning a fixed amount of time, Δt. The first realization is computed for a certain initial condition, ic_1. Using the same simulation parameters, a second set of realizations is produced using a second initial condition, ic_2. We refer to the time series of the first realization as $\vec{f}_1$, and that of the second realization as $\vec{f}_1'$, where each is of length n corresponding to the number of samples in the series. Then, using a new set of simulation parameters, two more sets of realizations are given using the same initial conditions used previously. These new sets of realizations will be referred to as $\vec{f}_2$ and $\vec{f}_2'$. Both $\vec{f}_1$ and $\vec{f}_1'$ belong to the first simulation configuration, while $\vec{f}_2$ and $\vec{f}_2'$ belong to the second. Similarly, $\vec{f}_1$ and $\vec{f}_2$ share initial conditions (ic_1), while $\vec{f}_1'$ and $\vec{f}_2'$ share a different set of initial conditions (ic_2). Each pair of time series elements $f_{1,i}$ and $f_{1,i}'$ for $i \in \{1, 2, \ldots, n\}$ (that is, each pair of values at a given time for a given system) is treated as a realization of a pair random variables with a joint density function, $\rho_{1,i}(f_1, f_1')$. The same is true for pairs $f_{2,i}$ and $f_{2,i}'$ which are presumed to be governed by a joint density function, $\rho_{2,i}(f_2, f_2')$. Consider the cumulative distribution cdf_A over pairs of simultaneous values of f_1 and f_1', and cdf_B over pairs of values of f_2 and f_2', where $cdf_A(F_1, F_1') = \sum_{i=1}^{n} \frac{1}{n} \int_{-\infty}^{F_1} \int_{-\infty}^{F_1'} \rho_{1,i}(f_1, f_1') df_1 df_1'$ and $cdf_B(F_2, F_2') = \sum_{i=1}^{n} \frac{1}{n} \int_{-\infty}^{F_2} \int_{-\infty}^{F_2'} \rho_{2,i}(f_2, f_2') df_2 df_2'$. If the marginal distribution over values of f_1 is approximately the same as that over f_2, then the difference between cdf_A and cdf_B is driven entirely by the difference in dynamical symmetries between systems A and B. We compute a metric to quantify this difference. Specifically, EUGENE computes an estimate of the energy distance between cdf_A and cdf_B [23]. A schematic for this algorithm is given in figure 4.

RESULTS AND DISCUSSION

By using the SPP model as an input into EUGENE, we confirm that EUGENE is able to identify the distinct behavioral states by distinguishing them as different dynamical kinds. In figure 5, the pairwise distances between the five behavioral states are shown as distance matrices for each of the global and local order parameters. The color axis on the right denotes the pairwise distances between the states, with distance $= 0$ corresponding to the two states being of the same dynamical kind and distance > 0 corresponding to the states being of different dynamical kinds. The magnitude of this difference is measured by the distance.

Despite the fact that the model has not converged to steady state as seen in figure 3, EUGENE is able to pick up trends for the global order parameters, which are shown in the first row of figure 5. The expected trends for the five system behaviors are the $[P, M, C]$ values in table 1. For global polarization P, for example, EUGENE finds that the behavioral States A, B, and C are the same, which is to be expected since their P values are low.

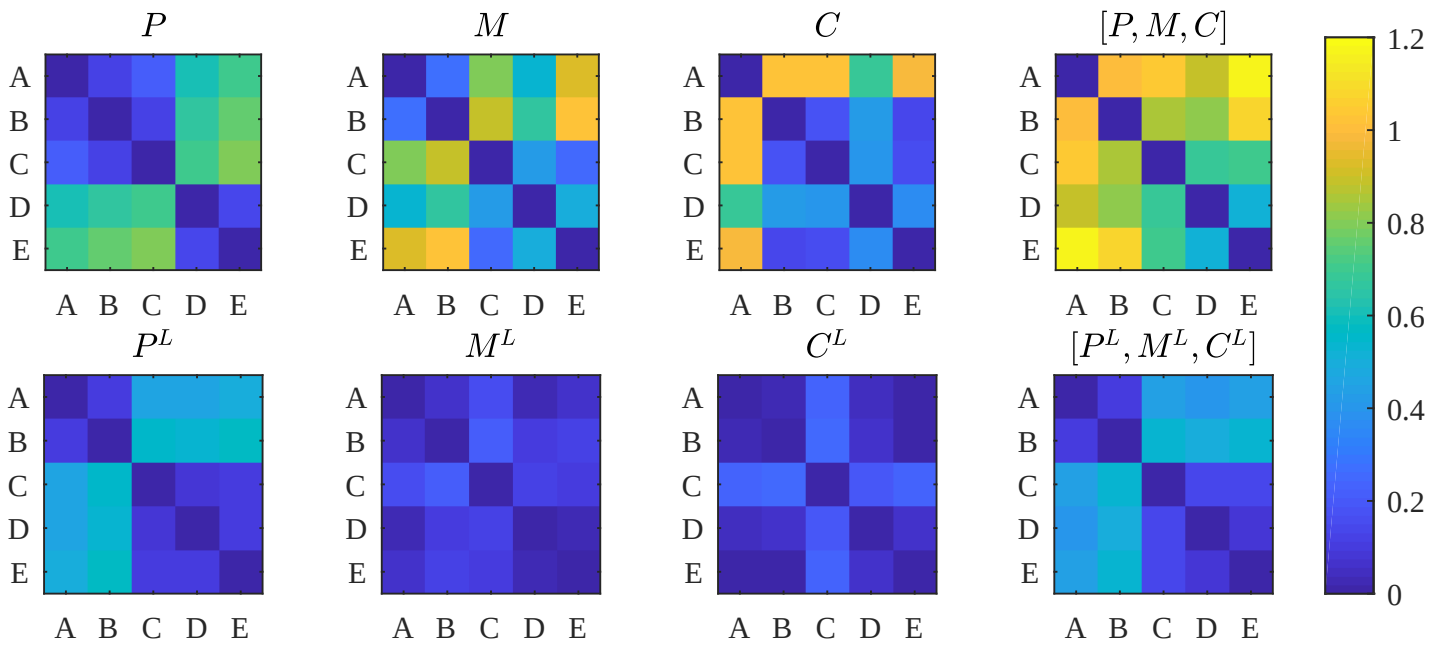

FIGURE 5. DISTANCE BETWEEN FIVE BEHAVIORAL STATES AS COMPUTED BY EUGENE. EACH SUBFIGURE SHOWS A DISTANCE MATRIX, WHERE THE PAIRWISE DISTANCE BETWEEN STATES LABELLED ON ROWS AND COLUMNS IS DENOTED BY THE COLOR CORRESPONDING TO THE COLOR AXIS ON THE RIGHT. THE TITLE OF EACH SUBFIGURE SHOWS WHICH ORDER PARAMETER WAS USED AS INPUT TO EUGENE; THE FIRST THREE COLUMNS EACH USE ONE ORDER PARAMETER AND THE FOURTH COLUMN USES ALL THREE EITHER LOCAL OR GLOBAL PARAMETERS.

The algorithm also detects that State D and E are the same (with high P values) and distinct from A, B, and C. When using angular momentum as the input, the States A and B are found to be similar with low angular momentum, and likewise with States C and E, which have high angular momentum. Interestingly, State D seems to be more alike the behavioral states with high angular momentum, despite having a low M value. For the cohesiveness parameter, behaviors with high cohesion, States B, C, and E, are found to be the same. State D, even though it has a small measure of C, appears closer to B, C, and E, likely due to its clustering behavior on a linear path. Importantly, when using all three global order parameters, EUGENE is able to distinguish that all the behavioral states are different (minimum value for off-diagonal entries > 0.5), which we expect to be true by design. However, these behaviors might be perceived differently if considered from the point of view of an individual rather than a group perspective.

When considering the local interactions, an individual might interpret the polarization of the agents around it much differently than if it could see the entirety of the group. EUGENE is able to detect these differences in the distance matrix for local polarization P^L. Similarly to the global polarization, States A and B are the same as each other, and likewise for States D and E. State C, on the other hand, is now more similar to D and E. The difference in the local and global polarization parameter is seen in figure 3, where State C distinctly changes from near zero, globally, to one, locally. The switch in similarity is likely because an agent that is milling on a circle, as in State C, would see itself as being more aligned with the individuals nearest to it. In the local angular momentum M^L, the individual would have more difficulty detecting its angular momentum without knowing the entirety of the group, and thus EUGENE finds that all the behavioral states are nearly the same. For local cohesion C^L, the result is similar with the behaviors having pairwise distances close to zero. Finally, when considering all three local order parameters, the distance matrix shows similar structures to the local polarization plot, which means the local interactions are dominated by the P^L parameter.

Notably, EUGENE is able to distinguish between the five states using only low-dimensional descriptors, rather than the high-dimensional state of the system. However, even though we select the previously defined order parameters as input, the results are not identical to the binary distinctions of steady-state high or low values that are usually used to classify these parameters. This suggests that some of the emergent dynamics which occur in the transient may be different or absent when only the

asymptotic behavior of the system is considered. Since animal group behavior may arguably never reach steady-state, EUGENE may give a more realistic classification of these behaviors when applied to biological data. We note that the absolute distances when local parameters are used are low overall, and that the states are generally grouped into two kinds. This suggests that from the perspective of a single agent, these behaviors are not distinct, which is understandable when superimposing a sensing space (circle of radius 1) on any of the agents in figure 2. For recovering these different group-level behaviors, other inputs for EUGENE should be considered.

This work represents a first exploration of causal discovery to understand qualitatively different behaviors using a model-agnostic method for classifying the underlying dynamics driving the system. Here, we show that states known to be different can be distinguished from their time series data. In future work, this method will be used to study data from biological systems, whose dynamics may be governed by observable and unobservable variables, such as physiology and sociality, respectively.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation under grants CMMI-1708622 and SES-1454190, by the Institute for Critical Technology and Applied Science at Virginia Tech, and by the Graduate School at Virginia Tech through a Cunningham Doctoral Scholarship to Amanda Hashimoto.

REFERENCES

[1] Krause, J., and Ruxton, G. D., 2002. *Living in groups*. Oxford University Press.

[2] Aranson, I. S., and Tsimring, L. S., 2006. "Patterns and collective behavior in granular media: Theoretical concepts". *Reviews of Modern Physics,* ***78***(2), p. 641.

[3] Elgeti, J., Winkler, R. G., and Gompper, G., 2015. "Physics of microswimmers-single particle motion and collective behavior: a review". *Reports on Progress in Physics,* ***78***(5), p. 056601.

[4] Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M., 2013. "Swarm robotics: a review from the swarm engineering perspective". *Swarm Intelligence,* ***7***(1), pp. 1–41.

[5] Helbing, D., 2012. "Agent-based modeling". In *Social self-organization*. Springer, pp. 25–70.

[6] Topaz, C. M., Bertozzi, A. L., and Lewis, M. A., 2006. "A nonlocal continuum model for biological aggregation". *Bulletin of Mathematical Biology,* ***68***(7), p. 1601.

[7] Couzin, I. D., Krause, J., James, R., Ruxton, G. D., and Franks, N. R., 2002. "Collective memory and spatial sorting in animal groups". *Journal of Theoretical Biology,* ***218***(1), pp. 1–11.

[8] Werfel, J., Petersen, K., and Nagpal, R., 2014. "Designing collective behavior in a termite-inspired robot construction team". *Science,* ***343***(6172), pp. 754–758.

[9] Aureli, M., and Porfiri, M., 2010. "Coordination of self-propelled particles through external leadership". *EPL (Europhysics Letters),* ***92***(4), p. 40004.

[10] Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., and Shochet, O., 1995. "Novel type of phase transition in a system of self-driven particles". *Physical Review Letters,* ***75***(6), p. 1226.

[11] Butail, S., Ladu, F., Spinello, D., and Porfiri, M., 2014. "Information flow in animal-robot interactions". *Entropy,* ***16***(3), pp. 1315–1330.

[12] Butail, S., Mwaffo, V., and Porfiri, M., 2016. "Model-free information-theoretic approach to infer leadership in pairs of zebrafish". *Physical Review E,* ***93***(4), p. 042411.

[13] Lord, W. M., Sun, J., Ouellette, N. T., and Bollt, E. M., 2016. "Inference of causal information flow in collective animal behavior". *IEEE Transactions on Molecular, Biological and Multi-Scale Communications,* ***2***(1), pp. 107–116.

[14] Dost, F., 2015. "A non-linear causal network of marketing channel system structure". *Journal of Retailing and Consumer Services,* ***23***, pp. 49–57.

[15] McBride, J. C., Zhao, X., Munro, N. B., Jicha, G. A., Schmitt, F. A., Kryscio, R. J., Smith, C. D., and Jiang, Y., 2015. "Sugihara causality analysis of scalp EEG for detection of early Alzheimer's disease". *NeuroImage: Clinical,* ***7***, pp. 258–265.

[16] Kiourktsidis, J. L., and Koulouras, G., 2011. "Adaptive routing provision by using Bayesian inference". *Mobile Ad Hoc Networks: Current Status and Future Trends*, p. 467.

[17] Oakes, J. M., 2004. "The (mis) estimation of neighborhood effects: causal inference for a practicable social epidemiology". *Social Science & Medicine,* ***58***(10), pp. 1929–1952.

[18] Jantzen, B. C., 2017. "Dynamical Kinds and their Discovery". *Proceedings of the UAI 2016 Workshop on Causation: Foundation to Application*. arXiv: 1612.04933.

