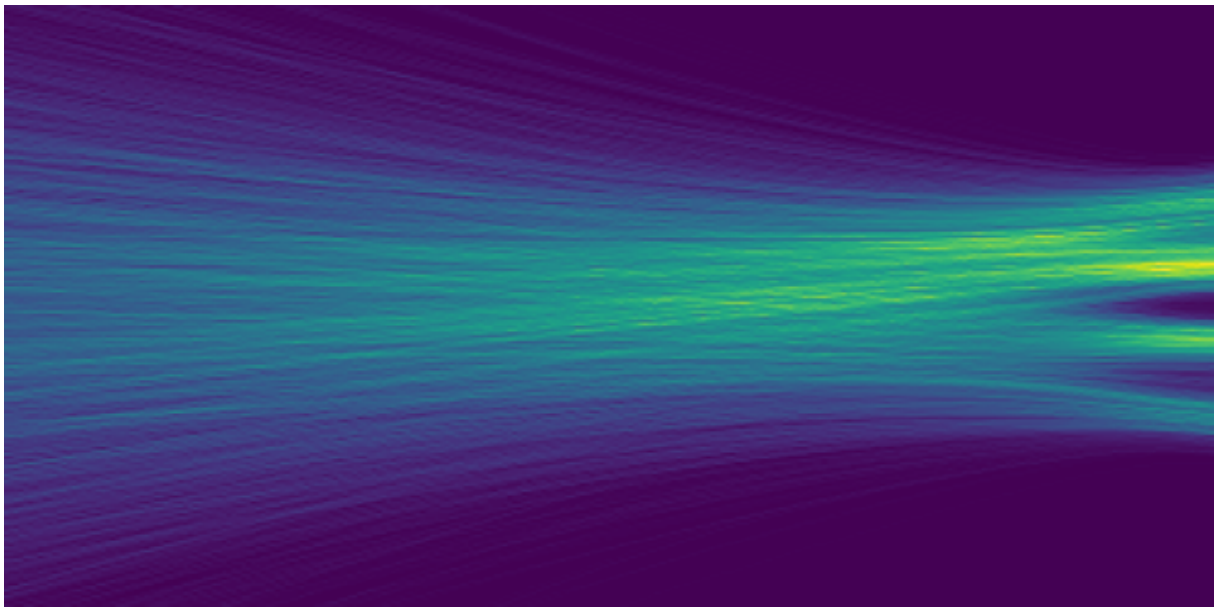
CHALMERS
UNIVERSITY OF TECHNOLOGY



Testing the Semigroup Property of Generative Models for Dynamical Systems

Developing a test based on the Chapman–Kolmogorov equation

Master's thesis in Computer science and engineering

Max Green & Hedvig Wennberg

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Master's thesis 2026

Testing the Semigroup Property of Generative Models for Dynamical Systems

Developing a test based on the Chapman–Kolmogorov equation

Max Green & Hedvig Wennberg

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden 2026

Testing the Semigroup Property of Generative Models for Dynamical Systems
Developing a test based on the Chapman–Kolmogorov equation
Max Green & Hedvig Wennberg

Supervisor: Simon Olsson, Department of Computer Science and Engineering
Examiner: Simon Olsson, Department of Computer Science and Engineering

Master's Thesis 2026
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A vector field transporting samples from a base distribution to a sampled distribution.

Typeset in LaTeX
Gothenburg, Sweden 2026

Testing the Semigroup Property of Generative Models for Dynamical Systems
Developing a test based on the Chapman–Kolmogorov equation
Max Green & Hedvig Wennberg
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Surrogate models for molecular dynamics, particularly those based on generative artificial intelligence, offer an efficient way to model molecular systems across timescales that may be difficult to access through simulation. However, such models should remain consistent with the underlying physics. For Markovian dynamics, the Chapman–Kolmogorov equation is a cornerstone of this consistency, describing how transition dynamics across different timescales should relate to each other. One such surrogate model, the Implicit Transfer Operator (ITO) framework, learns transition dynamics across multiple timescales, making it natural to question whether the learned dynamics remain consistent. Existing methods to assess this quantitatively use comparisons of distributions in the molecular space, while the test proposed in this work instead evaluates distributions in latent space, enabling metrics that were previously unavailable.

In this thesis, we develop and evaluate a Chapman–Kolmogorov test for ITO models operating in the latent space of the model. The test is evaluated on both a one-dimensional model trained on the dynamics from a potential well and a three-dimensional transferable model trained on molecular dynamics data. The one-dimensional model passes the test consistently, while the three-dimensional model gives more uncertain results, leading to a discussion about both the model and the multivariate version of the test. We further show that the CK-test's performance improves alongside the learning of correct dynamics during training, suggesting that the semigroup property is learned rather than being inherent to the model architecture. However, passing the test does not guarantee that the model has learned the correct dynamics, as models with poor dynamical accuracy can still satisfy the CK-test.

Keywords: Molecular Dynamics, Conditional Flow Matching, ITO, TITO, master thesis, semigroup property, Chapman–Kolmogorov equation

Acknowledgements

We would like to extend our gratitude to all the people who have helped and supported us over the course of this master thesis project.

Throughout this process, Simon has been both dependable and supportive, always providing insightful thoughts and ideas in regards to our project. The weekly meetings always left us feeling inspired and excited for the upcoming week, and if nothing was making sense he always had some insightful knowledge to impart on us. The recurring question "Are you having fun with this project?" was also greatly appreciated. He is the best supervisor we could have asked for and we really appreciate all the time and energy he has given us.

We would also like to thank all the members of our research group for always being helpful when we raised questions and problems during group meetings. It has truly been an environment for learning and all the weekly presentations have kept us curious. We would like to especially thank Selma, Flemming and Juan for their extra support and help in regards to our project.

Additionally, we wish to thank our friends for helping keep the work-life balance reasonable and making life at Chalmers not only educational, but also fun! We are also grateful for our families and their unconditional support.

Lastly we would like to thank Max's cat Miso for her constant encouragements and valuable feedback during this project.

Max Green & Hedvig Wennberg, Gothenburg, 2026-07-01

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

AI	Artificial Intelligence
CFM	Conditional Flow Matching
CK	Chapman–Kolmogorov
CNF	Continuous Normalizing Flow
FM	Flow Matching
GenAI	Generative Artificial Intelligence
ITO	Implicit Transfer Operator
MD	Molecular Dynamics
MSM	Markov State Model
ODE	Ordinary Differential Equation
RMS	Root Mean Square
std	Standard deviation
TICA	Time-Lagged Independent Component Analysis
TITO	Transferable Implicit Transfer Operator
UVF	Ultra Vector Field

Symbols

α	Significance level
τ	Lag time
θ	Trainable model parameters
$\tilde{z}$	Reconstructed noise sample
N_{ODE}	Number of ODE integration steps
u_t	Target velocity field
v_θ	Learned neural vector field
z_0	Sample from the base distribution

1

Introduction

1.1 Introduction

One of the fundamental branches of physics is classical mechanics, the theory built upon Newton's laws of motion. An important implication of classical mechanics is that if the position and velocity of all objects in a system are known, then it is possible to determine both how the system will evolve in the future and how it evolved to reach its current state. Although these dynamics can in principle be applied to systems of any size, a direct microscopic description becomes impractical for macroscopic systems because of the enormous number of particles involved. Furthermore, macroscopic thermodynamic behavior is often irreversible in time, despite the underlying microscopic equations typically being time-reversible [1]. Statistical mechanics was developed to bridge this gap.

Statistical mechanics is derived from the realization that macroscopic properties do not have a strong dependency on microscopic details, which allows modeling based on statistical probabilities instead of individual microscopic objects. This is achieved with the use of *ensembles*, a collection of systems that share the same macroscopic properties and obey the same microscopic laws of motion, but with differing initial conditions. As a result, each system in the ensemble may occupy a different microscopic state at a given time. Consequently, the thermodynamical properties of a system, as well as other equilibrium and dynamical properties, can be derived from the averages of the systems in an ensemble [1]. This ability to explain macroscopic properties in terms of microscopic parameters has proven useful in a variety of fields, particularly molecular science.

Since molecular science studies systems at a microscopic scale, it follows that statistical mechanics is naturally applicable. However, the rules of statistical mechanics cannot circumvent the complexities of a system, so realistic molecular systems remain difficult to solve. This is primarily due to the sheer number of interacting particles, leading to analytical solutions only being possible for highly simplified systems [1]. Therefore, rather than attempting to solve the equations exactly, different methods have been developed to approximate the dynamics of a system by sampling smaller representative systems. This is possible because many macroscopic properties converge quickly with respect to system size; thus, a much smaller system can still act as an adequate representation [1].

1.2 Background

Molecular systems of interest may contain particles on the order of 10^{23}, making a direct simulation of the system infeasible. *Molecular dynamics* (MD) addresses this by constructing a smaller representative system, typically containing between 10^2 and 10^9 particles, and evolving it numerically according to the equations of motion [1]. This makes MD a central tool for studying problems in statistical mechanics.

By solving the stochastic differential equations, MD models the evolution of the system over time by generating trajectories for the system's particles, enabling insight into the molecular behavior, properties and structural changes [1], [2]. However, in order to ensure accuracy and stability of this process, it demands tiny time steps, often on the order of femtoseconds (10^{-15}), which makes phenomena that occur on longer time scales infeasible to simulate due to the computational cost. This limitation is problematic because relevant biological processes can occur on the order of seconds [2], [3]. Considering that these processes are still relevant to study, approximation methods based on MD were developed with the goal of extending the time horizon while keeping down the computational costs.

Several such methods are based on the assumption that the dynamics can be approximated as Markovian [1]. In other words, the system is "memoryless", which means that any future evolution of the system depends only on its current state and not on any information about how it reached this state [4]. An important consequence of Markovianity is that the dynamics satisfy the *Chapman–Kolmogorov equation* (CK equation) and the associated transition operators satisfy the *semigroup* property. This states that evolving a system for a time $k\tau$ should have the same distribution as evolving it k consecutive steps of time τ [5]. Therefore, this property provides a natural criterion for evaluating whether a learned model manages to remain consistent with the underlying physics.

One such method that extends the time horizon under the assumption of Markovian dynamics is the *Implicit Transfer Operator* (ITO), which provides a framework for learning the effects of the transfer operator across multiple time lags using generative AI [6] and its extensions [7], [8], [9]. This operator describes how a probability distribution over system states evolves over time. By learning a surrogate model that approximates the transfer operator, the ITO framework is able to reproduce the corresponding distributional dynamics. Furthermore, the ITO framework enables efficient simulation of long time-scale dynamics compared to MD, significantly reducing computational cost [6]. However, since it uses generative AI, the lack of transparency into the model's decision making raises questions about the reliability of this approach and whether there is a way to verify that the model maintains the semigroup property.

The current approach to verify whether an ITO model satisfies the semigroup property is to compare the distribution obtained from traversing k steps of length τ with the distribution obtained from traversing one step of length $k\tau$. While this is a valid approach, it currently relies on visual inspection of the resulting distributions in molecular space, which raises the possibility of errors due to bias. Developing a quantitative and reproducible method for evaluating whether the model satisfies the semigroup property could increase confidence in the framework.