[19] Jantzen, B. C., 2017. The EUGENE project. *Available at* https://github.com/jantzen/eugene (accessed 9 March 2018).

[20] Schaub, M. T., Delvenne, J.-C., Lambiotte, R., and Barahona, M., 2018. "Multiscale dynamical embeddings of complex networks". *arXiv preprint arXiv:1804.03733*.

[21] Sugihara, G., May, R., Ye, H., Hsieh, C.-h., Deyle, E., Fogarty, M., and Munch, S., 2012. "Detecting causality in complex ecosystems". *Science*, p. 1227079.

[22] Jantzen, B. C., 2014. "Projection, symmetry, and natural kinds". *Synthese,* ***192***(11), pp. 3617–3646.

[23] Rizzo, M. L., and Székely, G. J., 2016. "Energy distance". *Wiley Interdisciplinary Reviews: Computational Statistics,* ***8***(1), Jan., pp. 27–38.

Chapter 5

Summary

In this work, I have introduced a novel algorithm for differentiating dynamical kinds among complex systems. I have shown the algorithm's performance as applied to various constructions of the Lotka-Volterra and Kuramoto systems of equations, under circumstances where control and observability of the system varied greatly. Further, the algorithm has been shown to be able to detect shifts in dynamical kinds within a system that is only being passively observed. Finally, I presented an application of the algorithm as a tool for dynamic systems analysis.

Given the wide applicability of the EUGENE algorithms, and their unique focus on an under-explored approach to system analysis, I believe they have an opportunity to motivate some very exciting research. The use of dynamical distance as a target metric for control problems also seems promising - being more robust than optimizing over the norm of a trajectory, and more principled than supervised learning. In a similar vein as chapter 4, I expect dynamical similarity to find use as a tool to explore systems with complex, emergent behavior, and as a tool to validate models meant to reflect such systems. With this metric's sensitivity to chaos in mind, we may also use it as a tool for validating models of chaotic behavior, such as the reservoir computers used in [10] to model the Kuramoto-Sivashinsky equation.

Many significant improvements can be made to the algorithms as well. Subroutines like clipping, and initial condition selection might benefit from a less data-hungry approach; trading computational efficiency for data efficiency. Further, estimations might be used in

place of some more computationally complex procedures being performed as part of the distance calculations. I am also interested in an attempt to train a model on data labeled with a kind to test the effectiveness of a supervised kind classifier. Finally, with appreciation for the number of applications for more standard distance metrics, I'm curious how dynamical distance might perform in the place of other metrics, and what we might learn from doing so.

Bibliography

[1] Rudolf Beran et al. Minimum hellinger distance estimates for parametric models. *The annals of Statistics*, 5(3):445–463, 1977.

[2] Benjamin C. Jantzen. Projection, symmetry, and natural kinds. *Synthese*, 192(11): 3617–3646, 2014. ISSN 0039-7857, 1573-0964. doi: 10.1007/s11229-014-0637-5. URL http://link.springer.com/article/10.1007/s11229-014-0637-5.

[3] Benjamin C. Jantzen. The EUGENE project, 2017. URL https://github.com/jantzen/eugene. *Available at* https://github.com/jantzen/eugene (accessed 9 March 2018).

[4] Benjamin C. Jantzen. Dynamical Kinds and their Discovery. *Proceedings of the UAI 2016 Workshop on Causation: Foundation to Application*, 2017. URL http://ceur-ws.org/Vol-1792/paper2.pdf. arXiv: 1612.04933.

[5] Benjamin C Jantzen. Dynamical symmetries and model validation. In *Algorithms and Complexity in Mathematics, Epistemology, and Science*, pages 153–176. Springer, 2019.

[6] Yoshiki Kuramoto. International symposium on mathematical problems in theoretical physics. *Lecture notes in Physics*, 30:420, 1975.

[7] Pat Langley, Gary L Bradshaw, and Herbert A Simon. Rediscovering chemistry with the bacon system. In *Machine learning*, pages 307–329. Springer, 1983.

[8] T Warren Liao. Clustering of time series data—a survey. *Pattern recognition*, 38(11): 1857–1874, 2005.

[9] Meng Liu and Meng Fan. Permanence of stochastic lotka–volterra systems. *Journal of Nonlinear Science*, 27(2):425–452, 2017.

[10] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical review letters*, 120(2):024102, 2018.

[11] Maria L. Rizzo and Gábor J. Székely. Energy distance. *Wiley Interdisciplinary Reviews: Computational Statistics*, 8(1):27–38, January 2016. ISSN 19395108. doi: 10.1002/wics.1375. URL http://doi.wiley.com/10.1002/wics.1375.

[12] Subhradeep Roy and Benjamin Jantzen. Detecting causality using symmetry transformations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7):075305, 2018.

[13] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[14] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1): 43–49, February 1978. ISSN 0096-3518. doi: 10.1109/TASSP.1978.1163055.

[15] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

[16] Michael Schmidt and Hod Lipson. Symbolic regression of implicit equations. In *Genetic Programming Theory and Practice VII*, pages 73–85. Springer, 2010.

[17] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[18] u/PartyLikeLizLemon and Yann LeCun. The future (and present) of artificial intelligence ama, Feb 2018. URL https://www.reddit.com/r/science/comments/7yegux/aaas_ama_hi_were_researchers_from_google/duglcrh/.

[19] Vito Volterra. Fluctuations in the abundance of a species considered mathematically, 1926.

[20] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 1317–1322. IEEE, 2016.

Appendix A

Distance Characterization

To better understand the behaviors of the metric for dynamical similarity presented in this work, I developed and executed more than 30 tests. Some of these are presented in chapter 3, and the rest shall be presented here to provide a deeper demonstration of the algorithm.

A.1 Lotka-Volterra Equations

The tests presented here use simulations of one of two versions of the Lotka-Volterra equations. The most commonly used system is the general Lotka-Volterra equation [19]:

$$\frac{dx_i}{dt} = r_i x_i \Big(1 - \frac{\sum_{j=0}^{n} a_{ij} x_j}{k_i}\Big), i = 0...n \tag{A.1}$$

where x_i represents the population of species i, k_i represents the carrying capacity of that species, r_i represents the growth rate of the species, and a_{ij} is the interaction coefficient of species j on species i.

The more specialized stochastic Lotka-Volterra equation modified from [9] is:

$$\Delta x_i(t) = x_i(t) r_i(t) \Big(1 - \sum_{j=0}^{n} a_{ij} x_j(t)\Big) \Delta t + \sigma_i x_i(t) \xi_i(t) \sqrt{\Delta t} + \frac{\sigma_i^2}{2} x_i(t) \Big((\xi_i(t))^2 - 1\Big) \Delta t \tag{A.2}$$

where the magnitude of the noise for species i is given by σ_i, and ξ_i is a random sample from

the distribution $\mathcal{N}(0, 1)$.

A.2 Test Framework

In order to make comparisons across tests, most test parameters were kept as similar as possible. Unless noted otherwise, assume the following hold true: Systems were simulated until $t = 5$, with $\Delta t = 0.01$. The received data set is non-SSD.

System A:

$$N = 2, A_A = \begin{bmatrix} 1 & 0.5 \\ 0.7 & 1 \end{bmatrix}, R_A = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, K_A = \begin{bmatrix} 100 \\ 100 \end{bmatrix}$$

System B: If system B is noted to be of the same kind as system A, then:

$$N = 2, A_B = \begin{bmatrix} 1 & 0.5 \\ 0.7 & 1 \end{bmatrix}, R_B = \begin{bmatrix} 1.5 \\ 3 \end{bmatrix}, K_B = \begin{bmatrix} 100 \\ 100 \end{bmatrix}$$

Otherwise, if systems A and B are noted to be of different kinds, then:

$$N = 2, A_B = \begin{bmatrix} 1 & 0.5 \\ 0.7 & 1 \end{bmatrix}, R_B = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, K_B = \begin{bmatrix} 150 \\ 150 \end{bmatrix}$$

If the systems are stochastic, then the number of replicates to generate is expected to be 10, and:

$$\sigma_A = \sigma_B = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$$

Note: most of these default values were chosen arbitrarily to provide an interesting, but common system.

A.3 Test Results

Results are grouped by their similarity to each other, and by complement.

A.3.1 Comparing Deterministic Systems as Data Increases

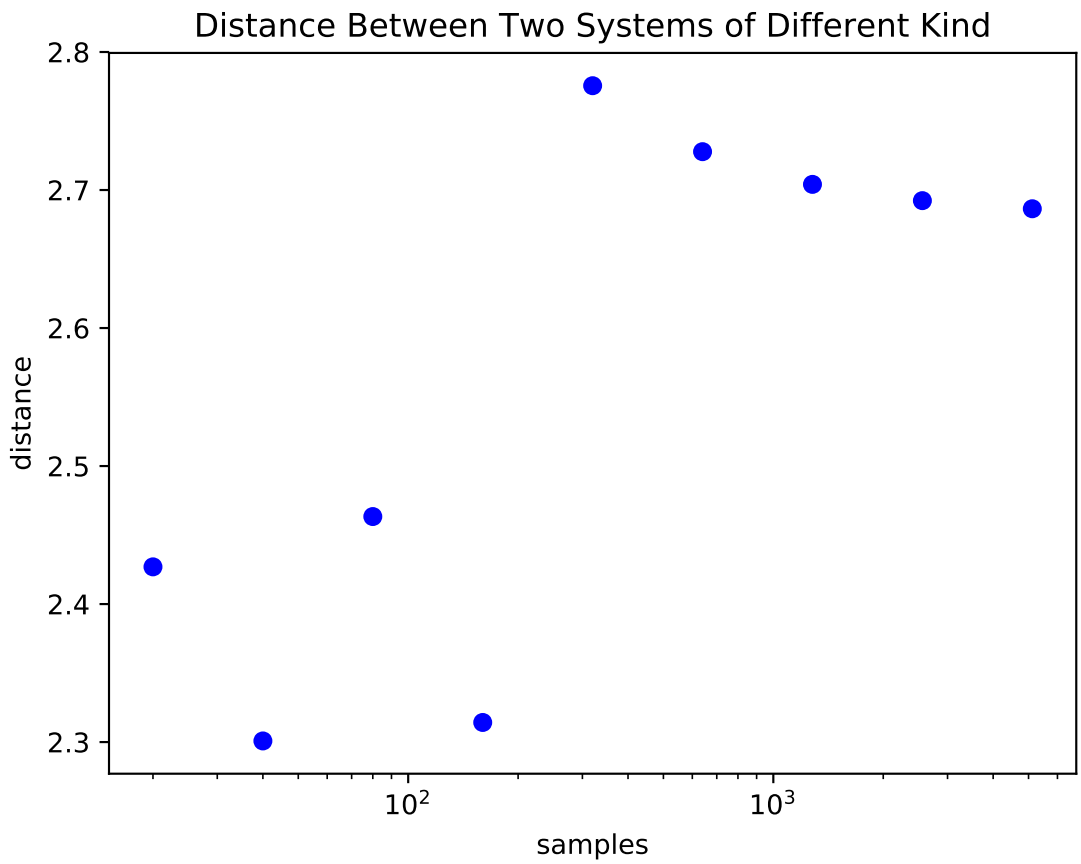

Figure A.1: Dynamical distance between system A and a different system B as sampling rate increases.

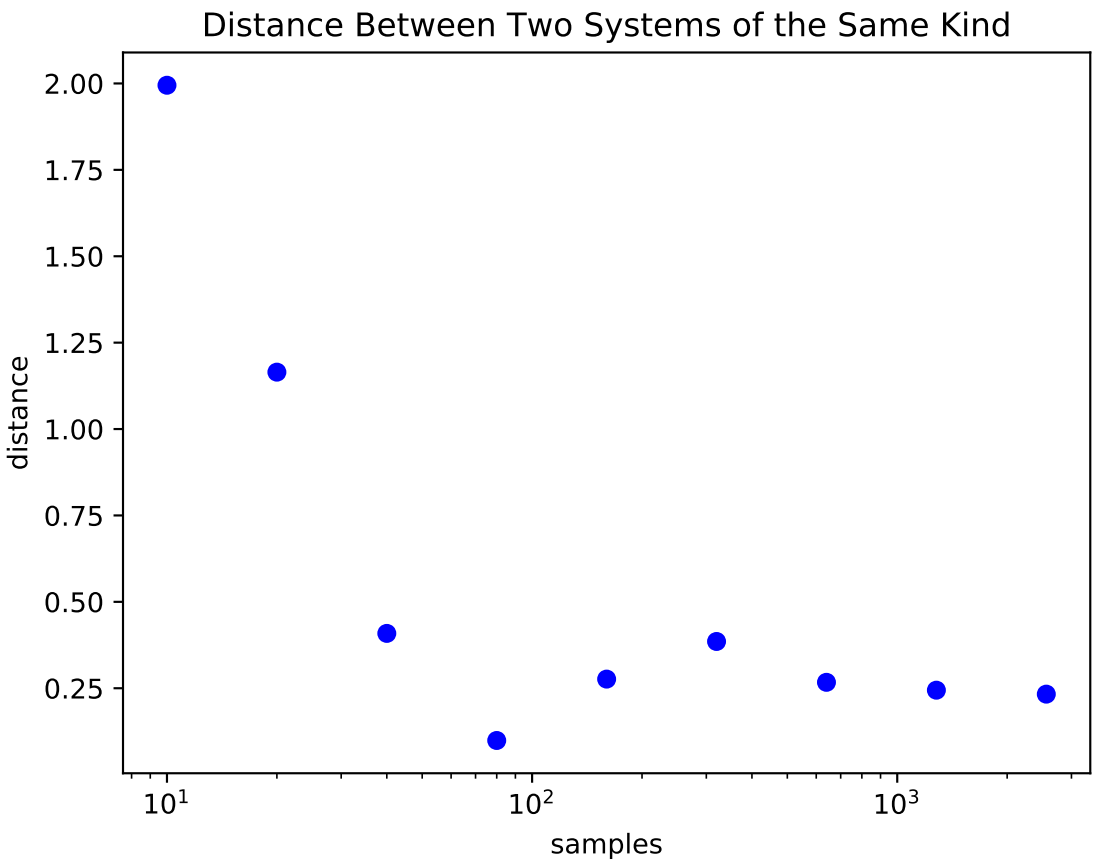

Figure A.2: Dynamical distance between system A and a similar system B as sampling rate increases.

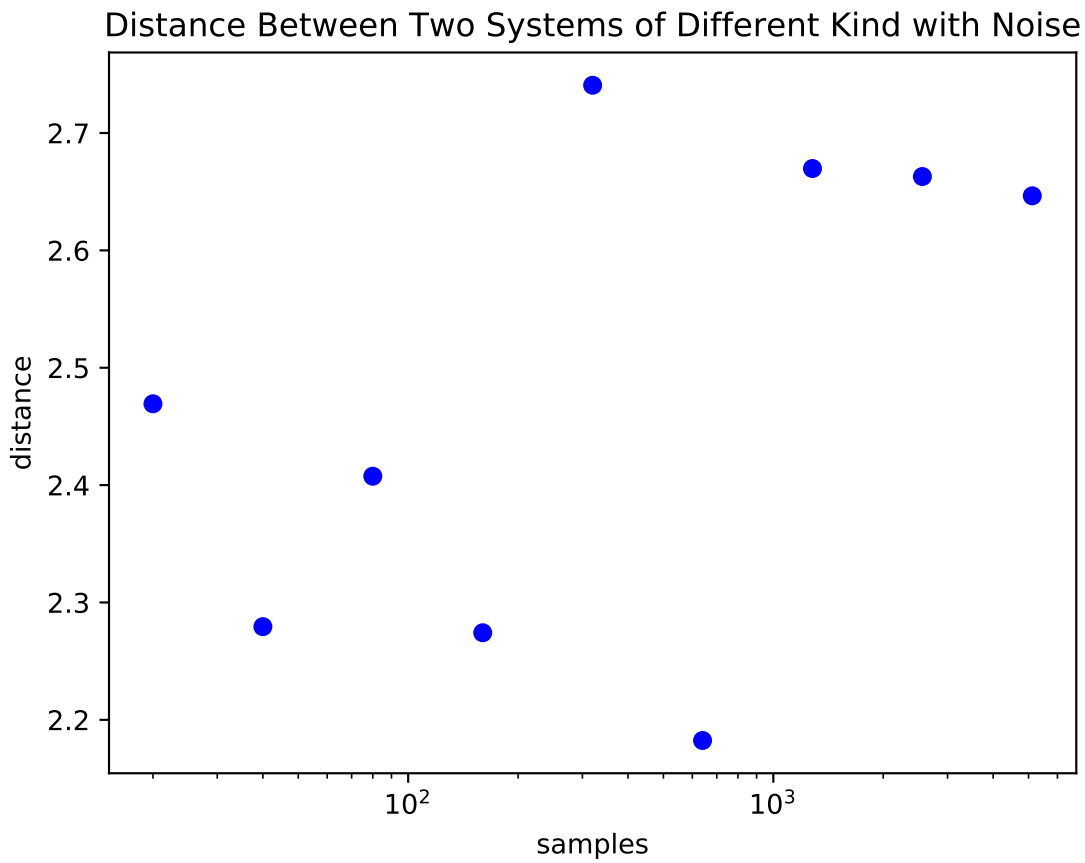

Figure A.3: Dynamical distance between system A and a different system B as sampling rate increases, and measurement noise is included.

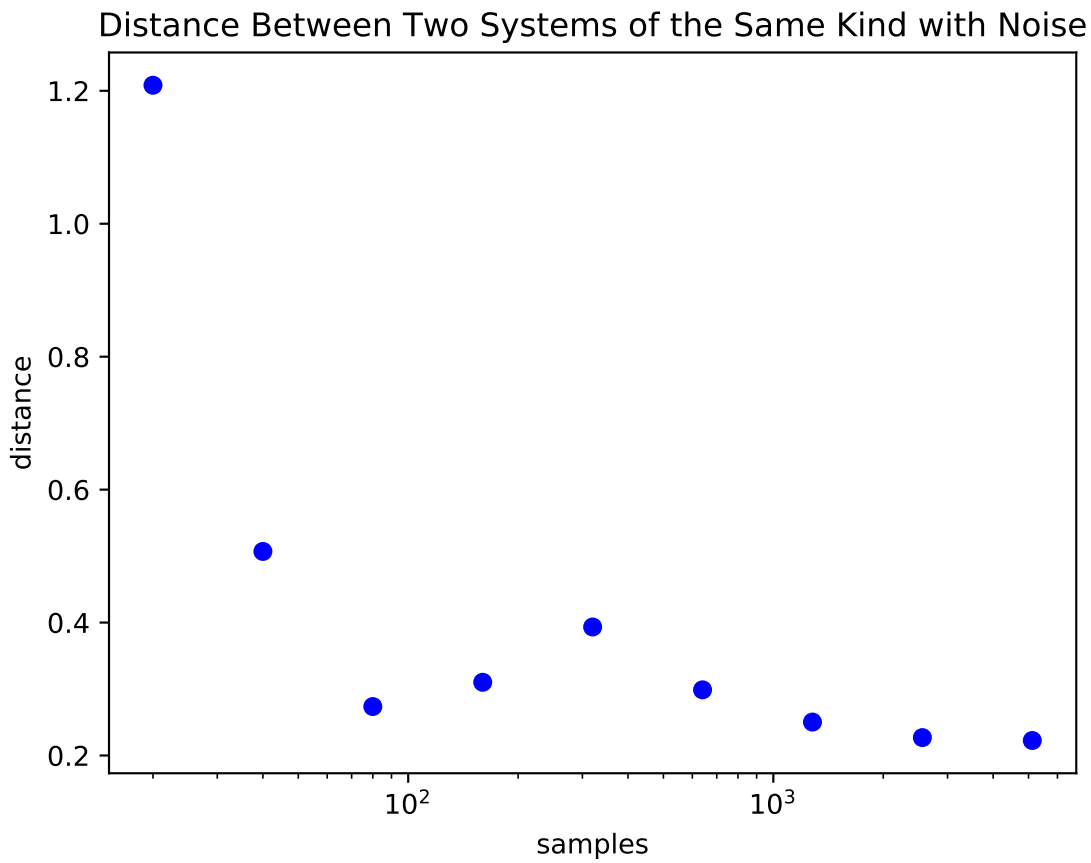

Figure A.4: Dynamical distance between system A and a similar system B as sampling rate increases, and measurement noise is included.

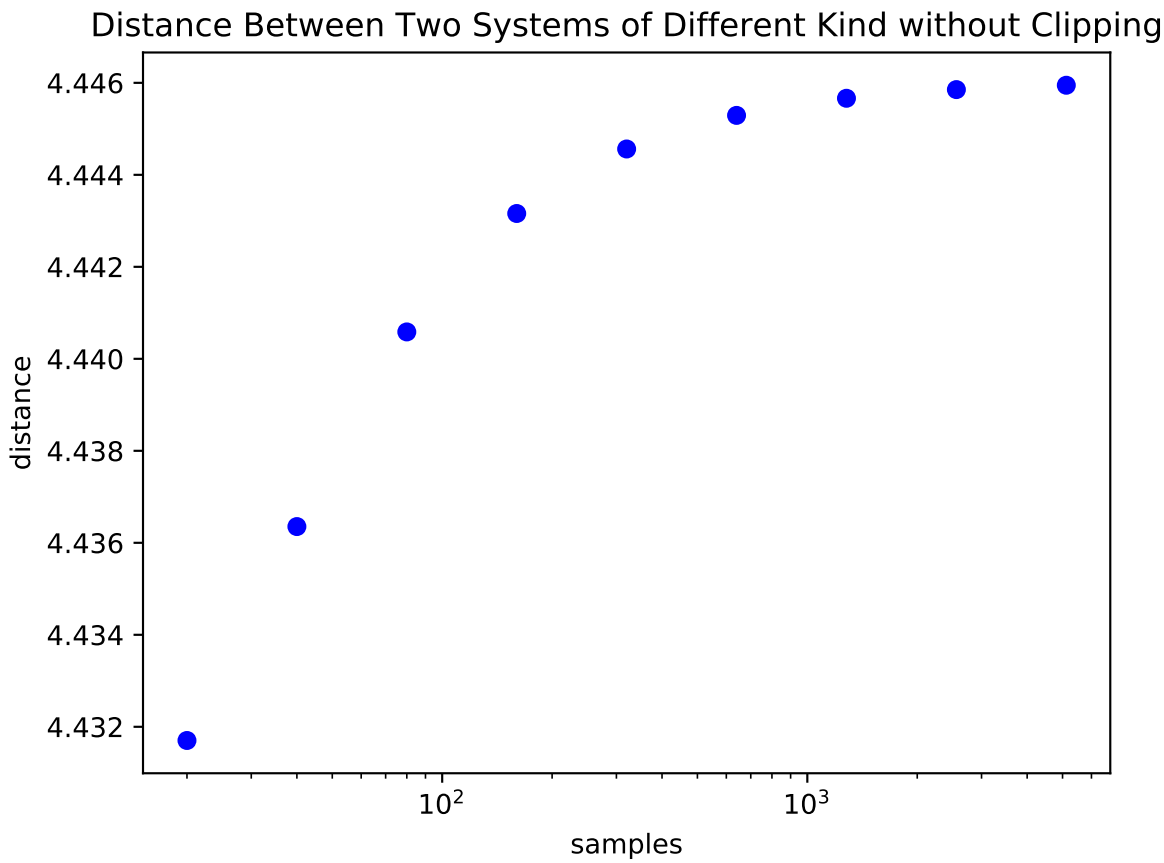

Figure A.5: Dynamical distance between system A and a different system B as sampling rate increases, and clipping is not used.

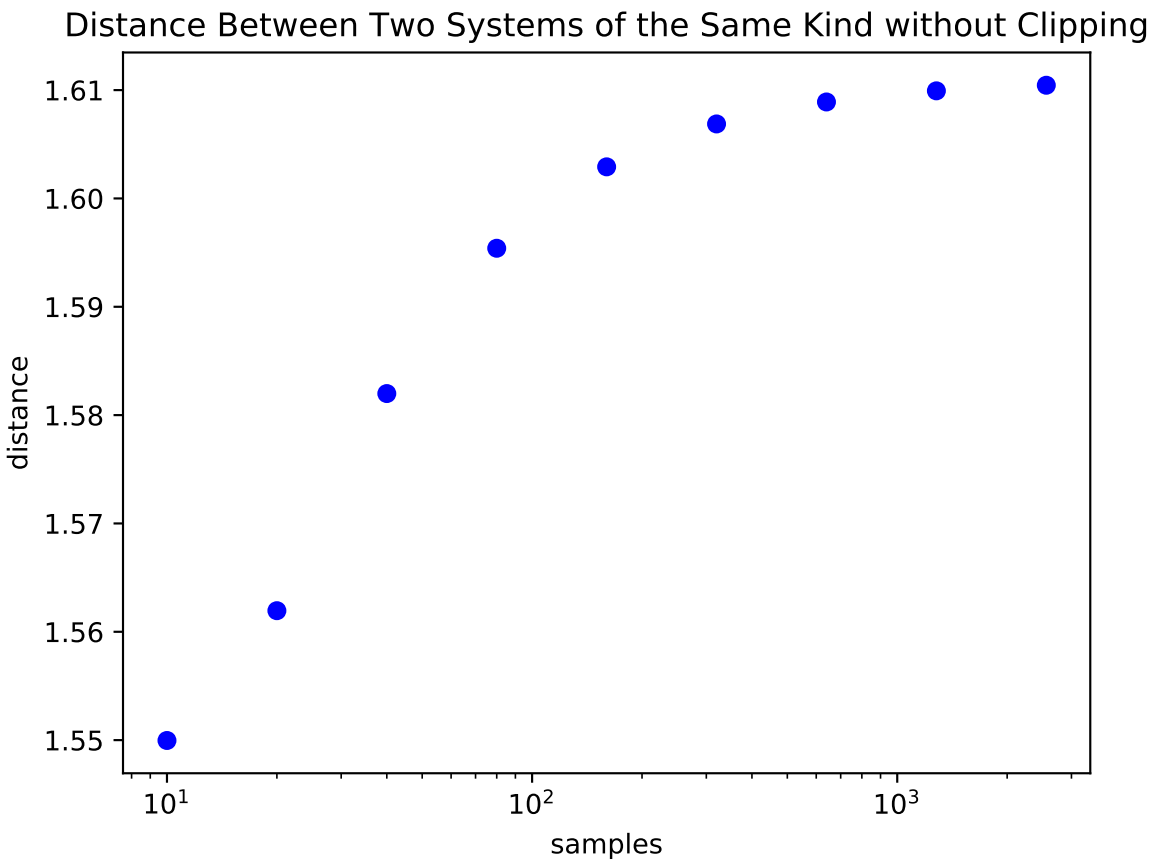

Figure A.6: Dynamical distance between system A and a similar system B as sampling rate increases, and clipping is not used.