1.3 Goals and Challenges

The overall goal of this project is to develop and evaluate a test for the semigroup property that can be applied to ITO models. In order to achieve this, the project aims to answer the following research questions:

1. Is it possible to implement a Markovianity test on ITO models based on the Chapman–Kolmogorov equation in latent space?
2. Is the Chapman–Kolmogorov test applicable across different scales and problems?
3. Do ITO models inherently follow the semigroup property or is it something that is learned during training?

To answer these questions, the project focuses on developing a quantitative CK-test which can evaluate whether an ITO model follows the semigroup property by being self-consistent under compositions of time lags. Instead of relying on visual inspection of sampled distributions, the test should give numerical metrics that can be compared across different models, time lags, and experimental settings. The test is also based on a combination of different normality tests and distributional measures to increase the reliability of the test.

Furthermore, the test is applied and evaluated on both one-dimensional and three-dimensional models. The one-dimensional model is used as a controlled setting where the test can be developed and interpreted more easily, while the three-dimensional model is used to investigate if the same approach can be applied to more advanced molecular systems. The test is also applied to both trained and untrained models, as well as to models at different stages of training. This is done to investigate whether passing the CK-test is actually connected to learning the correct dynamics, or whether the semigroup property is inherently followed by ITO models.

2

Theory

2.1 The Markov Property

A fundamental concept in stochastic processes is the *Markov property*, which describes memoryless dynamics. Intuitively, this means that if the current state of a system is known, then no additional information about how the system reached that state is needed to describe its future evolution. In other words, the present state contains all relevant information from the past. A process that satisfies this property is referred to as *Markovian* [4].

More formally, consider any times $t_1 < t_2 < \cdots < t_n < t < s$. The Markov property states that, given the current state X_t, knowing the previous states $X_{t_1}, X_{t_2}, \cdots, X_{t_n}$ provides no more information about the future than just knowing X_t [5]. This can be written as:

$$\mathbb{P}(X_s \in B \mid X_{t_1}, X_{t_2}, \ldots, X_{t_n}, X_t) = \mathbb{P}(X_s \in B \mid X_t) \tag{2.1}$$

where B denotes a set of possible future states. This equation states that once the current state X_t is known, conditioning on additional past states does not change the probability assigned to future states.

2.1.1 The Chapman–Kolmogorov Equation

Associated with a Markov process is a family of transition functions $P_{t,s}(x, A)$, which describe the probability of transitioning from state x at time t to a measurable set A at time $s > t$. These transition functions satisfy the consistency conditions of probability measures and measurability, and obey the *Chapman–Kolmogorov equation* (CK equation)[5]:

$$P_{t,u}(x, A) = \int P_{t,s}(x, dy)\, P_{s,u}(y, A), \quad t < s < u \tag{2.2}$$

This equation expresses that a transition over a longer time interval can be decomposed into successive transitions over intermediate time intervals. In the time-homogeneous case, where $P_{t,s}(x, A) = P_{0,s-t}(x, A)$, this reduces to

$$P_{0,t+s}(x, A) = \int P_{0,t}(x, dy)\, P_{0,s}(y, A) \tag{2.3}$$

which highlights the compositional structure of Markov dynamics [5].

2.1.2 Semigroup Interpretation

The CK equation admits a natural operator-theoretic interpretation. Instead of describing the dynamics directly through transition probabilities, one can describe how functions of the system state evolve over time. Let $f : E \to \mathbb{R}$ be a function that assigns a value to each state of the system. The operator T_t maps this function to a new function that gives the expected value of f after the system has evolved for time t, starting from the state x. This can be written as:

$$(T_t f)(x) = \int f(y) P_t(x, dy) \tag{2.4}$$

where $P_t(x, dy)$ is the transition probability from the initial state x to a future state y after time t. Thus, $T_t f(x)$ represents the average value of f over all possible states y that the process can reach after time t.

The CK equation states that evolving the system first for time t and then for time s is equivalent to evolving it once for the combined time $t + s$. In operator form, this becomes

$$T_{t+s} = T_t T_s \tag{2.5}$$

A family of operators satisfying this composition rule is called a *semigroup*. Therefore, the semigroup property of the operators $\{T_t\}_{t \geq 0}$ is the operator-theoretic version of the CK equation [5].

2.2 Molecular dynamics

Molecular dynamics (MD) is a computational simulation method used to analyze the physical movements of molecules and atoms at the microscopic scale. This is mainly done by simulating the time evolution of the particles and then solving Newton's equations of motion for the particles [1]. With this approach, each particle is assigned an initial position and velocity, based on some initial conditions, and the forces acting on the particles are computed from a defined potential energy function, also known as a force field. The acceleration of each particle can then be determined using Newton's second law of motion, $F = ma$, and each particle's position and velocity can be updated over very small time steps. Repeating this procedure many times will generate a trajectory that describes how the system configuration evolves over time.

2.2.1 Langevin dynamics

A widely used formulation of MD, particularly when modeling the interactions of a system with a thermal environment, is given by Langevin dynamics [10]. In this framework, the deterministic evolution governed by Newton's equation is augmented with stochastic and frictional forces that mimic the effect of a heat bath.

Let $q(t) \in \mathbb{R}^{3N}$ denote the positions and $v(t) \in \mathbb{R}^{3N}$ the velocities of N particles. The Langevin dynamics can be written in stochastic differential form as [1]:

$$dq(t) = v(t)\,dt \tag{2.6}$$

$$dv(t) = f(q(t))\,dt - \gamma v(t)\,dt + \sigma\,dw(t) \tag{2.7}$$

where $f(q) = F(q)/\mu$ denotes the force per unit mass, γ is the friction coefficient, μ is the particle mass, and $w(t)$ is a Wiener process. The noise amplitude is given by

$$\sigma = \sqrt{\frac{2k_B T\gamma}{\mu}} \tag{2.8}$$

2.2.2 Propagators and Transfer Operators

An alternative perspective on MD is obtained by considering the evolution of probability densities rather than individual trajectories. Let $p(\mathbf{x}, t)$ denote the probability density of the system in configuration space at time t. Under the assumption of Markovian dynamics, the time evolution of p is fully characterized by the transition density $p(\mathbf{x}_\tau \mid \mathbf{x}_0)$ over a lag time τ. The corresponding *Perron–Frobenius operator* $\mathcal{P}^\tau$ (also called the propagator) acts on an initial density p_0 to produce the propagated density [11]:

$$p(\mathbf{x}, \tau) = (\mathcal{P}^\tau p_0)(\mathbf{x}) = \int p(\mathbf{x} \mid \mathbf{x}_0, \tau)\, p_0(\mathbf{x}_0)\, d\mathbf{x}_0. \tag{2.9}$$

In molecular dynamics, this density propagator is closely related to the transfer operator. The distinction between the two is mainly a matter of normalization and choice of function space [12]. Thus, both operators encode the same underlying Markovian dynamics, but act on differently normalized representations of the density. In this thesis, the term transfer operator is used in this broader sense.

The spectral decomposition of these operators provides information about the slow dynamical processes of the system. Formally, this can be written as [11]:

$$\mathcal{P}^\tau \psi_i = \lambda_i \psi_i, \tag{2.10}$$

where the eigenvalues λ_i encode the characteristic relaxation timescales of the system.

The transfer operator provides a mathematical description of how probability distributions evolve over time. The challenge addressed in this thesis is how to approximate these operators efficiently for complex molecular systems where direct simulation is computationally expensive. One approach is to learn surrogate models of the dynamics using generative artificial intelligence.

2.3 Generative Artificial Intelligence and flow-based models

Generative artificial intelligence (GenAI) refers to a type of AI that is able to generate data, instead of just analyzing or classifying it [13]. This is usually done by allowing GenAI models to learn patterns and structures of the data, often called features, and then using those features to generate new data [13]. There are several different ways that GenAI can learn features and there are also several ways it can use it to generate new data [14].

Flow-based generative models refer to generative models that leverages *normalizing flow* to transform a simple probability distribution into a more complex data distribution [15]. This section will explain both the concepts behind flow-based models and the relevant models used during this project.

2.3.1 Normalizing flow

A *normalizing flow* is a transformation from a simple distribution, e.g. Gaussian distribution, into a different, often more complex, distribution [15], [16]. The transformation works by using a sequence of invertible and differentiable mappings, where each mapping incrementally warps the probability density while preserving bijectivity. This allows the resulting distribution to be evaluated by transforming a sample back to the base distribution and applying the change-of-variables formula [15].

Formally, let $x_0 \in \mathbb{R}^d$ be a random vector that follows a known simple base density, e.g. a normal distribution:

$$x_0 \sim p(x_0) = \mathcal{N}(0, I)$$

We define a transformation $T : \mathbb{R}^d \to \mathbb{R}^d$:

$$T(x_0) = x_1 \tag{2.11}$$

Assume that T is a *diffeomorphism*, i.e. T is bijective, invertible and both T and T^{-1} are differentiable, then the density of x_1 can be obtained by the change-of-variable formula:

$$q(x_1) = p(x_0)|\det J_T(x_0)|^{-1}, \qquad \text{where } x_0 = T^{-1}(x_1) \tag{2.12}$$

Since $x_0 = T^{-1}(x_1)$, equation 2.12 can be expressed directly in terms of x_1:

$$q(x_1) = p(T^{-1}(x_1))|\det J_{T^{-1}}(x_1)| \tag{2.13}$$

One useful property that diffeomorphism transformations have is that they are *composable*, meaning that if you have two diffeomorphisms, T_1 and T_2, then their composition is also a diffeomorphism. This means that complex transformations can be built using compositions of simpler transformations while still maintaining the ability to calculate the probability density $q(x_1)$ [16]. Combined with the free choice of base density, this property implies that there are infinitely many ways to construct a normalizing flow.

When normalizing flows are used in models to fit a base distribution to a target distribution one of the most common ways is to minimize the Kullback-Leibler (KL) divergence [16], which is the same as maximizing the log likelihood equation. This equation can be expressed as the following based on equation 2.13:

$$\log q(x_1) = \log p(T^{-1}(x_1)) + \log |\det J_{T^{-1}}(x_1)| \tag{2.14}$$

2.3.2 Continuous normalizing flow

A normalizing flow can be expressed using the change-of-variable formula, resulting in the log likelihood as seen in Equation (2.14). When models try to maximize this equation, the main bottleneck in the calculations is to determine the value of the determinant of the Jacobian [17]. However, this computation can be simplified by using a continuous transformation, resulting in a model called *Continuous Normalizing Flow* (CNF).

Similarly to normalizing flow, a CNF model is used to transform a simple prior density into a more complicated one by modeling a vector field v_t using a neural network [18]. Moreover, in contrast to models based on normalizing flow, the CNF models can perform a reverse transformation that costs approximately the same as a forward pass [17]. It is also possible to construct CNF models with an increased amount of hidden units with only a linear cost increase, which allows for 'wider' flow layers than models using normalizing flow [17].