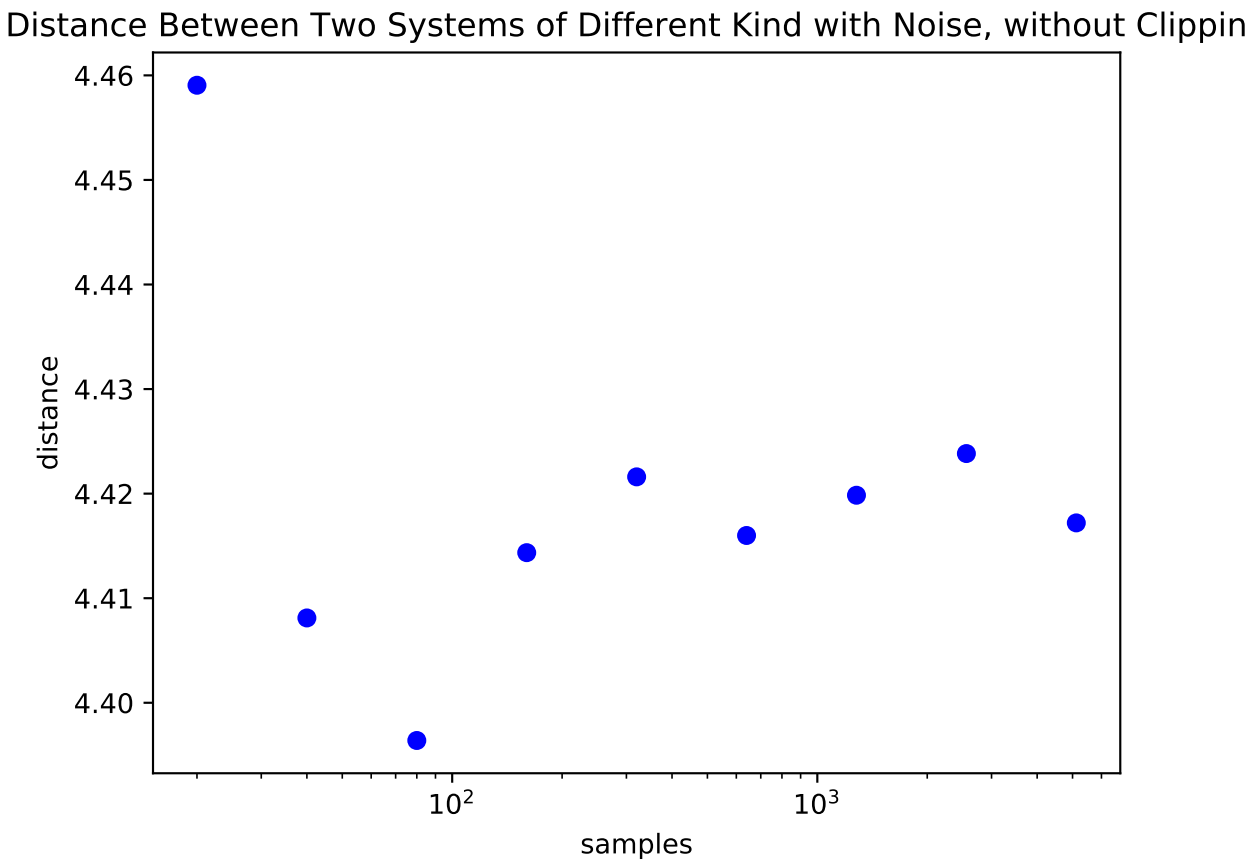

Figure A.7: Dynamical distance between system A and a different system B as sampling rate increases, measurement noise is included, and clipping is not used.

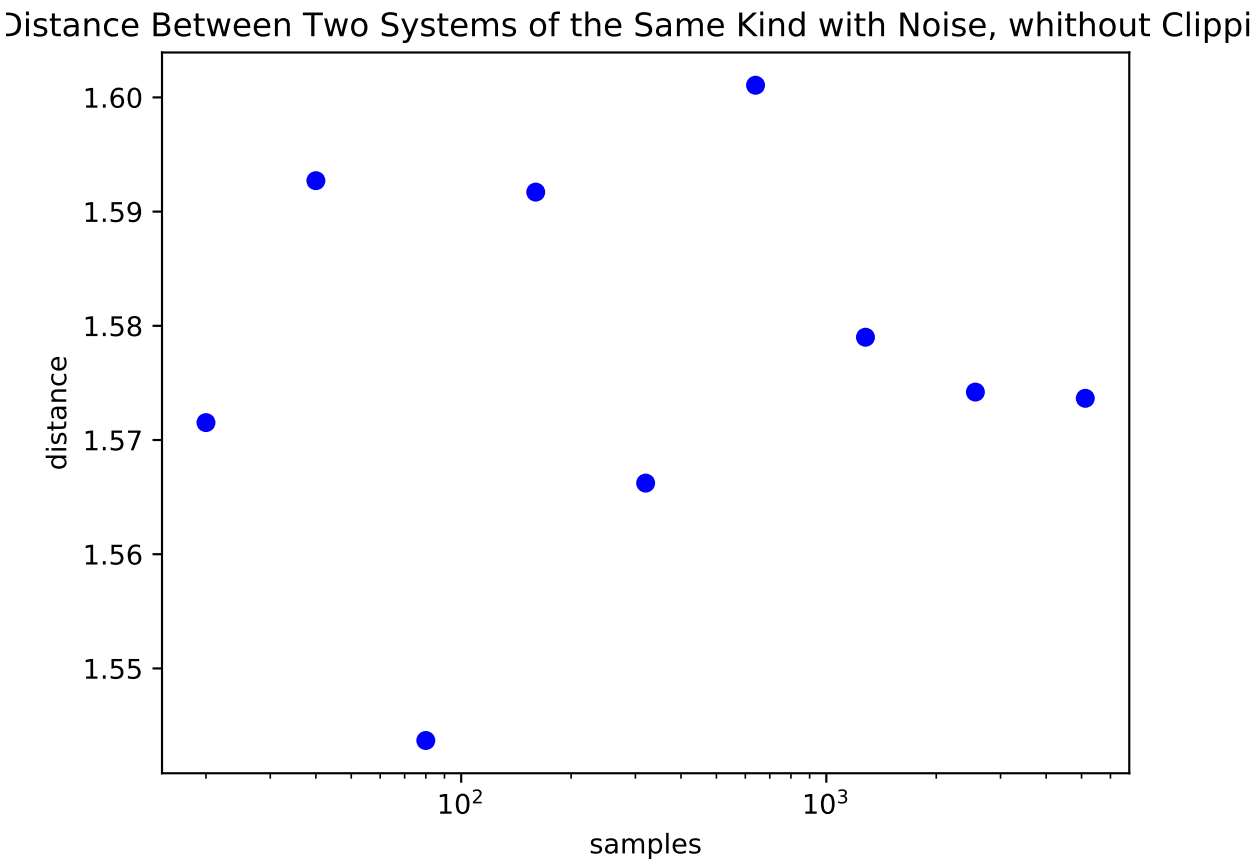

Figure A.8: Dynamical distance between system A and a similar system B as sampling rate increases, measurement noise is included, and clipping is not used.

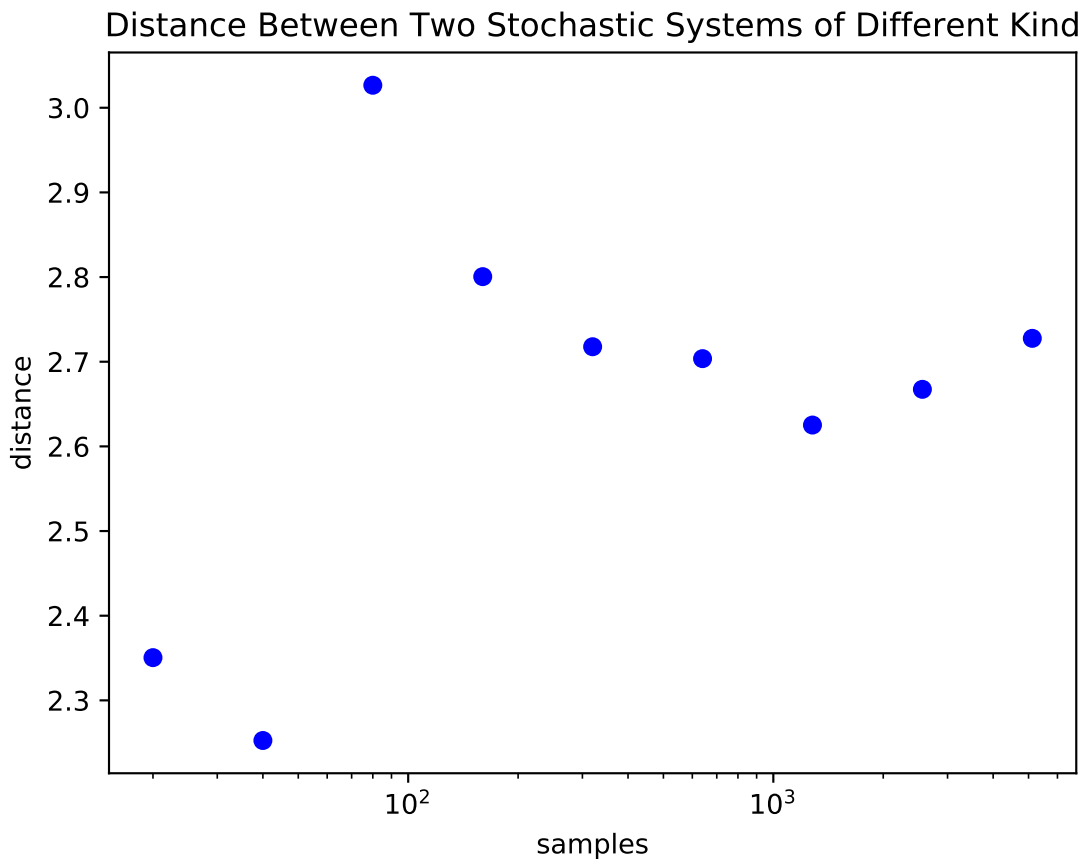

Figure A.9: Dynamical distance between system A and a different system B as sampling rate increases.

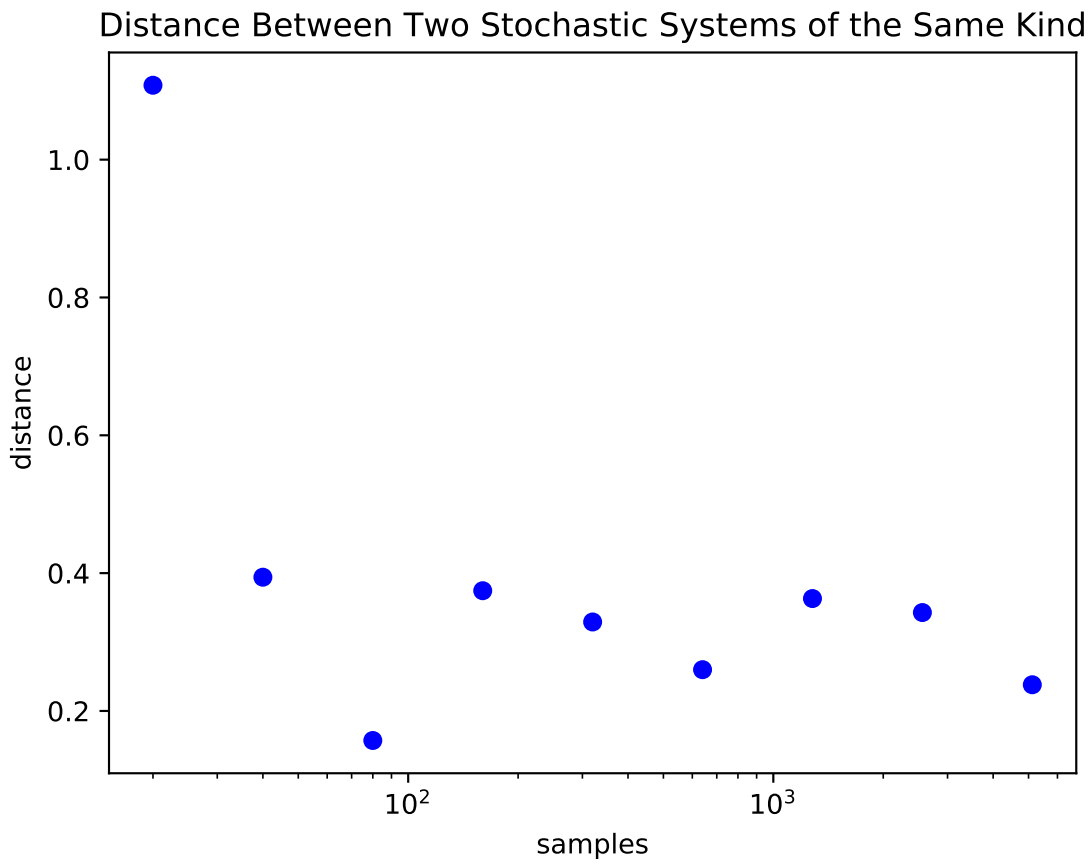

Figure A.10: Dynamical distance between system A and a similar system B as sampling rate increases.

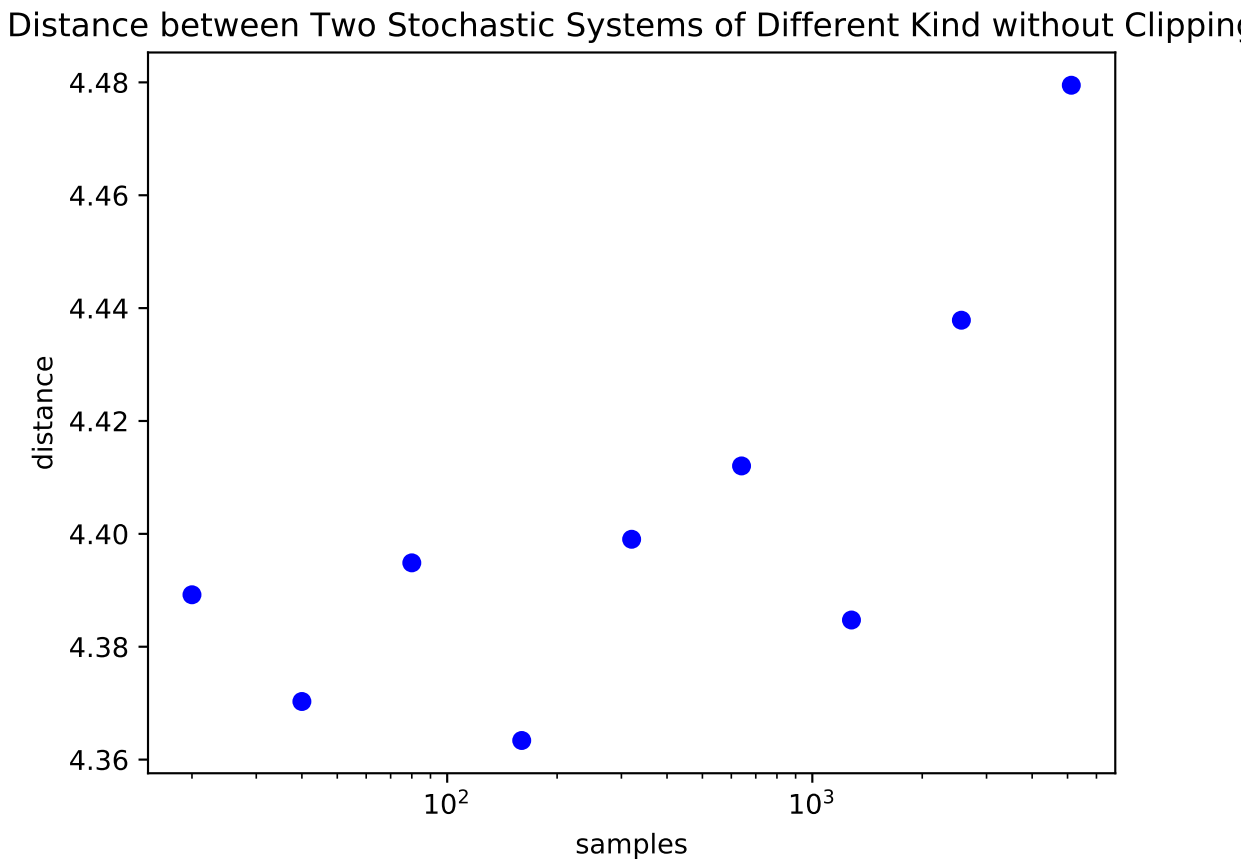

Figure A.11: Dynamical distance between system A and a different system B as sampling rate increases, without clipping.

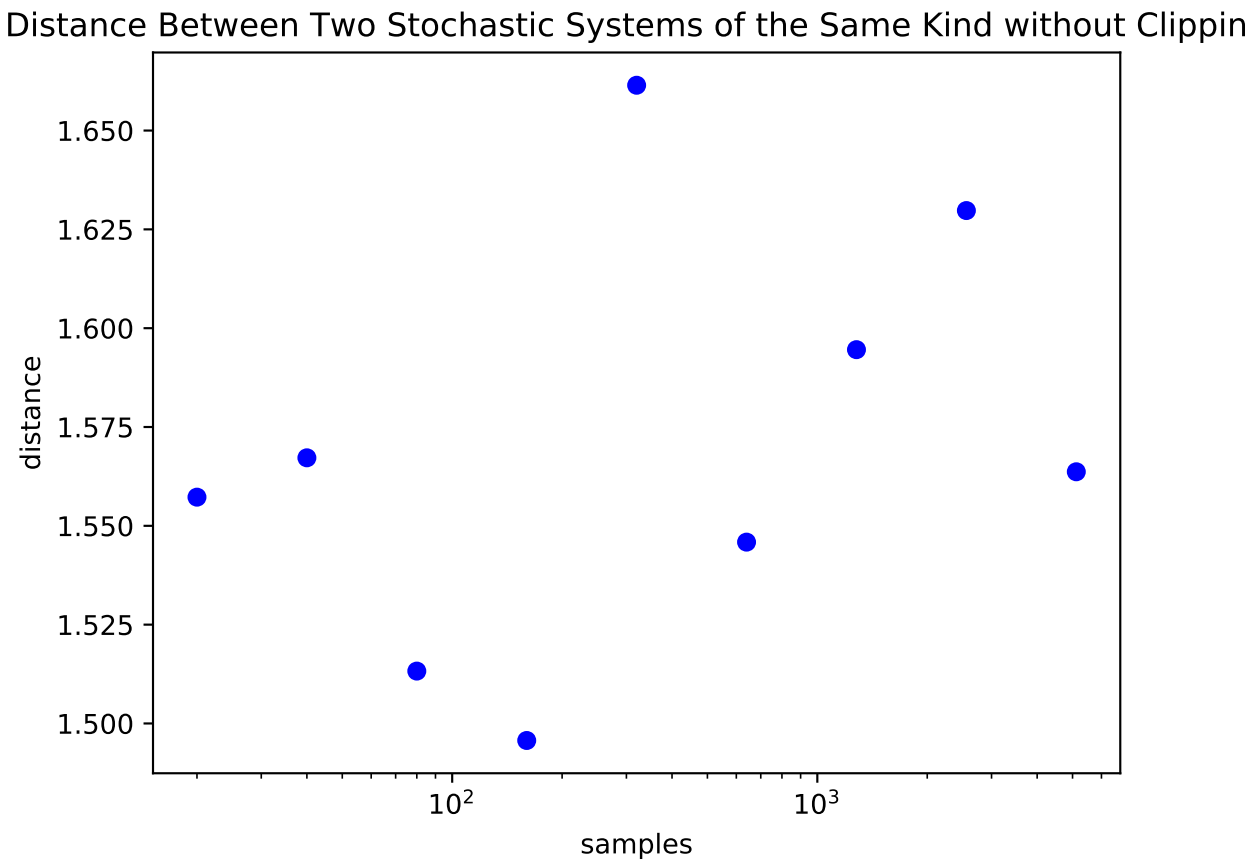

Figure A.12: Dynamical distance between system A and a similar system B as sampling rate increases, without clipping.

A.3.2 Comparing Stochastic Systems as Data Increases

A.3.3 Comparing Stochastic Systems as Number of Replicates Increases

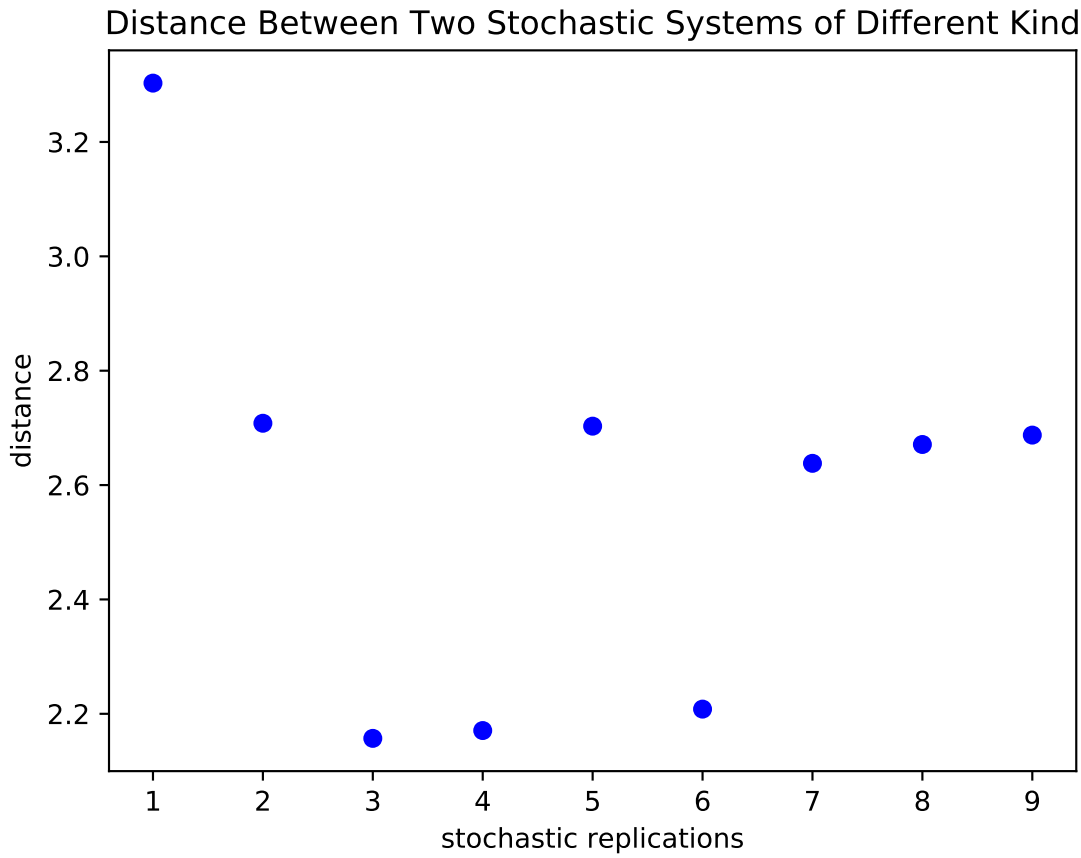

Figure A.13: Dynamical distance between system A and a different system B as number of replicates increases.

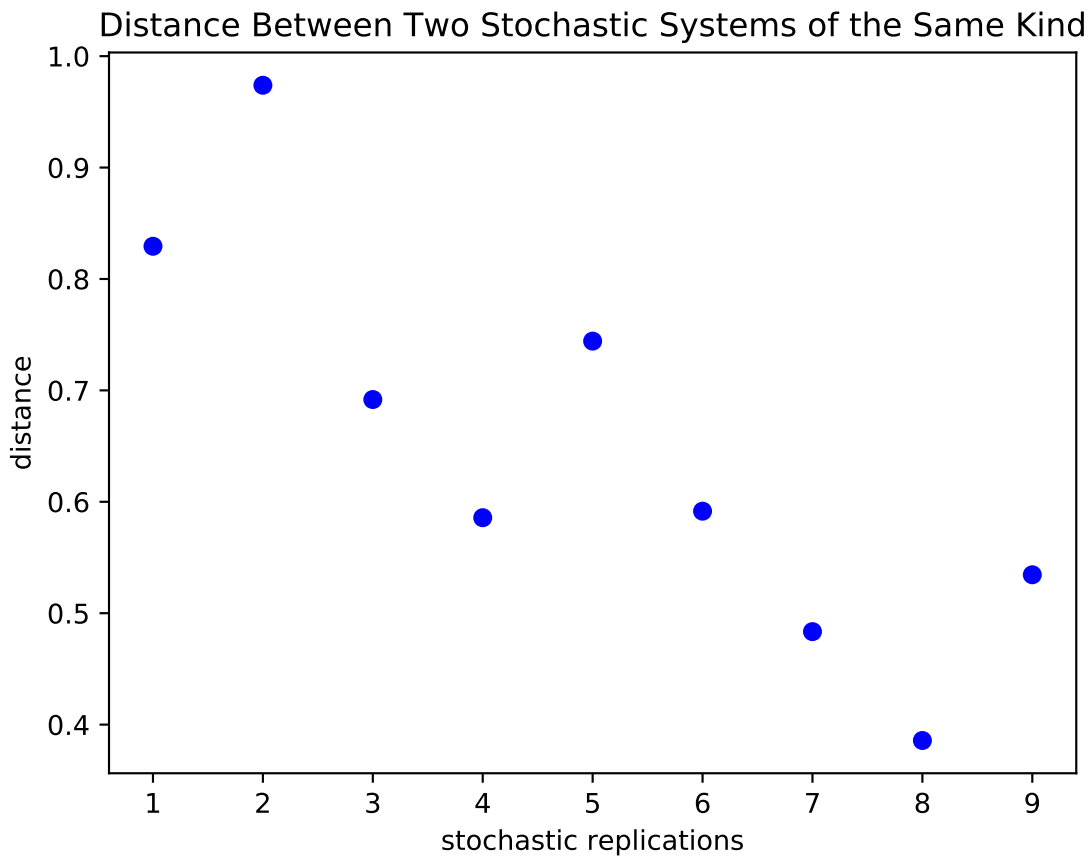

Figure A.14: Dynamical distance between system A and a similar system B as number of replicates increases.