The CNF model is able to achieve this by defining a continuous-in-time transformation through an ordinary differential equation. Using the notation from earlier, let x_0 be a continuous random variable that follows a known simple base density, such as a normal distribution:

$$x_0 \sim p(x_0) = \mathcal{N}(0, I)$$

The ordinary differential equation used for the transformation of x_0 can then be described as follows:

$$\frac{dx_0}{dt} = f(x_0, t) \tag{2.15}$$

Contrary to normalizing flow, the transformation f does not need to be bijective, since if uniqueness is satisfied, the entire transformation will automatically satisfy bijectivity [17]. Now, assuming $f(x_0, t)$ is Lipschitz continuous in x_0 and continuous in t, the change in log probability also follows a differential equation:

$$\frac{\partial \log p(x_0)}{\partial t} = -tr\left(\frac{df}{dx_0}\right) \tag{2.16}$$

Since f is parameterized as a neural network the Jacobian trace in equation (2.16) becomes computationally tractable, instead of requiring the determinant in equation (2.14), which reduces the computational costs [17]. The trace operation is also linear, meaning the log density is a sum, which is why CNF models can have many hidden units with only a linear cost increase [17].

The resulting flow that is modeled by the CNF is determined as a function of t, since the differential equation $f(x_0, t)$ is dependent on t. This is done by introducing a gating mechanism $\sigma_n(t)$, which is a neural network that learns when the transformation $f_n(x_0)$ should be applied [17]:

$$\frac{dx_0}{dt} = \sum_n \sigma_n(t) f_n(x_0)$$

2.3.3 Conditional Flow Matching

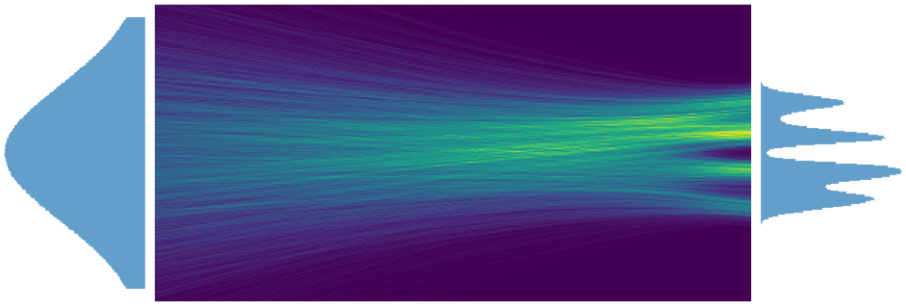


Figure 2.1: A visualization of how samples can be transformed from one distribution to another using CFM.

While CNF models are a general improvement from models based on normalizing flows, they still rely on maximum likelihood training. This requires repeatedly solving ODEs for every optimization step, both for the forward propagation and the backward propagation, which is not only computationally expensive, but also difficult to scale for high-dimensional data [18]. *Flow matching* (FM) addresses these limitations by replacing the likelihood-based training with a simulation-free regression objective defined over a probability path connecting a simple base distribution to the target data distribution. Instead of simulating trajectories during training, FM directly learns the vector field that transports samples along this path.

Let $p_0(z)$ denote a simple base distribution and $q(x)$ the target data distribution. FM introduces a family of intermediate probability distributions $(p_t)_{t\in[0,1]}$ that form a probability path between p_0 and q. In practice, this marginal probability path is constructed from a family of conditional probability paths $p_t(x_t \mid x_1)$, where $x_1 \sim q(x)$, such that

$$p_t(x_t) = \int p_t(x_t \mid x_1) q(x_1)\, dx_1. \tag{2.17}$$

Here, x_t denotes a point along the probability path at interpolation time t, while x_1 denotes a sample from the target data distribution. The flow matching objective is then given by [18]:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\, \mathbb{E}_{x_t\sim p_t(x_t)} \left[\left\| v_\theta(x_t, t) - u_t(x_t) \right\|^2 \right]. \tag{2.18}$$

Where $\mathcal{U}(0,1)$ refers to the uniform distribution, $u_t(x_t)$ denotes the target (marginal) velocity field associated with the probability path $(p_t)_{t\in[0,1]}$, and $v_\theta(x_t, t)$ is a neural

network parameterization of this vector field. A visualization of how such a vector field can transform the data between the two distributions can be seen in Figure 2.1.

The objective of FM is, in other words, to learn a neural network $v_\theta(x_t, t)$ that approximates the marginal velocity field. However, the marginal velocity field $u_t(x_t)$ is generally intractable, as it depends on the unknown data distribution $q(x)$ [18]. To address this, *Conditional Flow Matching* (CFM) instead considers the conditional probability paths $p_t(x_t \mid x_1)$, for which the corresponding conditional velocity fields $u_t(x_t \mid x_1)$ can be computed in closed form.

Using the marginalization in equation (2.17) the FM objective can be expressed as an expectation over the joint distribution of (x_t, x_1), yielding the CFM objective:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\, \mathbb{E}_{x_1\sim q(x)}\, \mathbb{E}_{x_t\sim p_t(x_t|x_1)} \left[\left\| v_\theta(x_t, t) - u_t(x_t \mid x_1) \right\|^2 \right] \tag{2.19}$$

This reformulation avoids the need to compute the marginal velocity field $u_t(x_t)$ directly, replacing it with the conditional velocity field $u_t(x_t \mid x_1)$, which is tractable for suitable choices of the conditional path. Importantly, the CFM objective is equivalent to the FM objective in expectation, ensuring that minimizing $\mathcal{L}_{\text{CFM}}$ recovers the correct marginal dynamics [18]. CFM is part of a broader family of continuous-time generative modeling approaches that were developed around the same time, including optimal-transport conditional flow matching [19], stochastic interpolants [20], and rectified flow [21].

2.3.4 Implicit Transfer Operator

The *Implicit Transfer Operator* (ITO) is a framework for learning stochastic transition dynamics of complex systems, across multiple time scales, using generative models [6]. Rather than simulating the dynamics of a system through small integration time steps, like MD simulations, ITO seeks to approximate the time-lagged transfer operator of the system. It does this by learning a surrogate model that approximates the conditional transition distribution. In particular, if the surrogate model is trained on simulated MD data, the transition probability can be expressed as [6]:

$$p(\mathbf{x}_{N\tau} \mid \mathbf{x}_0) = \sum_{i=1}^{\infty} \underbrace{\lambda_i^N(\tau)}_{\text{time-variant}} \underbrace{\alpha_i(\mathbf{x}_{N\tau})\, \beta_i(\mathbf{x}_0)}_{\text{time-invariant}} \tag{2.20}$$

where τ is the lag time, N is the number of lag steps, so that $N\tau$ is the total physical lag time, $\lambda_i(\tau)$ are the eigenvalues of the transfer operator, and α_i, β_i are the corresponding left and right eigenfunctions. The eigenfunctions describe the state-dependent modes of the dynamics, while the eigenvalues determine how much each mode contributes after a given lag time. Modes with eigenvalues close to one decay slowly and therefore correspond to long-lived dynamical processes, while modes with smaller eigenvalues decay more rapidly.

This decomposition separates the dynamics into a time-variant component, given by $\lambda_i^N(\tau)$, and a time-invariant component, given by $\alpha_i(\mathbf{x}_{N\tau})\beta_i(\mathbf{x}_0)$. By training on

transitions at different lag times, the ITO framework exposes the model to different combinations of the same underlying dynamical modes. This is intended to improve generalization across time scales and to make long-time sampling more stable [6].

2.4 Normality tests

Since the CK-test developed in this thesis evaluates whether reverse sampling recovers the latent normal distribution, the following section introduces the normality tests used to assess its success quantitatively.

Numerous methods have been developed to assess whether a dataset follows a normal distribution. A common approach is to visualize the distribution using, for example, histograms or QQ-plots and comparing it with the known shape of a normal distribution [22]. This approach, however, only works in low dimensions and is an inherently subjective technique since it relies on manual inspection of the generated figures. Another approach is to use analytical methods that evaluate the distribution numerically, enabling a quantitative and reproducible assessment of normality. The following sections present the normality tests used in this thesis.

2.4.1 P-value

When evaluating the test statistic of a normality test, the p-value is used to quantify the evidence against the null hypothesis. In the context of normality testing, the null hypothesis typically states that the data is drawn from a normal distribution. The p-value is defined as the probability of observing a test statistic at least as extreme as the one obtained from the sample data, assuming that the null hypothesis is true [23]. Formally, this can be written as:

$$p = P(T(X) \geq T(x_{\text{obs}})|H_0) \tag{2.21}$$

where $T(X)$ is the test statistic computed from the data, $T(x_{\text{obs}})$ is the observed value of the test statistic, and H_0 denotes the null hypothesis.

A small p-value indicates that such an observation would be unlikely under the assumption of normality, providing evidence against the null hypothesis. Conversely, a large p-value suggests that the observed data is consistent with a normal distribution. When used in practice, the p-value is compared to a predefined significance level α and if the p-value is less than α, the null hypothesis is rejected, indicating that the data does not follow a normal distribution. This also implies that if a normality test is applied to a perfect normal distribution, for a given α, it is expected that the test will falsely reject normality at about a rate of α.

2.4.2 Energy distance

Energy distance is a metric that can be used to compare the difference between two distributions. It is based on the expected Euclidean distance between independent samples from the two distributions [24], [25]. Let X and Y be independent random vectors with distributions Z_0 and $\tilde{Z}$ respectively, and let X' and Y' denote independent copies of X and Y. The squared energy distance is defined as:

$$D^2(Z_0, \tilde{Z}) = 2\mathbb{E}\|X - Y\| - \mathbb{E}\|X - X'\| - \mathbb{E}\|Y - Y'\| \tag{2.22}$$

The energy distance is then given by $D(Z_0, \tilde{Z})$. The value of the energy distance satisfies $D(Z_0, \tilde{Z}) \geq 0$ and is only exactly 0 if the distributions are identical $Z_0 = \tilde{Z}$ [24].

For two empirical samples $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$, the corresponding two-sample energy statistic is [24]:

$$E_{n,m}(X, Y) = 2A - B - C \tag{2.23}$$

where

$$A = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\|x_i - y_j\| \qquad B = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|x_i - x_j\| \qquad C = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\|y_i - y_j\| \tag{2.24}$$

Thus, the statistic compares the average distance between samples from the different distributions with the average distances within each sample. A value close to zero indicates that the two empirical distributions are similar, while larger values indicate a larger distributional discrepancy. Importantly, this can be used to compare two distributions of the same, but arbitrary dimensions.

2.4.3 Univariate tests

For univariate data, several tests can be used to detect different deviations from normality. This thesis uses the Shapiro–Wilk test and D'Agostino's K^2 test, which assess normality through different properties of the sample distribution.

2.4.3.1 The Shapiro–Wilk test for normality

The Shapiro-Wilk test evaluates normality by using a statistic based on the ordered sample values and the expected order statistics of a normal sample. Power comparisons have reported that the Shapiro–Wilk test has the highest power relative to commonly used alternatives [26], [27].

Given an ordered random sample $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$, where n is the number of data points, the Shapiro–Wilk test statistic is defined as:

$$W = \frac{(\sum a_i x_{(i)})^2}{\sum (x_i - \bar{x})^2} \tag{2.25}$$

where $x_{(i)}$ is the i^{th} order statistic, $\bar{x}$ is the sample mean,

$$\mathbf{a} = (a_1, ..., a_n) = \frac{m^T V^{-1}}{(m^T V^{-1} V^{-1} m)^{1/2}}$$

and $\mathbf{m} = (m_1, ..., m_n)^T$ are the expected values of the order statistics of independent and identically distributed random variables sampled from the standard normal distribution and $\mathbf{V}$ is the covariance matrix of those normal order statistics [28].

The test statistic W can take values between zero and one. For small values of W normality is rejected for the data distribution and values close to one indicate normality. The original Shapiro-Wilk test was designed for data of size at most $n = 50$, but the test has been improved and can be used for data distributions consisting of $3 \leq n \leq 5000$ points [27].

2.4.3.2 D'Agostino's K-squared normality test

Tests based on sample moments, such as skewness and kurtosis, are commonly referred to as moment based tests for normality. D'Agostino's K^2 test, also known as D'Agostino–Pearson normality test combines measures of sample skewness and kurtosis into a single statistic and compares it with the skewness and kurtosis of a standard normal distribution [26], [27].

The sample skewness (g_1) and kurtosis (g_2) are defined as:

$$g_1 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2)^{3/2}} \tag{2.26}$$

$$g_2 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2)^2} \tag{2.27}$$

where x_i are the observations of the data, $\bar{x}$ is the sample mean and n is the number of samples. Skewness describes the asymmetry of the data distribution. In particular, if $g_1 > 0$, the distribution is right-skewed (i.e. it has a longer tail to the right), and when $g_1 < 0$ the distribution is instead skewed to the left [26]. Kurtosis describes the tail heaviness of the distribution relative to a normal distribution. A value of $g_2 = 3$ corresponds to a normal distribution, $g_2 > 3$ indicates heavier tails and $g_2 < 3$ indicates lighter tails [29].

The test uses a transformation of the sample skewness and kurtosis before defining the test statistic [30], [31]:

$$Z_1(g_1) = \delta \sinh^{-1}(\frac{g_1}{\alpha\sqrt{\mu_2}}) \tag{2.28}$$

$$Z_2(g_2) = \sqrt{\frac{2}{9A}}\left((1 - \frac{2}{9A}) - \left(\frac{1 - \frac{2}{A}}{1 + \frac{g_2}{\sqrt{\beta_2}}}\right)^{1/3}\right) \tag{2.29}$$

where Z_1 and Z_2 are approximately standard normally distributed variables under the null hypothesis of normality. The quantities α, δ, μ_2, β_2, and A are functions of the sample size n, derived from the moments of the sampling distributions of skewness and kurtosis under the assumption of normality.

The test statistic is then defined as:

$$K^2 = Z_1^2 + Z_2^2 \tag{2.30}$$

2.4.4 Multivariate tests

Testing for multivariate normality requires different, often more complex methods, compared to the univariate case [32]. A common first step in many multivariate normality tests is therefore to standardize the observations so that location and covariance effects are removed [25]. For multivariate data let $X_1, \ldots, X_n \in \mathbb{R}^d$ denote the observations of the data, with sample mean $\bar{X}$ and sample covariance matrix S. The *scaled residuals* are defined as:

$$Y_{n,j} = S^{-1/2}(X_j - \bar{X}), \qquad j = 1, \ldots, n \tag{2.31}$$

This notation for Y will be used instead of the observations X to make the equations presented in the following sub-sections easier to read.

2.4.4.1 The Henze–Zirkler multivariate normality test

The BHEP class of tests evaluates multivariate normality by comparing the empirical characteristic function of the sample with the characteristic function of a multivariate normal distribution [25]. The Henze–Zirkler test is a developed BHEP test that works for any dimension and sample size [33], [34].

Let $Y_{n,1}, \ldots Y_{n,n}$ be the scaled residuals as defined in (2.31), the empirical characteristic function will then be:

$$\Psi_n(t) = \frac{1}{n}\sum_{j=1}^{n} \exp\left(it^\top Y_{n,j}\right) \qquad t \in \mathbb{R}^d$$

Furthermore, let:

$$\Psi_0(t) = \exp\left(-\frac{\|t\|^2}{2}\right)$$

be the characteristic function of the standard multivariate normal distribution. The BHEP test statistic is then given by [25], [34]:

$$\mathrm{BHEP}_{n,\beta} = n\int_{\mathbb{R}^d} \left|\Psi_n(t) - \Psi_0(t)\right|^2 w_\beta(t)\, dt, \tag{2.32}$$

where the Gaussian weight function is:

$$w_\beta(t) = \left(2\pi\beta^2\right)^{-d/2} \exp\left(-\frac{\|t\|^2}{2\beta^2}\right), \tag{2.33}$$

and $\beta > 0$ is a fixed constant that is defined as [25]:

$$\beta_n = \frac{1}{\sqrt{2}}\left(\frac{2d+1}{4}\right)^{1/(d+4)} n^{1/(d+4)}$$

In Equation (2.32), $t \in \mathbb{R}^d$ is the argument of the characteristic function, and the integral compares the empirical and theoretical characteristic functions over all values of t, with the weight function $w_\beta(t)$ controlling the contribution from different regions.

2.4.5 Mardia's test of multivariate skewness and kurtosis

Normality tests based on multivariate skewness and kurtosis are among the oldest and most widely used tests for assessing multivariate normality [25], [33]. Mardia's test compares the multivariate skewness and kurtosis structure of the standardized residuals with the corresponding values expected under multivariate normality [33]. A noticeable difference from the one dimensional moment tests of skewness and kurtosis is that Mardia's test treats skewness and kurtosis through two separate test statistics.

Let $Y_{n,1}, \ldots, Y_{n,n}$ be the scaled residuals as defined in equation (2.31). Mardia's sample skewness and kurtosis statistics are defined as [25], [35]:

$$b_{n,d}^{(1)} = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(Y_{n,i}^{\top} Y_{n,j} \right)^3 \tag{2.34}$$

$$b_{n,d}^{(2)} = \frac{1}{n} \sum_{i=1}^{n} \| Y_{n,i} \|^4 \tag{2.35}$$

where d denotes the dimension of the observations. These statistics measure deviations from multivariate normality by comparing the skewness and kurtosis structure of the scaled residuals to what is expected from a multivariate normal distribution. The skewness statistic $b_{n,d}^{(1)}$ measures asymmetric dependence between the components, while the kurtosis statistic $b_{n,d}^{(2)}$ measures how concentrated or heavy-tailed the observations are relative to the multivariate normal case. For data that are close to normal, $b_{n,d}^{(1)}$ should be close to zero, while $b_{n,d}^{(2)}$ should be close to $d(d+2)$ [25].

3

Methods

3.1 Creating a one-dimensional ITO model

To develop and evaluate the semigroup test in a controlled setting, a one-dimensional ITO model was created and trained. This provided both a test system for the proposed method and a clearer understanding of how ITO models behave in practice. This section describes the construction, training, and sampling procedure of the model.

3.1.1 One-dimensional Prinz potential

The dataset chosen for the ITO model to train on was several generated trajectories from a one-dimensional energy landscape called the Prinz potential [36], using the Deeptime Python library [37]. The potential used to generate the trajectories was given by the following equation (see Appendix A.1 for the potential landscape):

$$V(x) = 4\left(x^8 + 0.8e^{-80x^2} + 0.2e^{-80(x-0.5)^2} + 0.5e^{-40(x+0.5)^2}\right). \tag{3.1}$$

Using this potential, 200 trajectories were generated with a length of 100 000 time steps each. Figure 3.1 shows the empirical distribution of the sampled positions across all trajectories and time steps.

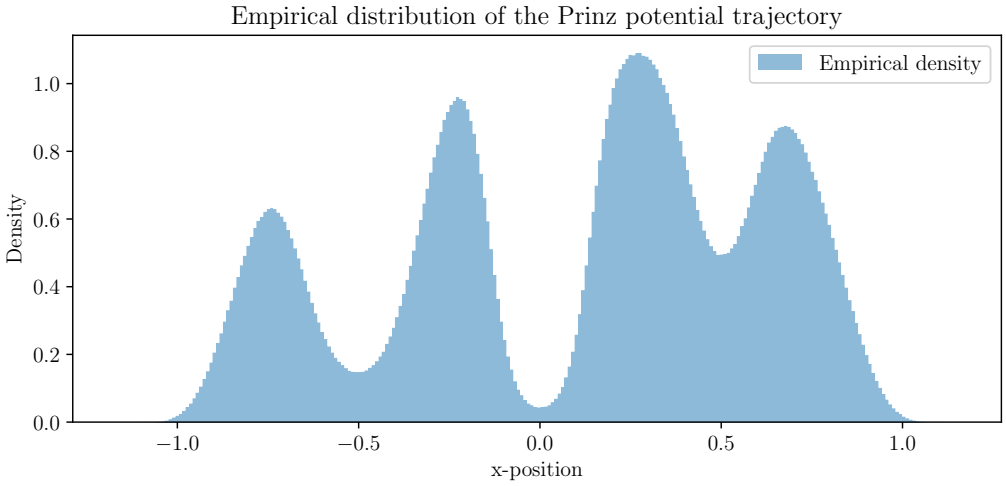


Figure 3.1: Histogram of the sampled positions from all time steps of the 200 generated trajectories in the Prinz potential giving a total of 20 000 000 points.

3.1.2 Architecture of the model

The model trained in this work was based on the ITO framework and implemented as a CFM model, where the vector field was parameterized by a fully connected neural network. The neural network will be referred to as an *Ultra Vector Field* (UVF). It consists of an initial input layer with dimension 146, followed by 4 hidden residual blocks with 256 dimensions, and finally an output layer. Its purpose was to approximate the time-dependent velocity:

$$v_\theta(t, x_t \mid x, \tau), \tag{3.2}$$

where θ denotes the trainable parameters of the network, x_t represents the state along the probability path at interpolation time t, while x, and τ provide the conditioning information. The output of the UVF was a scalar velocity that determines how x_t evolves during the learned flow. For a more detailed explanation of the vector field see Appendix A.2.

3.1.3 Training of the model

The training data described in Section 3.1.1 was used by a data loader that loaded the full trajectory data into memory and constructed pairs of configurations separated by a sampled lag time. The maximum lag was set to $\tau_{max} = 200$ and the batch size to 4096. For each data point, the data loader selected an initial position x from one of the trajectories. Then it sampled a lag $1 \leq \tau \leq \tau_{\max}$ and selected the corresponding position x_τ from the same trajectory to be used as the target. The selected lag time τ was sampled uniformly in log-space, causing shorter lag times to be sampled more frequently than longer lag times, while still being exposed to the full range of possible lag times.