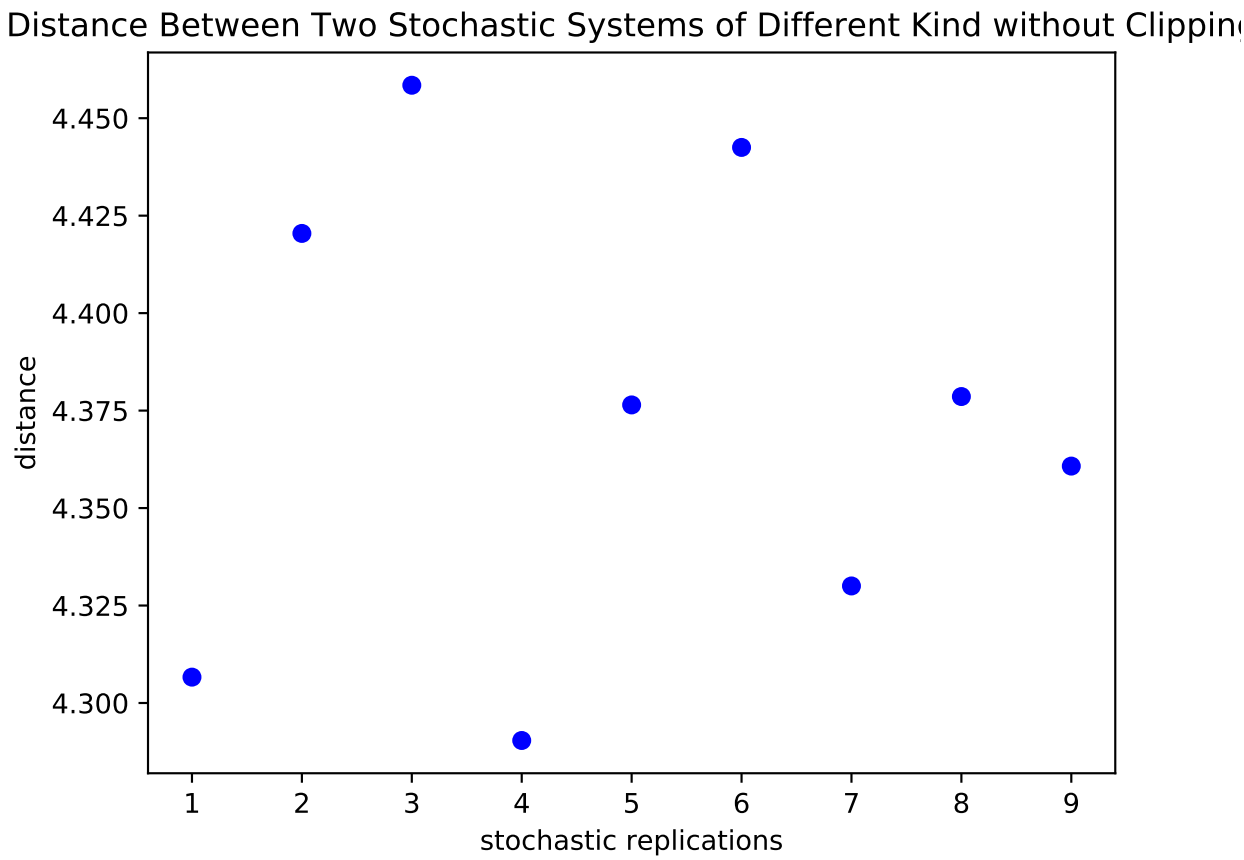

Figure A.15: Dynamical distance between system A and a different system B as number of replicates increases, without clipping.

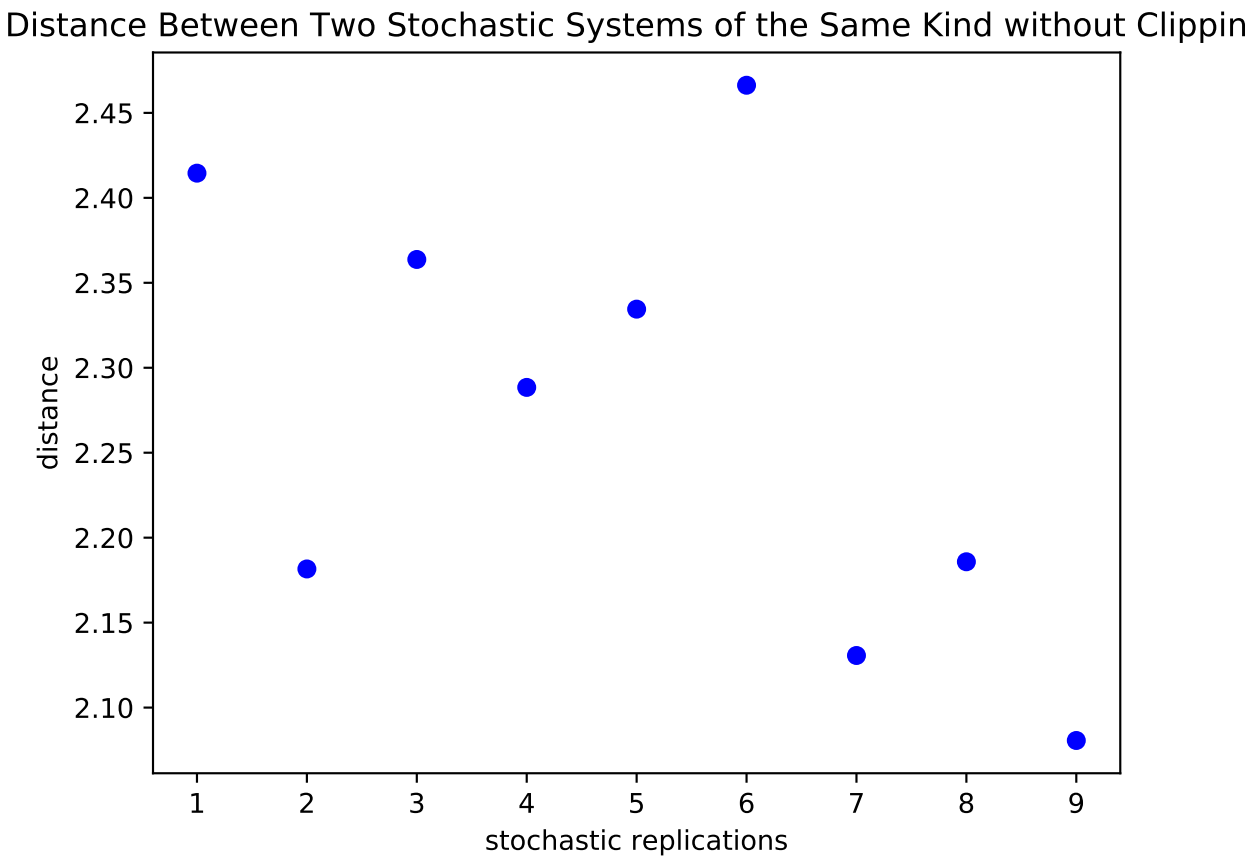

Figure A.16: Dynamical distance between system A and a similar system B as number of replicates increases, without clipping.

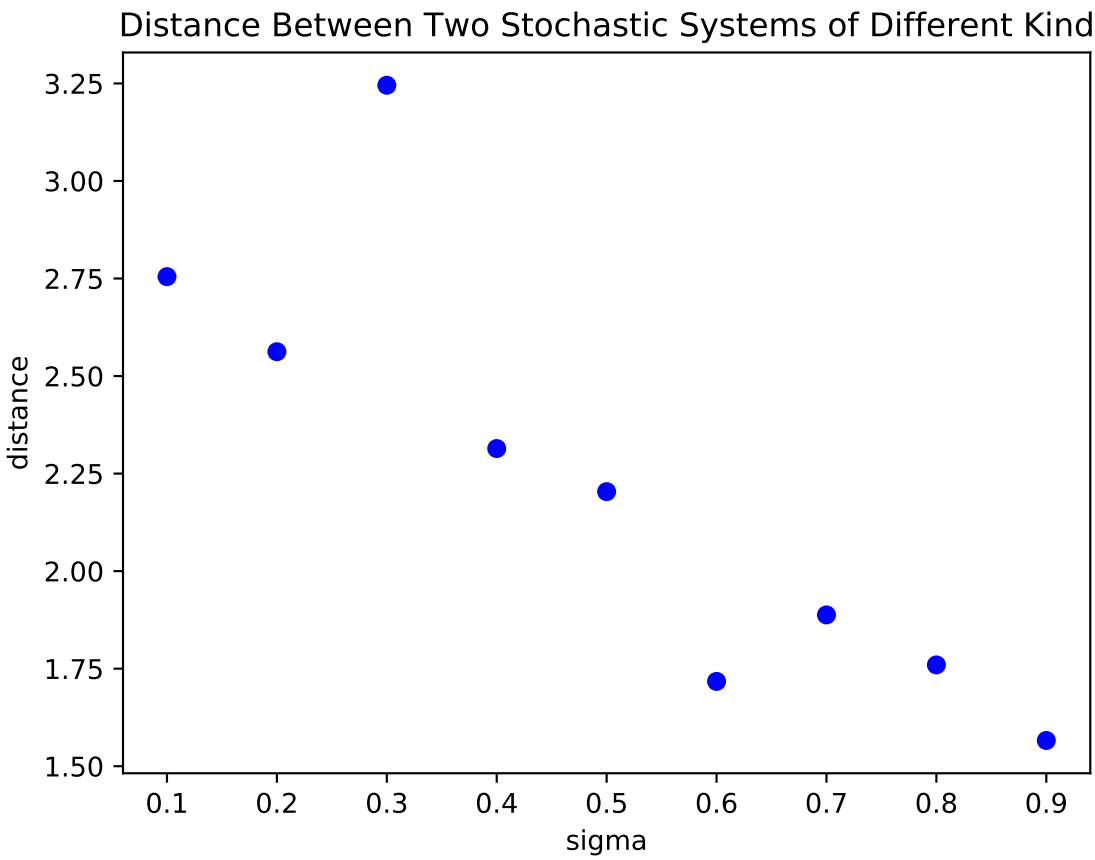

Figure A.17: Dynamical distance between system A and a different system B as magnitude of stochastic noise increases.

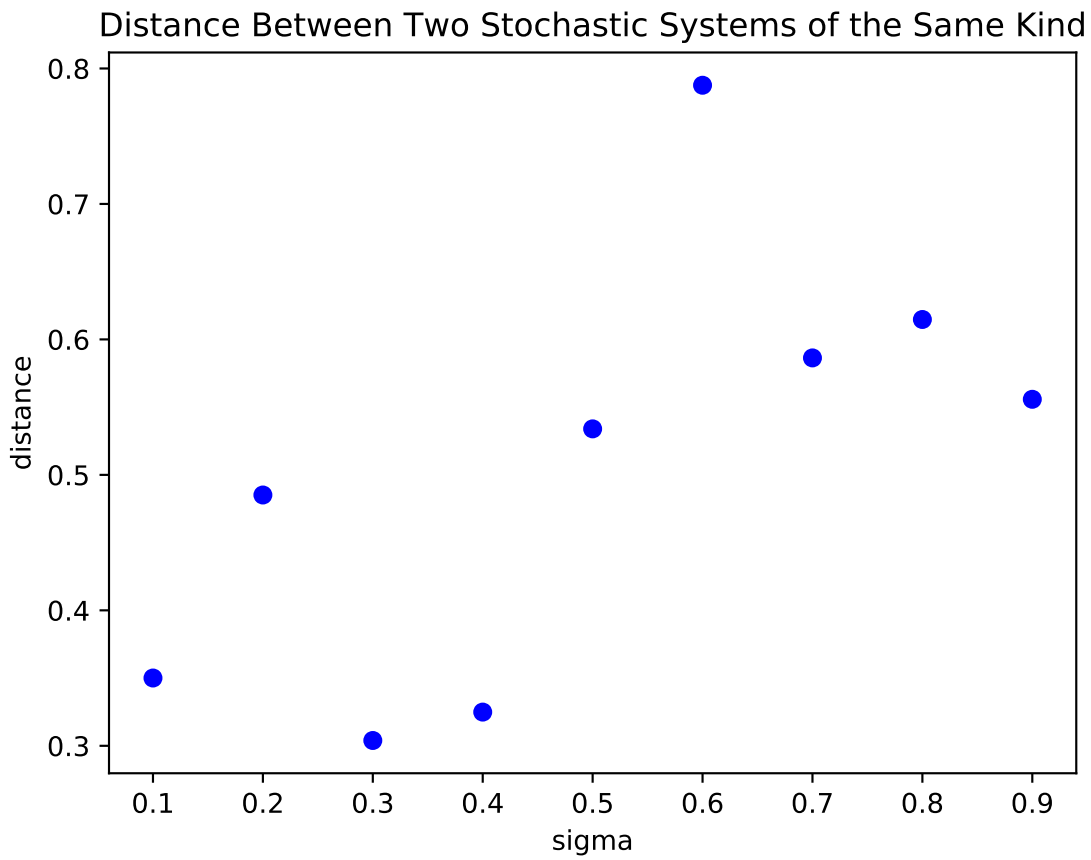

Figure A.18: Dynamical distance between system A and a similar system B as magnitude of stochastic noise increases.