The data loader also sampled a base point for each sample in the batch:

$$z_0 \sim \mathcal{N}(0, I)$$

The CFM model was then trained to learn a probability flow from this base distribution to the lagged target distribution. For a sampled time $t \sim \mathcal{U}(0, 1)$, an intermediate point along the probability path was constructed as:

$$x_t = (1 - t)z_0 + tx_\tau + \sigma\epsilon \tag{3.3}$$

where $\sigma = 0.01$ is a noise parameter and $\epsilon \sim \mathcal{N}(0, I)$ is used to vary the size of the noise. This follows the general CFM framework, but differs from the standard linear CFM interpolation by adding a small independent Gaussian perturbation $\sigma\epsilon$. The purpose of this perturbation is to smooth the conditional target distribution around each lagged sample x_τ, rather than making the endpoint collapse onto a single point [18]. The target velocity for the conditional flow matching objective was:

$$u_t = x_\tau - z_0 \tag{3.4}$$

and the model was trained by minimizing the mean squared error, which corresponds with the CFM objective as described in equation (2.19):

$$\mathcal{L}(\theta) = \mathbb{E}\left[\left\|v_\theta(t, x_t \mid x, \tau) - u_t\right\|^2\right] \tag{3.5}$$

Given this, the training taught the model to transport samples from a normal distribution to the distribution of configurations observed after a physical lag τ, conditioned on the initial state x. The ITO model mainly used in this thesis was trained on the dataset described in Section 3.1.1 for 1000 epochs.

3.1.4 Sampling of the model

To evaluate and use the CFM after training, samples were generated by integrating the learned vector field from interpolation time $t = 0$ to $t = 1$. For the first sampling step, the model was conditioned on an initial state x_0, and sampling started from a base sample distribution $z_0 \sim \mathcal{N}(0, I)$. The vector field was then evaluated repeatedly along the flow and the sample was updated using the Euler method for N_{ODE} integration steps with step size $\Delta t = \frac{1}{N_{\text{ODE}}}$. The Euler update was given by

$$z_{k+1} = z_k + \Delta t \, v_\theta(t_k, z_k \mid x_0, \tau_1) \qquad t_k = k\Delta t \tag{3.6}$$

for $k = 0, \ldots, N_{\text{ODE}} - 1$. Here, x_0 is the conditioning state, τ_1 is the physical lag time, and z_k is the current sample along the learned flow. After the final Euler step, the generated sample was given by $x_1 \approx z_{N_{\text{ODE}}}$. Thus, forward sampling maps a base sample z_0 to a generated sample x_1, conditioned on x_0:

$$z_0 \xrightarrow{v_\theta(t, z_t \mid x_0, \tau_1)} x_1 \tag{3.7}$$

To sample a trajectory, this procedure was repeated recursively; after the first generated sample x_1 was obtained, it was used as the new conditional state for the next sampling step.

For the reverse sampling, the Euler step was instead applied from the interpolation time $t = 1$ to $t = 0$. In order to achieve this, the Euler update in equation (3.6) was rewritten as:

$$z_{k-1} = z_k - \Delta t \, v_\theta(t_{k-1}, z_k \mid x_0, \tau_1) \qquad t_k = k\Delta t \tag{3.8}$$

for $k = N_{\text{ODE}}, \ldots, 1$. The rest of the process remained the same and after the final Euler step the generated distribution was expected to be equal to the input z_0.

To decide the number of ODE steps, N_{ODE}, used when sampling the model for the CK-test, a simple experiment comparing the models sampled distribution with the original base distribution for different set values of N_{ODE} steps was made. The sampling was done with a lag value of 1, batch size of 4096 and for 1000 steps. Figure 3.2 shows that no noticeable improvements can be seen after a value of $N_{ODE} = 150$. For robustness, a slighter higher value of $N_{ODE} = 200$ was the value used for all the univariate CK-tests.

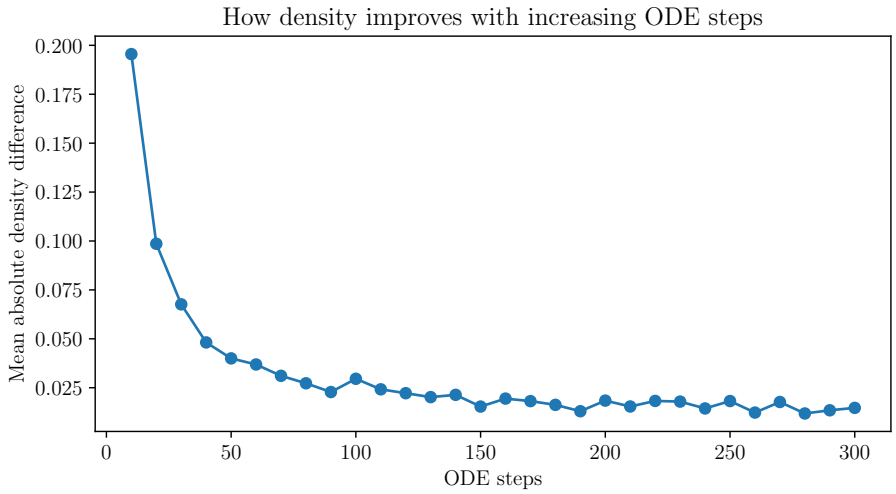


Figure 3.2: Density difference between the sampled distribution and the simulated base distribution. After about 150 ODE steps, no noticeable difference is observed.

3.2 Developing the univariate CK-test

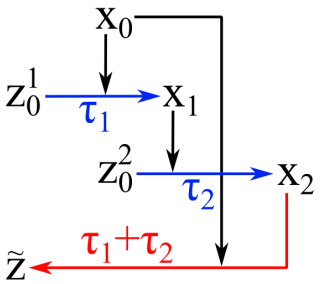


Figure 3.3: A schematic of the two step CK-test. The blue arrows (pointing right) represent a forward sampling with the corresponding lag time τ and the red arrow (pointing left) represent the reverse sampling. The black arrows (pointing down) represent the conditional positions used in the respective sampling. z_0^1 and z_0^2 are both sampled from a normal distribution and if the test is successful, then $\tilde{z}$ should also be a normal distribution.

The development of the CK-test was based on the one-dimensional ITO model. The concept of the test is to sample the model forward in time in two separate steps using two arbitrary lag times τ_1 and τ_2, followed by a sample backwards in one step with the lag time $\tau_{\text{total}} = \tau_1 + \tau_2$, as seen in Figure 3.3. Since the base distribution z_0 used by the model is a standard normal distribution, the distribution after sampling the model backwards $\tilde{z}$ should also be a normal distribution, if the model satisfies the semigroup property. To assess whether the distribution obtained after backwards sampling was Gaussian, three different metrics were implemented to provide a robust evaluation: the Shapiro–Wilks normality test, the D'Agostino K^2 test based on skew and kurtosis and the energy distance ratio.

In order to evaluate the model for many different configurations of lag times τ_1 and τ_2, the test was implemented as two separate evaluations. Firstly, the two lag times were kept equal while varying the total lag $\tau_1 + \tau_2$. This was done in order to evaluate if the semigroup property was satisfied regardless of the size of the lag time. Secondly, the total lag was instead kept constant while the difference between the lag times $(\tau_1 - \tau_2)$ was increased. This was done in order to evaluate if the model was able to pass the test regardless of any imbalance between the two lag times. In both these evaluations, each measurement was made several times in order to reduce any stochastic errors that may occur when sampling the model.

3.2.1 Univariate metrics

The metrics implemented to evaluate the univariate CK-test were the Shapiro–Wilk normality test, D'Agostino's K^2 test based on skewness and kurtosis, and the energy distance ratio. For the Shapiro–Wilk and D'Agostino's K^2 tests, p-values were calculated under the null hypothesis that the samples were drawn from a normal distribution. The rejection rate of the null hypothesis was then computed using a significance level of $\alpha = 0.05$, and used as the final metric. This means that, for a perfect model, the expected rejection rate is approximately equal to the significance level.

To make the energy distance easier to interpret, it was used to make a normalized energy ratio R_E. The numerator was given by the energy distance between the sampled distribution and the original base distribution. The denominator was computed as a baseline energy distance between the base distribution and an independently sampled standard normal distribution. This baseline was estimated by repeating the calculation five times and taking the average. This can also be expressed as an equation:

$$R_E = \frac{E_{n,m}(\tilde{Z}, Z_0)}{\frac{1}{5}\sum_{k=1}^{5} E_{n,m}(Z_k, Z_0)} \qquad Z_k \sim \mathcal{N}(0, I). \tag{3.9}$$

where E is the empirical energy statistic described in Equation (2.24). Values below 1 indicate that the sampled distribution is closer to the base distribution than the independent normal baseline, while values above 1 indicate a deviation from the base distribution and thus a deviation from normality.

3.3 Multivariate ITO model

To develop and assess the semigroup test on multivariate data, a multivariate ITO model was required. Since data of higher dimensions and more complex structures require more advanced networks with more training, creating one such model was outside the scope of this project. Instead, the model used was a *Transferable Implicit Transfer Operator* (TITO) developed by Juan Viguera Diez, Mathias Schreiner and Simon Olsson in their paper "Transferable Generative Models Bridge Femtosecond to Nanosecond Time-Step Molecular Dynamics" [7] and will henceforth be referred to as the TITO model.

Similarly to the one-dimensional ITO model, the TITO model is also a CFM model. The specific TITO model used for this project was trained on the MDQM9-nc dataset [38], [39], which contains MD simulations for small non-cyclic molecules. For a more in-depth description of the model and its architecture please refer to the original paper [7].

A test molecule from the MDQM9-nc dataset was used in order to be able to sample the TITO model. This molecule had the composition $C_7H_8O_2$ and the exact structure can be seen in Figure 3.4. The actual sampling of trajectories was done using the Euler method. The sampling procedure was the same as for the one-dimensional ITO model described in Section 3.1.4, except that the positions were three-dimensional and the learned vector field was replaced by the TITO vector field. The reverse sampling was also done in the same way by reversing the Euler update equation.

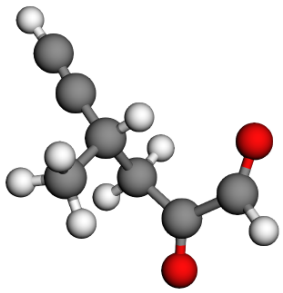


Figure 3.4: Visual representation of the molecule used when sampling the TITO model. Its composition is $C_7H_8O_2$, the IUPAC name is *(S)-4-methyl-2-oxohex-5-ynal* and the SMILES code is C[C@@H](CC(=O)C=O)C#C.

3.3.1 Evaluating the reverse sampling of the TITO model

In order to evaluate the TITO models ability to reverse the sampling process, two experiments were conducted. In the first experiment, samples were sampled forward with a lag time τ and then subjected to reverse sampling using the same lag. The value of τ was varied in order to evaluate the stability across different lag times and several measurements were made for each value of τ in order to reduce the effect of stochastic errors. In the second experiment, the lag was kept constant and instead the number of ODE steps used in the sampling was varied. The output from the reverse sampling was compared to the Gaussian noise sampled as the input z_0 and the accuracy of the recreation was evaluated using the Henze–Zirkler multivariate test, Mardia's skewness and kurtosis test, as well as the energy distance ratio.

3.4 Developing the multivariate CK-test

After the CK-test had been developed for the univariate model, it was extended to multivariate data. The concept remained the same as for the univariate case, but since the univariate tests for normality that were implemented were not compatible with multivariate data, the metrics used for the CK-test had to be revised. More specifically, the previous normality tests were replaced by the Henze–Zirkler multivariate test and Mardia's skewness and kurtosis test. The energy distance ratio, however, could be adapted and therefore remain as a metric.

As with the univariate CK-test, the multivariate CK-test was tested across two separate evaluations: firstly while varying the total lag, followed by varying the lag imbalance. In order to reduce any stochastic error, each test was run several times and returned an average across these runs.

3.4.1 Multivariate metrics

The metrics used for the multivariate CK-test were the Henze–Zirkler test, Mardia's skewness and kurtosis test and the energy distance ratio. In similar fashion to the univariate case, described in Section 3.2.1, the Henze–Zirkler and Mardia's tests were used to calculate p-values and get a rejection rate with a significance level of $\alpha = 0.05$. The energy distance ratio was calculated in the same way as the univariate case.

3.4.2 Developing an alternate multivariate CK-test

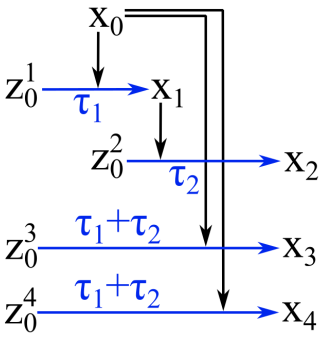


Figure 3.5: A schematic of the alternate CK-test. The blue arrows (pointing right) represent a forward sampling with the corresponding lag time τ and the black arrows (pointing down) represent the conditional positions used in the respective sampling. z_0^1, z_0^2, z_0^3 and z_0^4 are all sampled from a normal distribution and if the test is successful, then x_2 should have the same distribution as x_3. The sampling of x_4 is used in order to create a baseline.

For the purpose of creating a baseline for the TITO model, an alternate CK-test was developed that only utilized forward sampling. The first part of the test, sampling two steps using two separate lag times, remained the same as the original CK-test, but the second step of sampling the resulting distribution back to a normal

distribution was changed. Instead, the model was sampled forward from a normal distribution with a lag time of $\tau_1 + \tau_2$ and the resulting distribution was compared to the distribution generated from the first part. A schematic of this alternate test can be seen in Figure 3.5.

The problem with this new method was that the resulting distributions x_2 and x_3 were not expected to be normal, which meant that the metrics used for the original CK-test were not applicable, with the exception of the energy distance ratio. This metric is not necessarily a normality test, but instead a metric used to compare two distributions. In order to apply it to this case, the baseline had to be adjusted since the distribution was no longer expected to be Gaussian. The baseline should represent the energy distance that arises as an effect of the variability of the sampling. Therefore, the longer sampling with lag time $\tau_1 + \tau_2$ was performed twice, but with different original noise z_0, in order to generate both x_3 and x_4, as seen in Figure 3.5. The energy distance between these two distributions was then used as the baseline in order to determine the energy distance ratio:

$$R_E = \frac{E_{n,m}(X_2, X_3)}{E_{n,m}(X_4, X_3)} \tag{3.10}$$

Unlike the original CK-test, this energy distance ratio was no longer determined in the latent space, but instead in the molecular space. Therefore, in order to evaluate the semigroup structure, the samples were projected using *Time-Lagged Independent Component Analysis* (TICA) [40], [41]. Firstly, the MD simulation trajectory for the testing molecule was extracted from the MDQM9-nc dataset [38]. This was then converted from Cartesian coordinates to torsion angles, considering only the torsions between the heavier atoms (carbon and oxygen) and the resulting trajectory was used in order to fit the TICA. The test was then run in Cartesian coordinates, but before applying the metric, the outputs x_2, x_3, x_4 were converted to the torsion angles instead and then transformed using the fitted TICA.

3.5 Comparison with untrained models

Untrained models were evaluated to investigate whether CK-test performance is correlated with the actual learned performance of the model, and whether the ITO architecture has an inductive bias toward satisfying the CK-test even before training. If untrained models can pass the CK-test, this indicates that there may not be a direct correlation between learning the correct dynamics and satisfying the semigroup property.

To test this, 100 independently initialized but completely untrained models were evaluated using the same architecture as the one-dimensional CFM model. Each model was tested on the same lag pair, making it possible to compare the best, worst, and average CK-test performance of untrained models. This procedure was also done with the three-dimensional TITO model.

3.5.1 Experiment on the vector field transportation

Based on the results from applying the CK-test to untrained models, a small experiment was made to investigate whether the untrained models transported the base noise significantly. The motivation was to test whether the CK-test performance of some untrained models could be explained by them behaving approximately like identity maps.

To estimate the typical magnitude of the vector field, the *root mean square* (RMS) magnitude of the vector field was evaluated. For a batch of samples x_t, the RMS vector-field magnitude at time t was computed as:

$$v_{\mathrm{RMS}}(t) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|v_\theta(t, x_t^{(i)})\right|^2} \tag{3.11}$$

where N is the number of samples in the batch. This value measures the average magnitude of the velocity field predicted by the model at a given interpolation time t. The statistic was evaluated over a grid of $n_t = 50$ time points in the interval $t \in [0, 1]$, after which the mean and standard deviation over time were recorded. The experiment was repeated for 500 independently initialized untrained models and compared to the trained model.

3.5.2 Applying the CK-test during model training

The CK-test was also applied to the same model at several points during training to evaluate whether, and at what stage, CK-test performance improved. For this experiment, the initial model was chosen from the set of untrained models and had approximately average CK-test performance before training. The untrained model was trained for 100 epochs and the CK-test was applied 10 times, evenly spaced, during every epoch.

4

Results

4.1 Performance of the univariate model

The model was trained on the dataset for 1000 epochs and sampled using a lag time of $\tau = 1$ to evaluate how well the model could capture the equilibrium distribution from the training data. Figure 4.1 shows the distribution from the model compared to the training data with 200 bins, the absolute density difference is about 0.018. This, along with visual inspection of the figure, indicates that the model is able to capture the equilibrium distribution fairly well and not miss any of the potential wells. While it does slightly favor the three larger wells and underestimates the smallest, it was determined to be accurate enough for the purposes of this project.

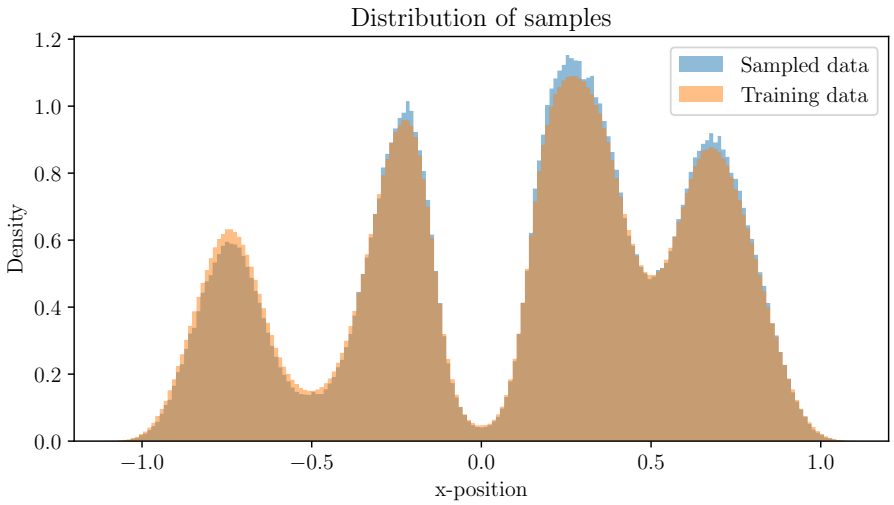


Figure 4.1: Two overlapping histograms showing the empirical distribution of the training data (blue) and the sampled distribution from the model (orange). The model was sampled with a batch size of 4096, over 10 000 time steps, where each step had a lag of $\tau = 1$.

4.2 The univariate CK-test

This section will present the results obtained from creating and applying the CK-test on the trained univariate ITO model.

4.2.1 Increasing total lag

The developed CK-test was applied while varying the total lag, meaning that $\tau_1 = \tau_2$ while $\tau_1 + \tau_2$ varied. The total lag was increased from 2 to 200 in increments of 4 and the CK-test was run 20 times for each total lag. Figure 4.2 shows the resulting rejection rates and energy distance from these runs. The average rejection rate for both the Shapiro–Wilk test and D'Agostino K^2 test is about the same as the significance level $\alpha = 0.05$, which indicates that the model has managed to, on average, reconstruct a distribution consistent with normality and passed the test. The energy distance ratio averages 0.621, which also indicates that the model succeeds in reconstructing the noise. Additionally, all three tests show no significant change in the rate of rejection with the increased total lag.

4.2.2 Increasing lag imbalance

Subsequently, the CK-test was applied while increasing the lag imbalance, which means that τ_1 was increased and τ_2 decreased after every run of the CK-test, keeping the total lag fixed at 200. The imbalance was increased in increments of 2 and the CK-test was run 20 times for each lag pair. Figure 4.3 shows the results and gives the model an average rejection rate of 0.062 for the Shapiro–Wilk test and 0.088 for the D'Agostino K^2 test. This is slightly higher than the significance level of 0.05, which implies that the model struggles slightly at certain imbalance values. The energy distance ratio shows an average of 0.712, which indicates that the model manages to successfully reconstruct the noise. However, the energy distance ratio seems to struggle slightly at higher imbalance values, as opposed to the other two tests which show no significant change with the increased imbalance.