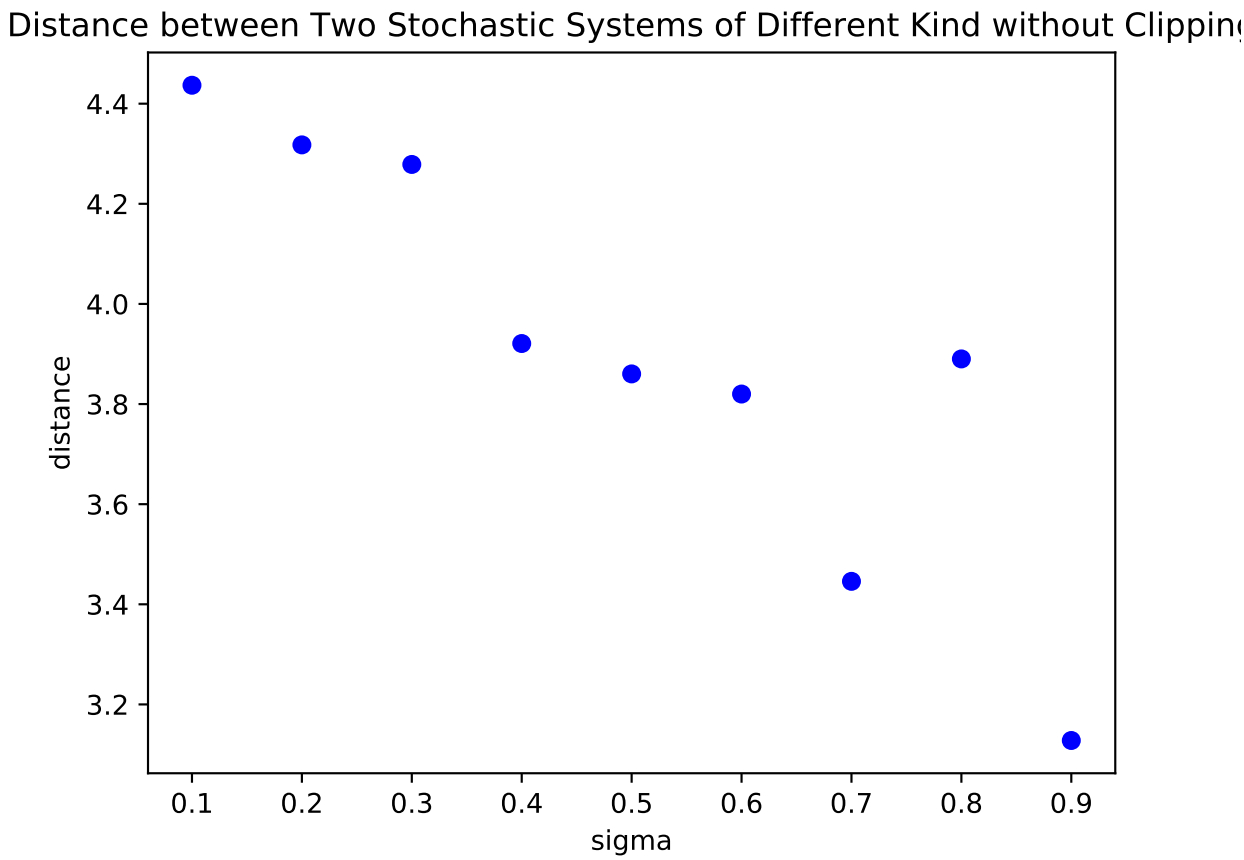

Figure A.19: Dynamical distance between system A and a different system B as magnitude of stochastic noise increases, without clipping.

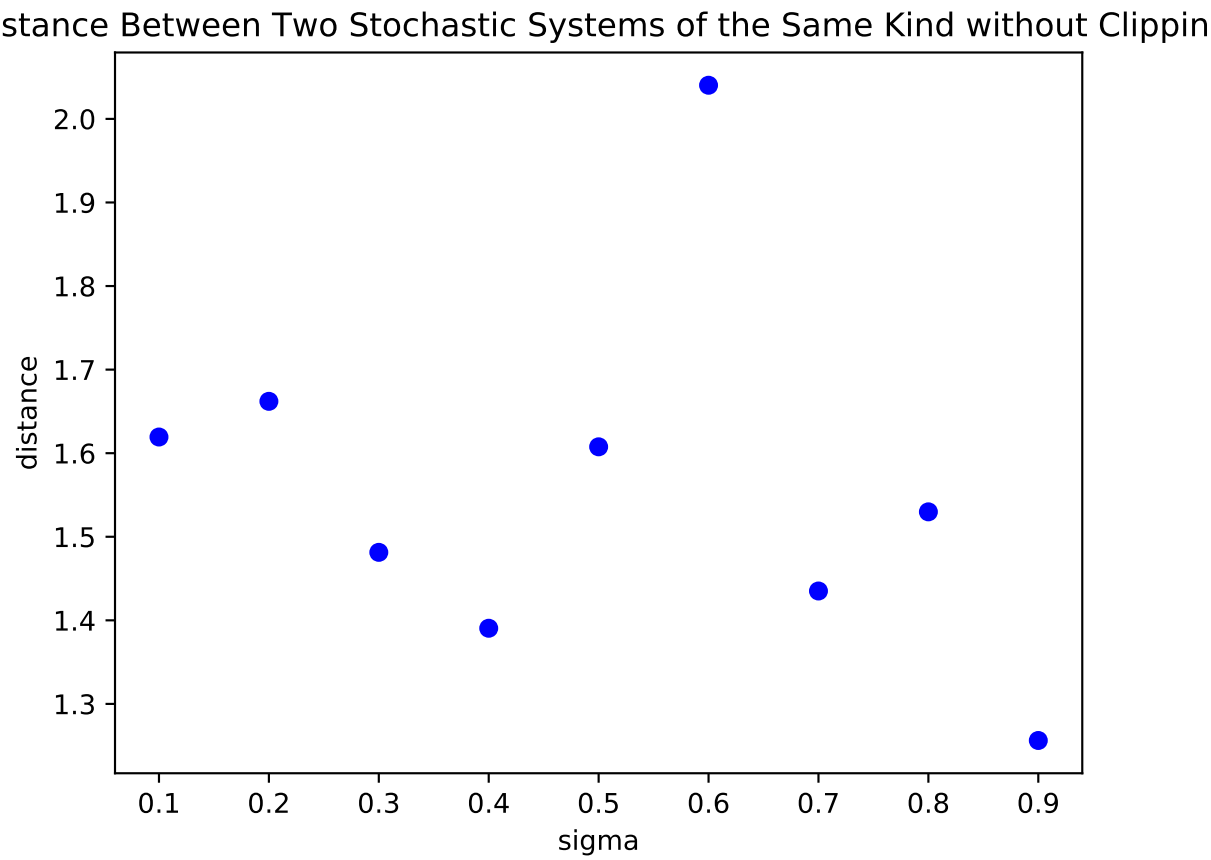

Figure A.20: Dynamical distance between system A and a similar system B as magnitude of stochastic noise increases, without clipping.

A.3.4 Comparing Stochastic Systems as Magnitude of Stochastic Noise Increases

A.3.5 Comparing Deterministic Systems with Different Levels of Observability, as Sampling Rate Increases

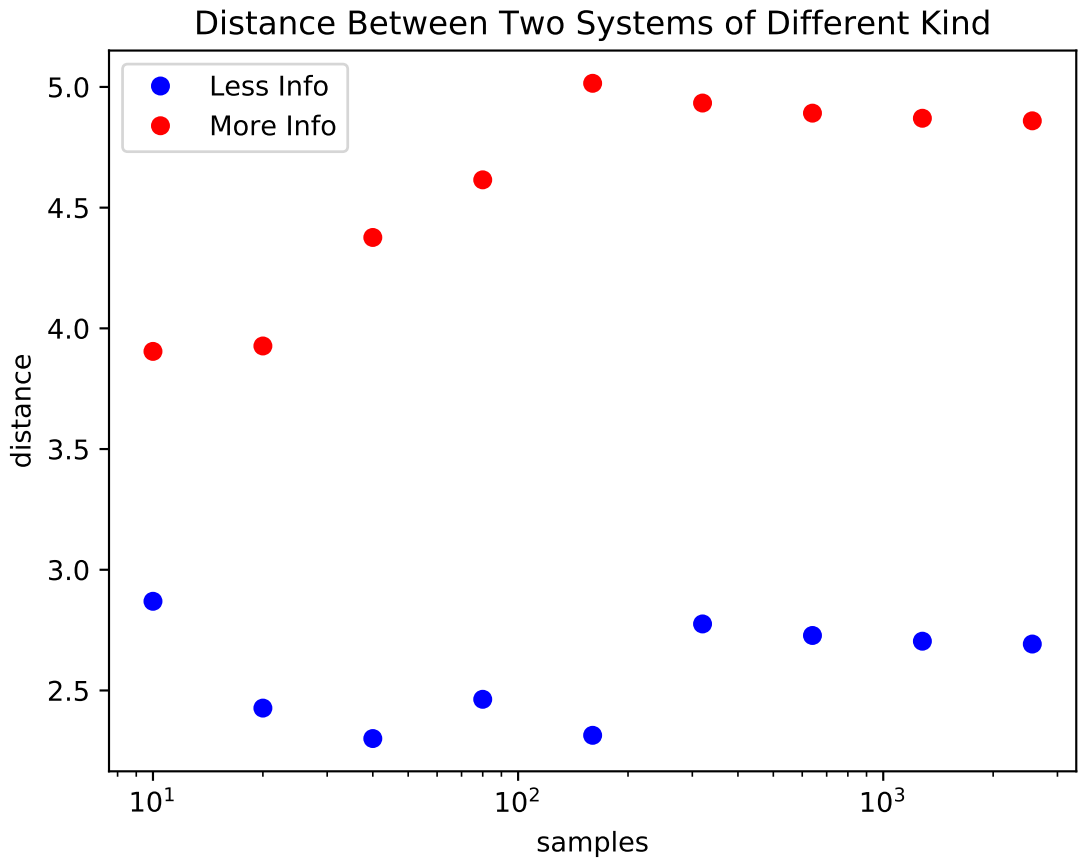

Figure A.21: Dynamical distance between system A and a different system B as sampling rate increases. Red denotes a system with a SSD variable set. Blue denotes a system with a non-SSD variable set.

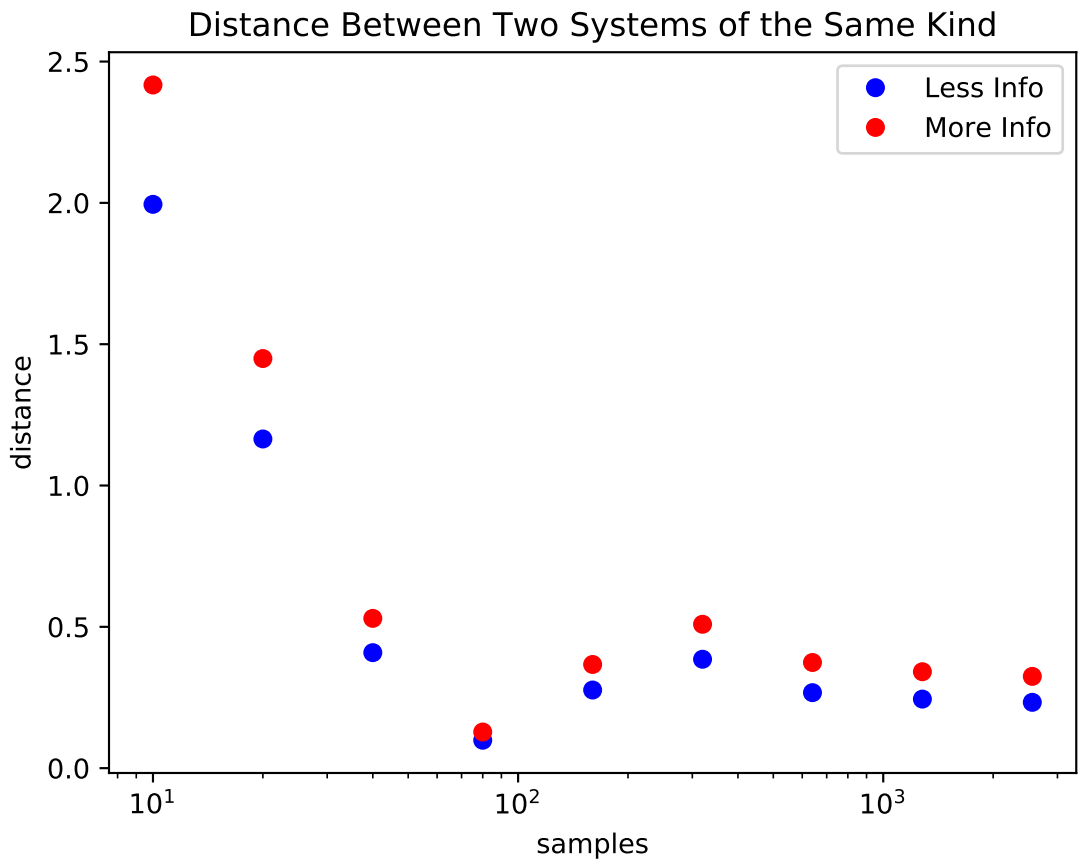

Figure A.22: Dynamical distance between system A and a similar system B as sampling rate increases. Red denotes a system with a SSD variable set. Blue denotes a system with a non-SSD variable set.