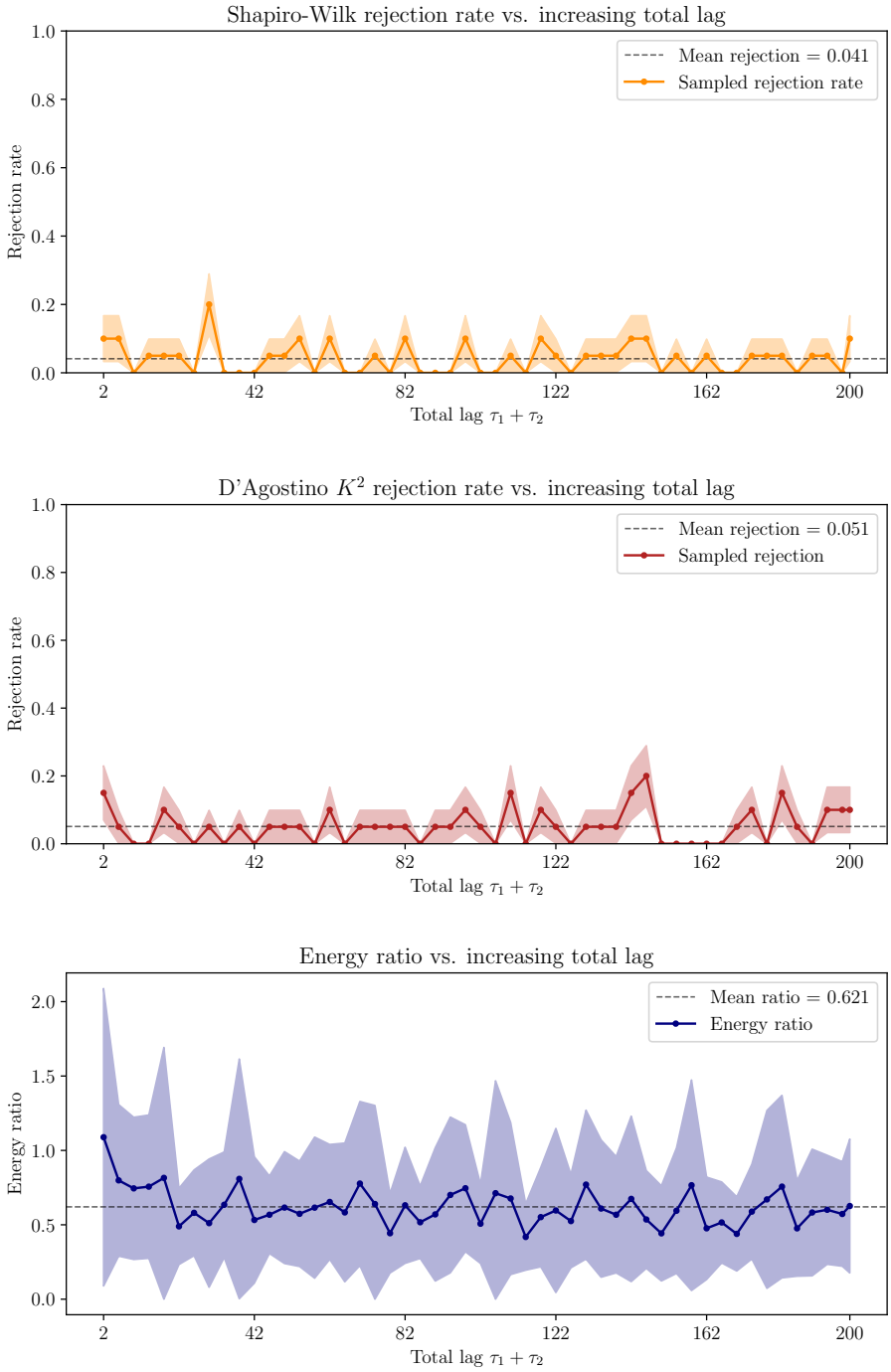


Figure 4.2: CK-test results for the univariate ITO model with varying total lag time. The total lag was increased from 2 to 200 in increments of 4 and the CK-test was run 20 times for each total lag.

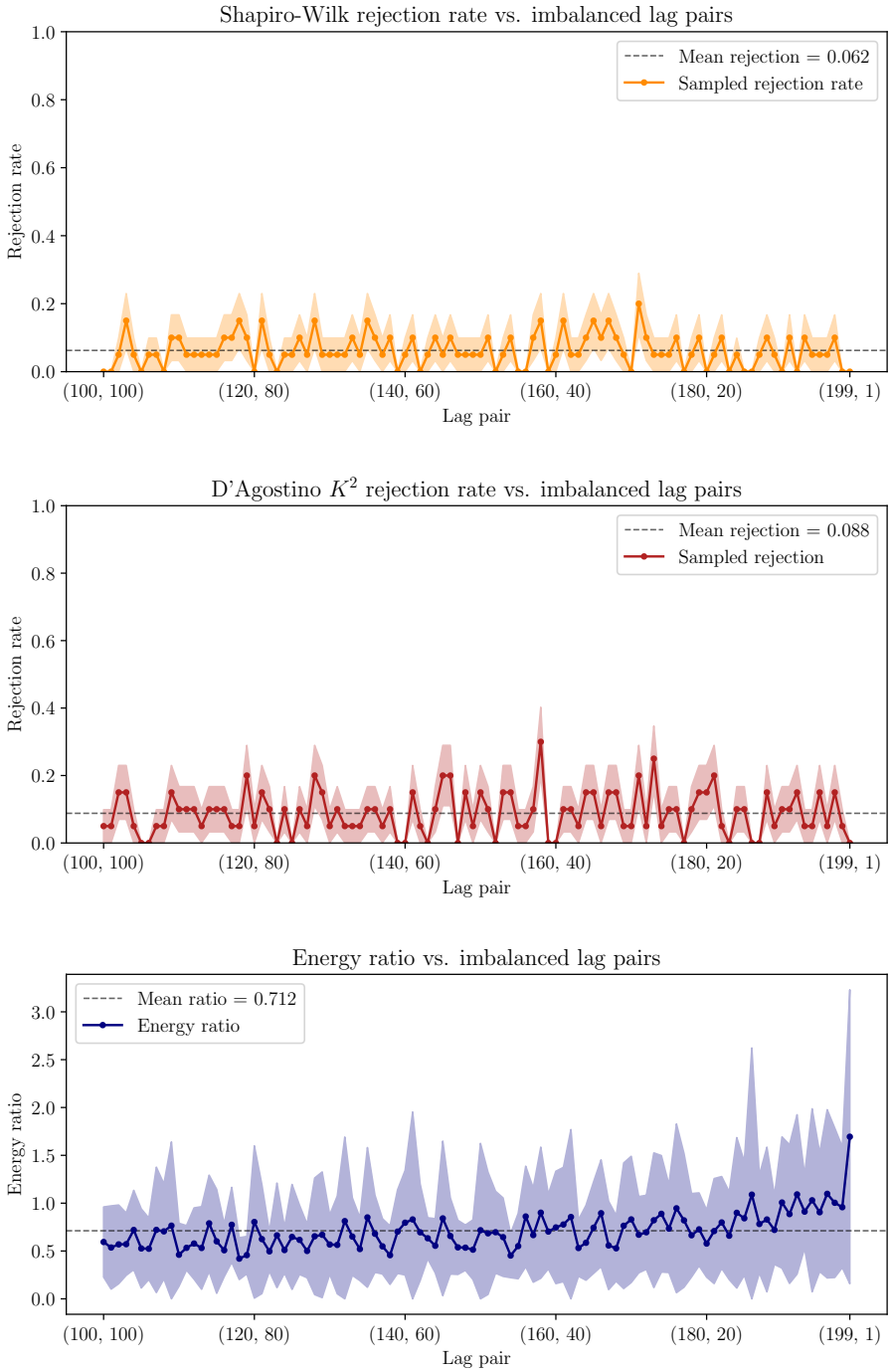


Figure 4.3: CK-test results for the univariate ITO model with varying imbalance between the two lag times. The total lag was fixed at 200 and the difference between the two lag times was increased by 2 each step. The CK-test was run 20 times for each lag pair.

4.3 The multivariate CK-test

This section will present the results from applying the multivariate CK-test on the TITO model.

4.3.1 Increasing total lag

The multivariate CK-test was applied to the TITO model with $\tau_1 = \tau_2$, while the sum of the two were varied. The total lag was increased from 200 to 1800 in increments of 20 and the CK-test was run 20 times for each total lag. Figure 4.4 shows the resulting metrics, with both the Henze–Zirkler test and Mardia's test displaying an average rejection rate of above 0.9. This suggests that the model is essentially unable to accurately replicate the normal distribution across any time lag. Additionally, the energy distance is on average 6.830 times greater than the expected distance between two Gaussian distributions, which further indicates that the TITO model does not pass the multivariate CK-test under this evaluation.

4.3.2 Increasing lag imbalance

The multivariate CK-test was then applied to the TITO model while varying the difference between lag τ_1 and τ_2. The total lag was fixed at 1000 and the difference $(\tau_1 - \tau_2)$ was increased by 20 each step. The CK-test was run 20 times for each lag pair. The results can be seen in Figure 4.5 and the average rejection rate is above 0.9 for both the Henze–Zirkler test and Mardia's test, which suggests that the model fails to reconstruct the normal distribution. Moreover, the energy distance ratio has an average of 7.055, which also suggest that the model fails across all lag pairs.

4.4 Reversing the multivariate model

This section will present the results from reversing the TITO model and applying the metrics from the CK-test.

4.4.1 Increasing the total lag

The metrics of the multivariate CK-test was applied to a single forward and reversed step of the model with a lag time varying from 200 to 1800 in increments of 20. This sampling and evaluation was done 20 times for each lag time. Figure 4.6 shows the results, with the Henze–Zirkler test showing an average rejection of 0.999 and Mardia's kurtosis test showing an average rejection of 1.000. In contrast, Mardia's skewness test shows an average rejection rate of only 0.006, which is below the significance level of 0.05. This indicates that the recreated distribution maintains its symmetry, but lacks the overall shape in order to be considered a successfully recreated Gaussian distribution. Furthermore, the energy distance ratio has an average of 27.155, which is much higher than the target value of 1, indicating that the reverse sampling fails to recreate the original normal distribution.

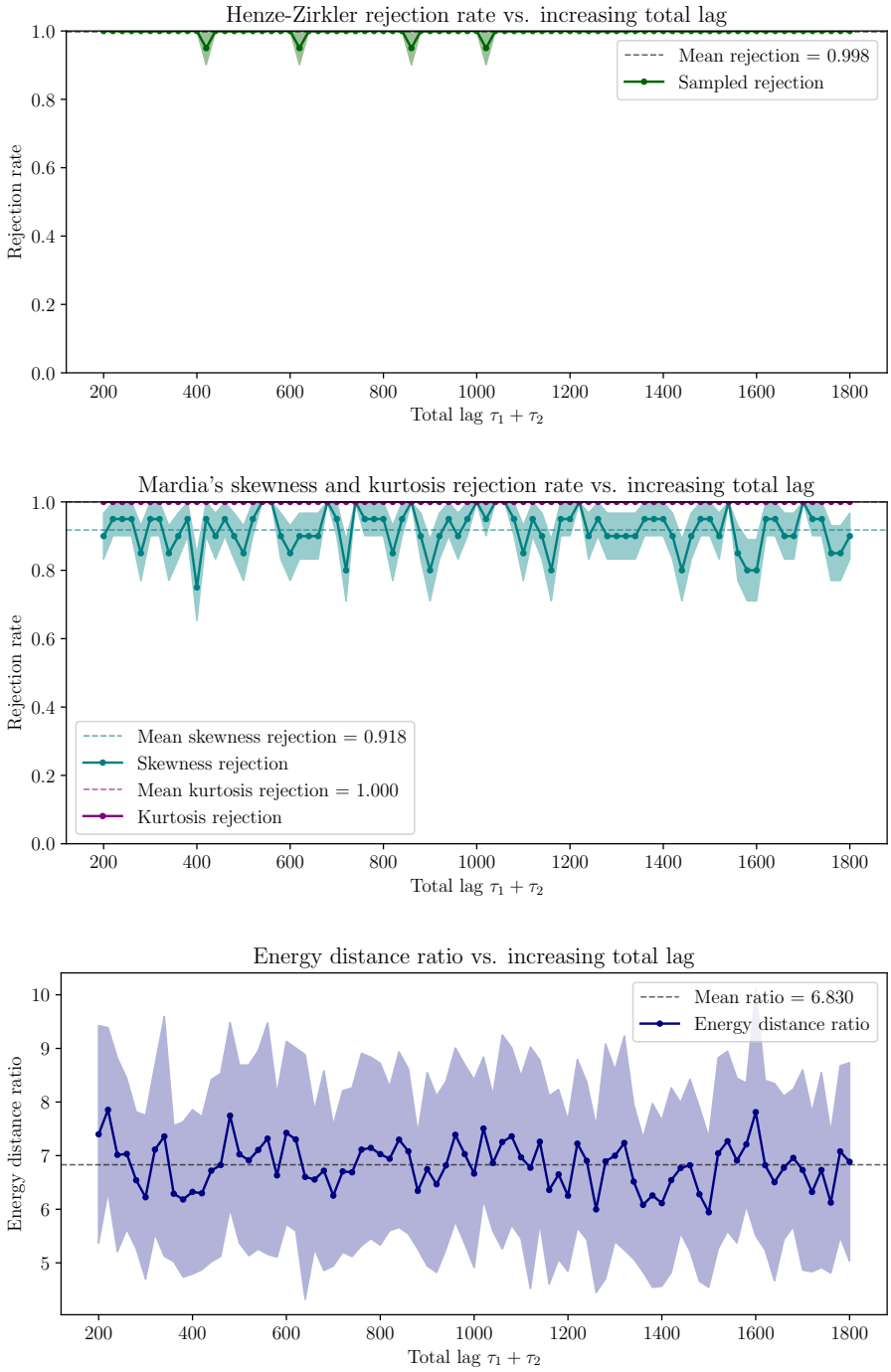


Figure 4.4: CK-test results for the multivariate TITO model with varying total lag time. The total lag was increased from 200 to 1800 in increments of 20 and the CK-test was run 20 times for each total lag.