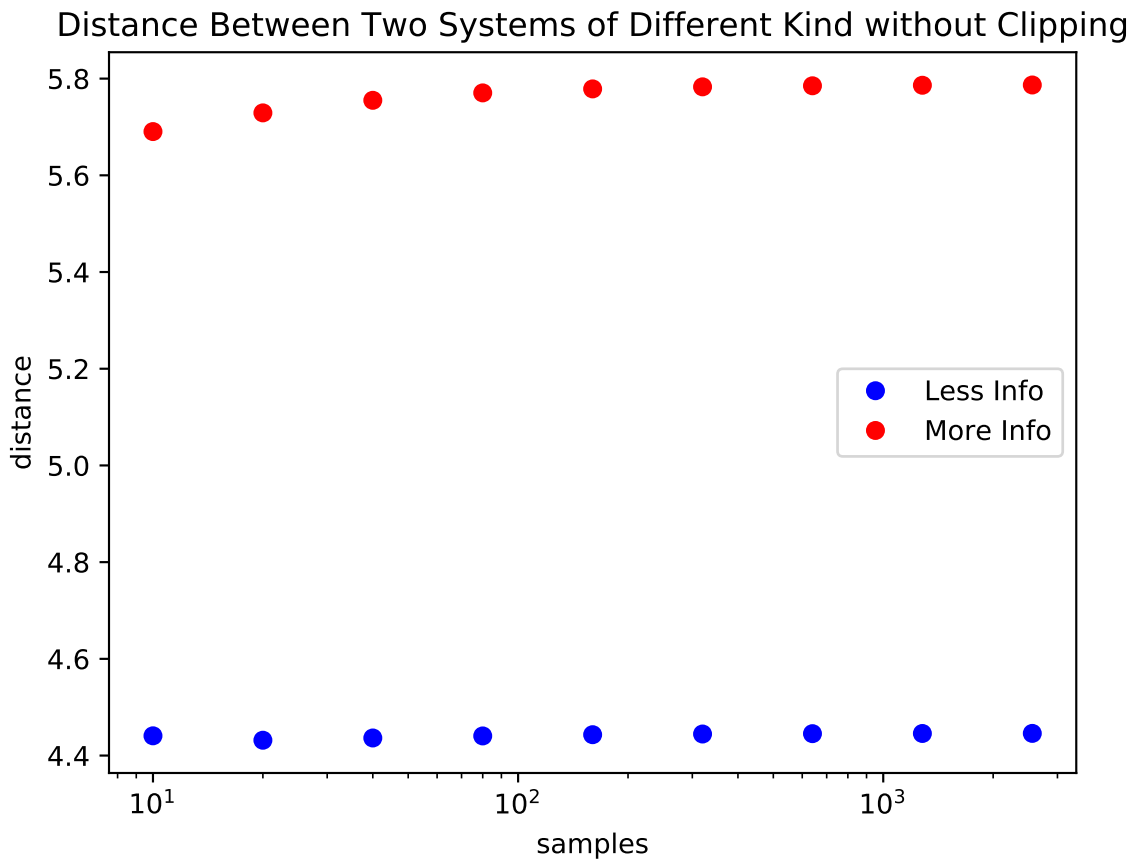

Figure A.23: Dynamical distance between system A and a different system B as sampling rate increases, without clipping. Red denotes a system with a SSD variable set. Blue denotes a system with a non-SSD variable set.

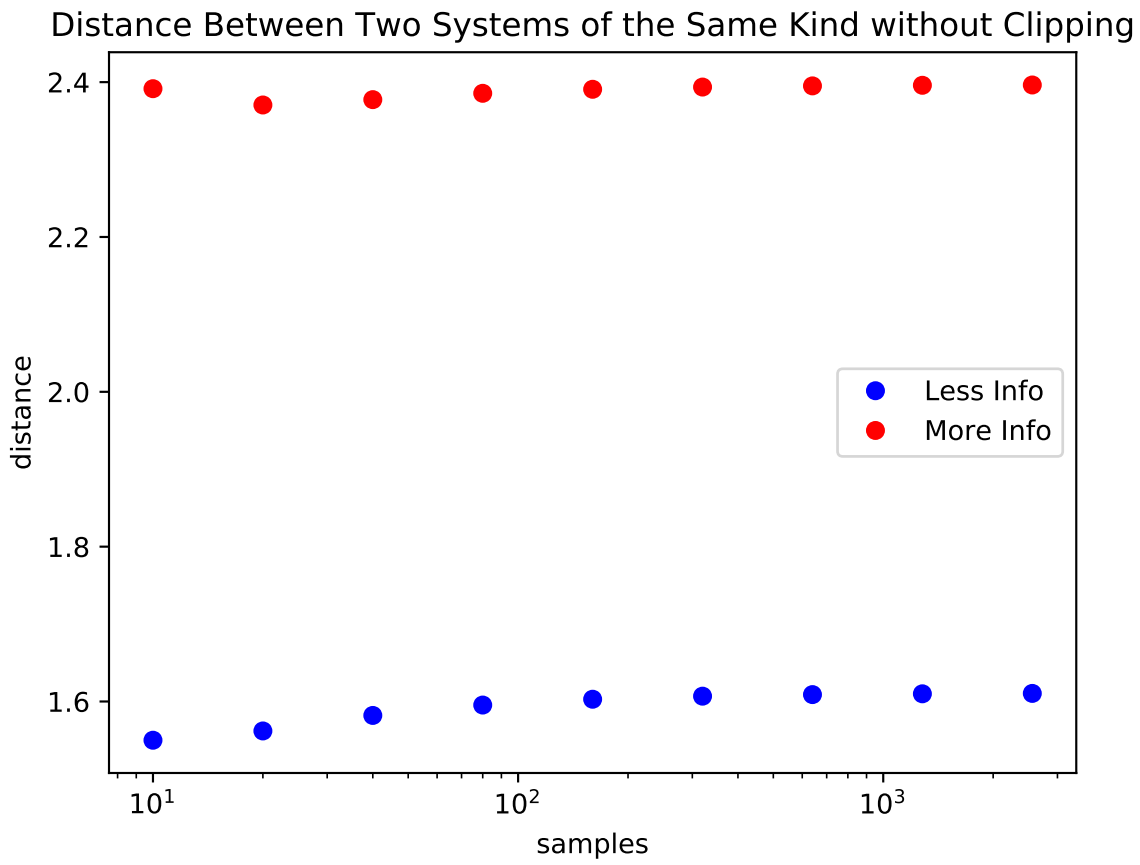

Figure A.24: Dynamical distance between system A and a similar system B as sampling rate increases, without clipping. Red denotes a system with a SSD variable set. Blue denotes a system with a non-SSD variable set.

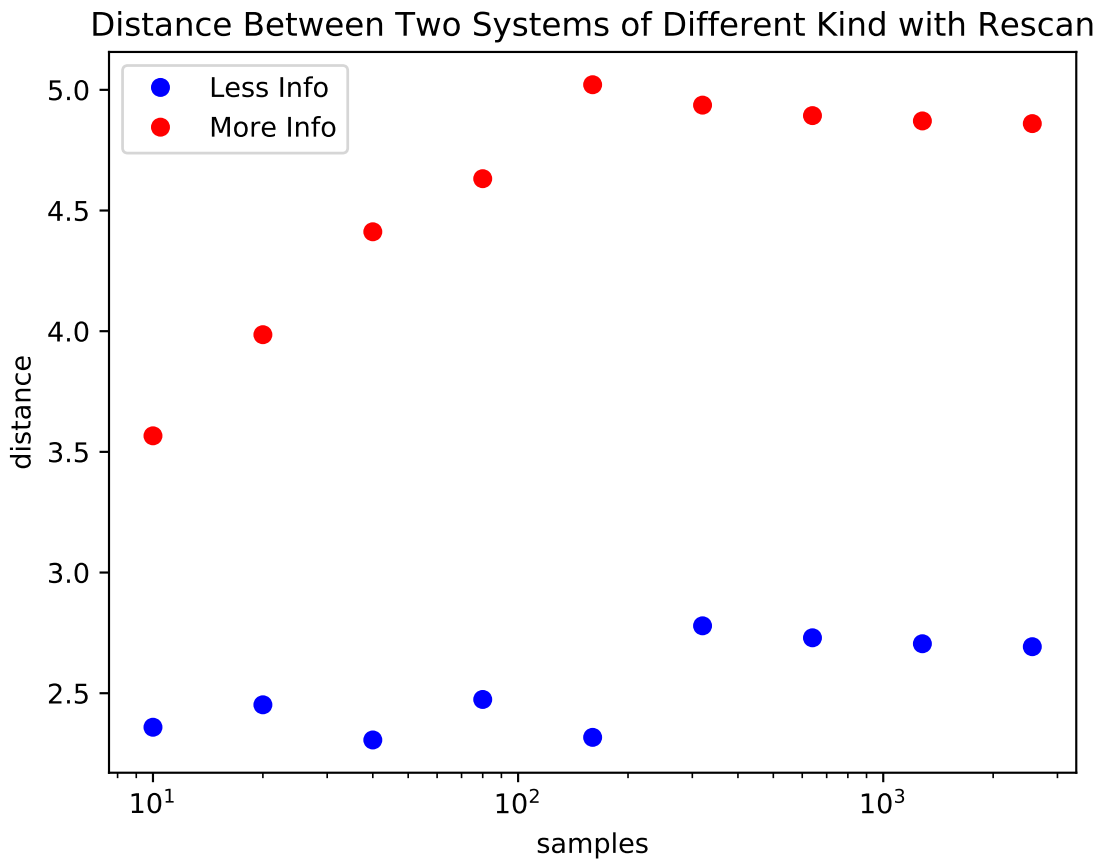

Figure A.25: Dynamical distance between system A and a different system B as sampling rate increases. Data is clipped, and system is rescanned to maximize data resolution. Red denotes a system with a SSD variable set. Blue denotes a system with a non-SSD variable set.

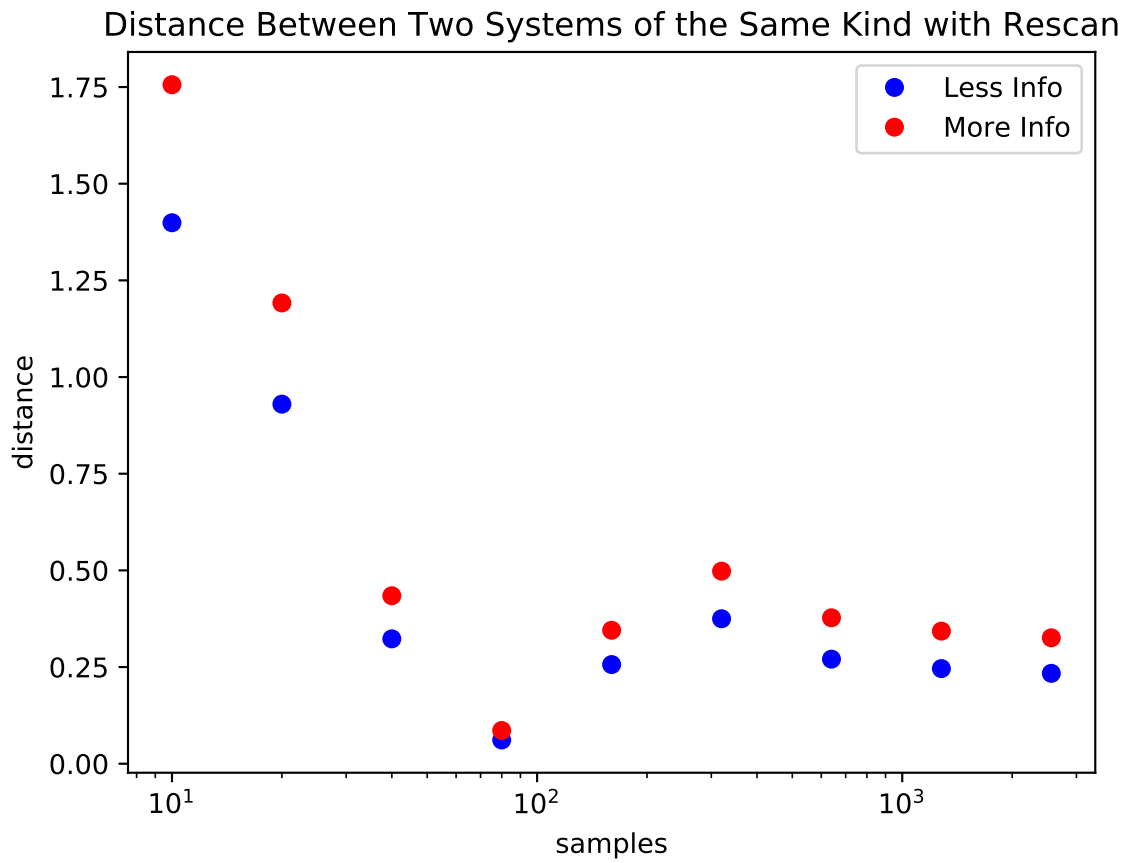

Figure A.26: Dynamical distance between system A and a similar system B as sampling rate increases. Data is clipped, and system is rescanned to maximize data resolution. Red denotes a system with a SSD variable set. Blue denotes a system with a non-SSD variable set.

A.4 Test Results Discussion

Tests in section A.3.1 show strong trends toward greater accuracy as the number of data samples increases. Removing clipping for systems without noise stabilizes the distance, but makes distances in noisy systems far less regular.

Sections A.3.2 and A.3.3 repeat these behaviors. As data resolution or replicates increase, similar systems are identified as being more similar, and different systems are identified as being more different. When clipping is removed, this trend is eliminated, though absolute distance does remain in a smaller span, and within a reasonable range.

In section A.3.4, we see that different systems seem more similar as the noise is increased, but the trend is not so clear for similar systems. The removal of clipping does not seem to affect that.

Finally, section A.3.5 shows that more information makes it easier to tell two different systems apart, while it does little to make similar systems appear more similar. Removing clipping again stabilizes the distances, but in the case of comparing two systems of the same dynamical kind, the introduction of more information decreases the performance of our algorithm below the performance of the algorithm given non-SSD variables. When clipping is used to then rerun the simulations to the clipped length, maintaining the original resolution of the data, we see a similar response as the case where we use clipping but do not rescan.

In almost all of these results the distance between two systems that are of similar kind is less than half the distance between two systems that are of different kind within the same battery of tests.