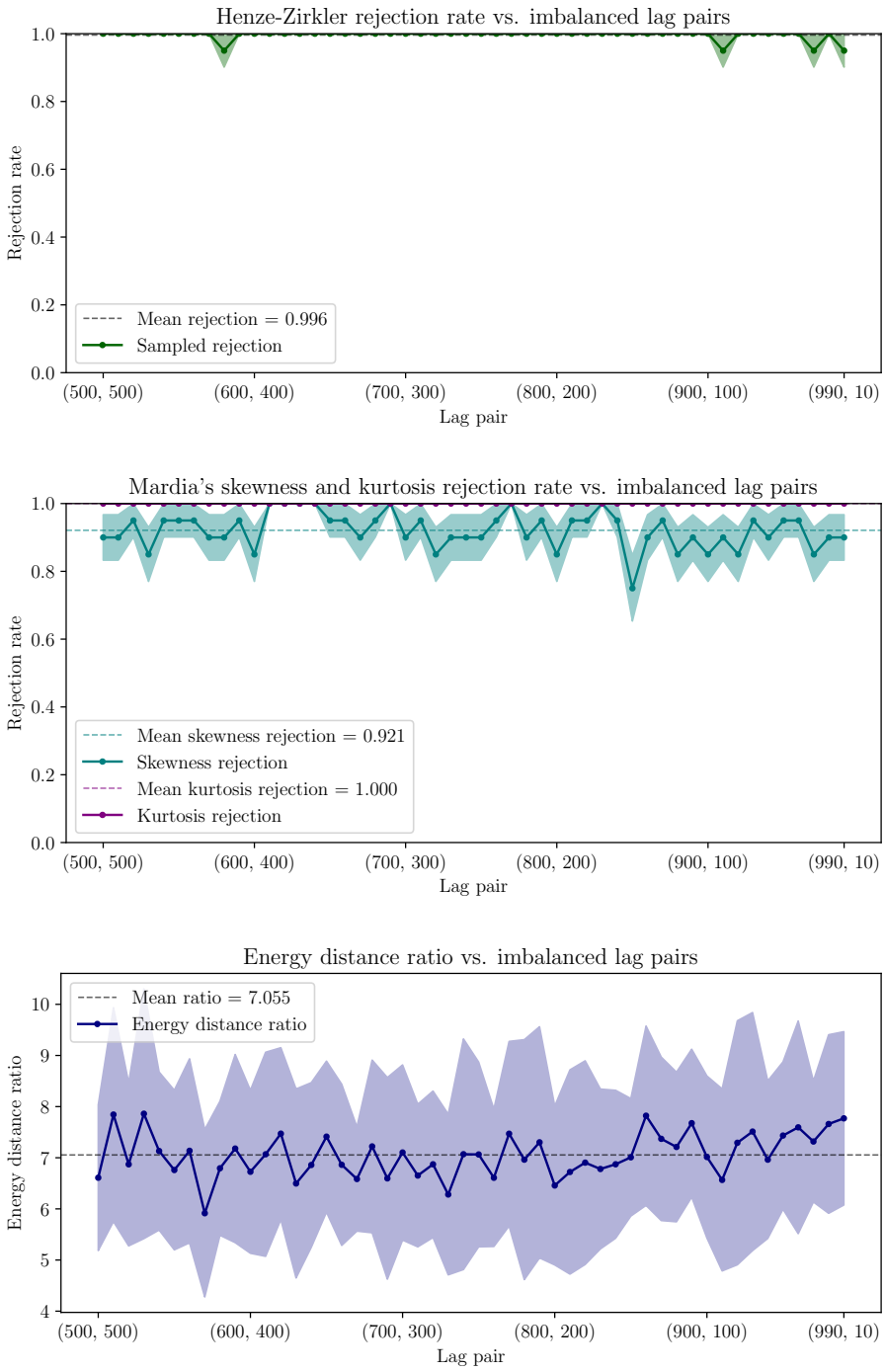


Figure 4.5: CK-test results for the multivariate TITO model with varying imbalance between the two lag times. The total lag was fixed at 1000 and the difference between the two lag times was increased by 20 each step. The CK-test was run 20 times for each lag pair.

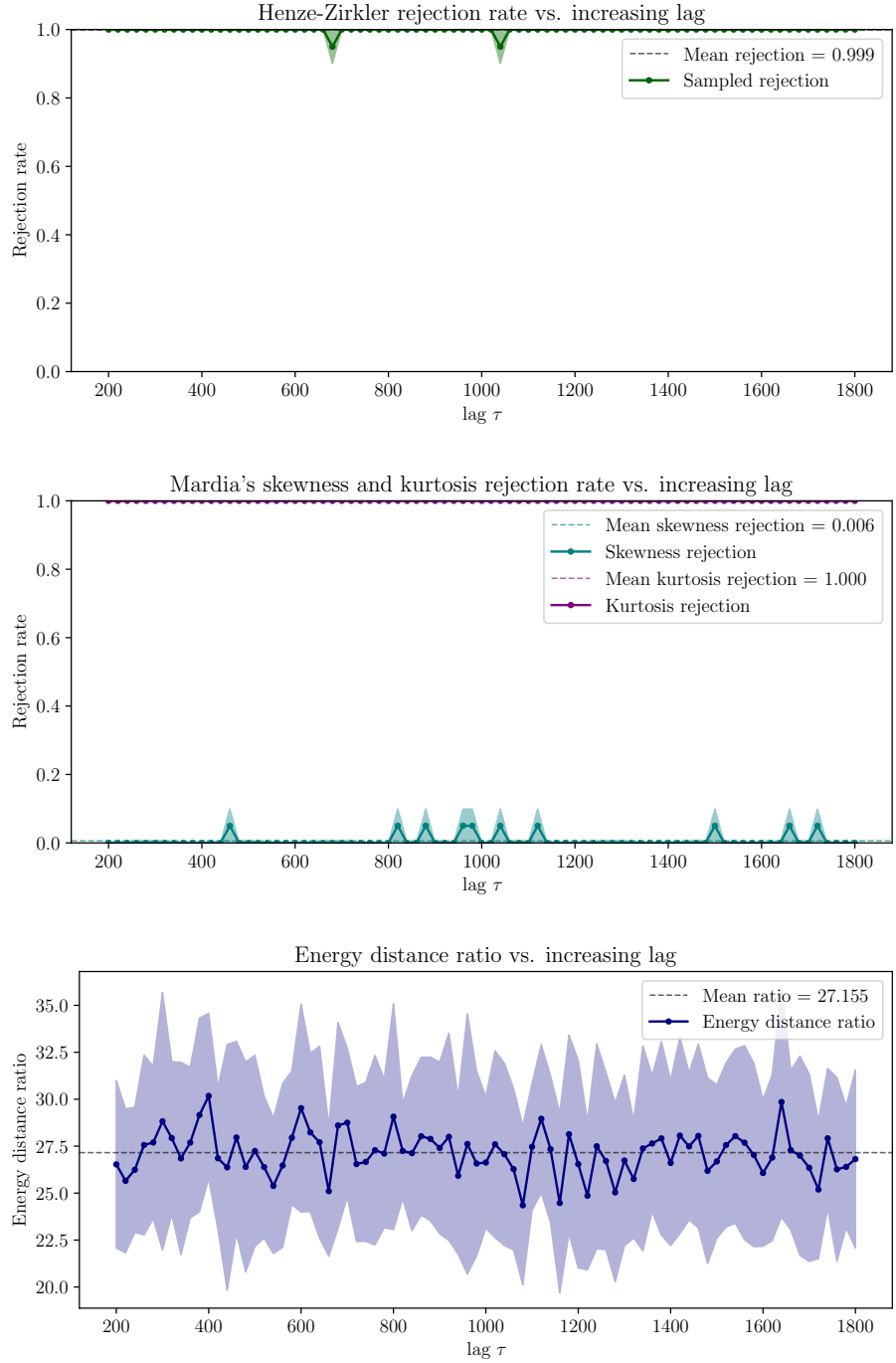


Figure 4.6: Results from sampling the multivariate TITO model forward for a single time step and then reversing it. The time lag used was increased from 200 to 1800 in increments of 20 and the test was run 20 times for each lag.

4.4.2 Increasing the number of ODE steps

The metrics of the multivariate CK-test was applied to a single forward and reversed step of the model while varying the number of ODE steps used in both the forward and reversed sampling. The number of ODE steps were varied on the interval 20 to 1000 and the evaluation was done 10 times for each value. The results can be seen in Figure 4.7 and all metrics, with the exception of Mardia's skewness test, show an improvement in performance with the increase in ODE steps. The Henze–Zirkler test seems to converge the fastest, followed by the energy distance ratio and lastly Mardia's kurtosis test, which seems to converge at the slowest rate. A complete evaluation of the reversed sampling using 1000 ODE steps while varying the lag can be found in Appendix A.3.1.

4.5 The alternate multivariate CK-test

This section will show the results from applying the alternate CK-test on the TITO model.

4.5.1 Increasing total lag

The alternate CK-test was applied to the TITO model with $\tau_1 = \tau_2$ while gradually increasing the total sum of the two. The total lag was increased from 200 to 1800 in increments of 20 and the CK-test was run 20 times for each total lag. Since the output isn't expected to be a normal distribution, only the energy distance ratio metric could be used to evaluate the performance. Figure 4.8 shows the average ratio of 5.947, which is lower than the 6.830 from the original CK-test method (as seen in Figure 4.4), but still significantly higher than the target value of 1. This suggests that, although the result is relatively consistent across lag times, the TITO model does not satisfy the semigroup property under this test.

4.5.2 Increasing lag imbalance

Subsequently, the developed alternate CK-test was applied to the TITO model while instead varying the difference between the two lag times τ_1 and τ_2. The total lag was fixed at 1000 while the difference between the two lag times was increased by 20 each step. The CK-test was run 20 times for each lag pair. Figure 4.9 shows the resulting energy distance ratio with an overall average of 5.042. This is also a lower ratio when compared to the original CK-test (as seen in Figure 4.5), but still significantly higher than the target value. This would further suggest that the TITO model fails to satisfy the semigroup property under the conditions evaluated by this test.

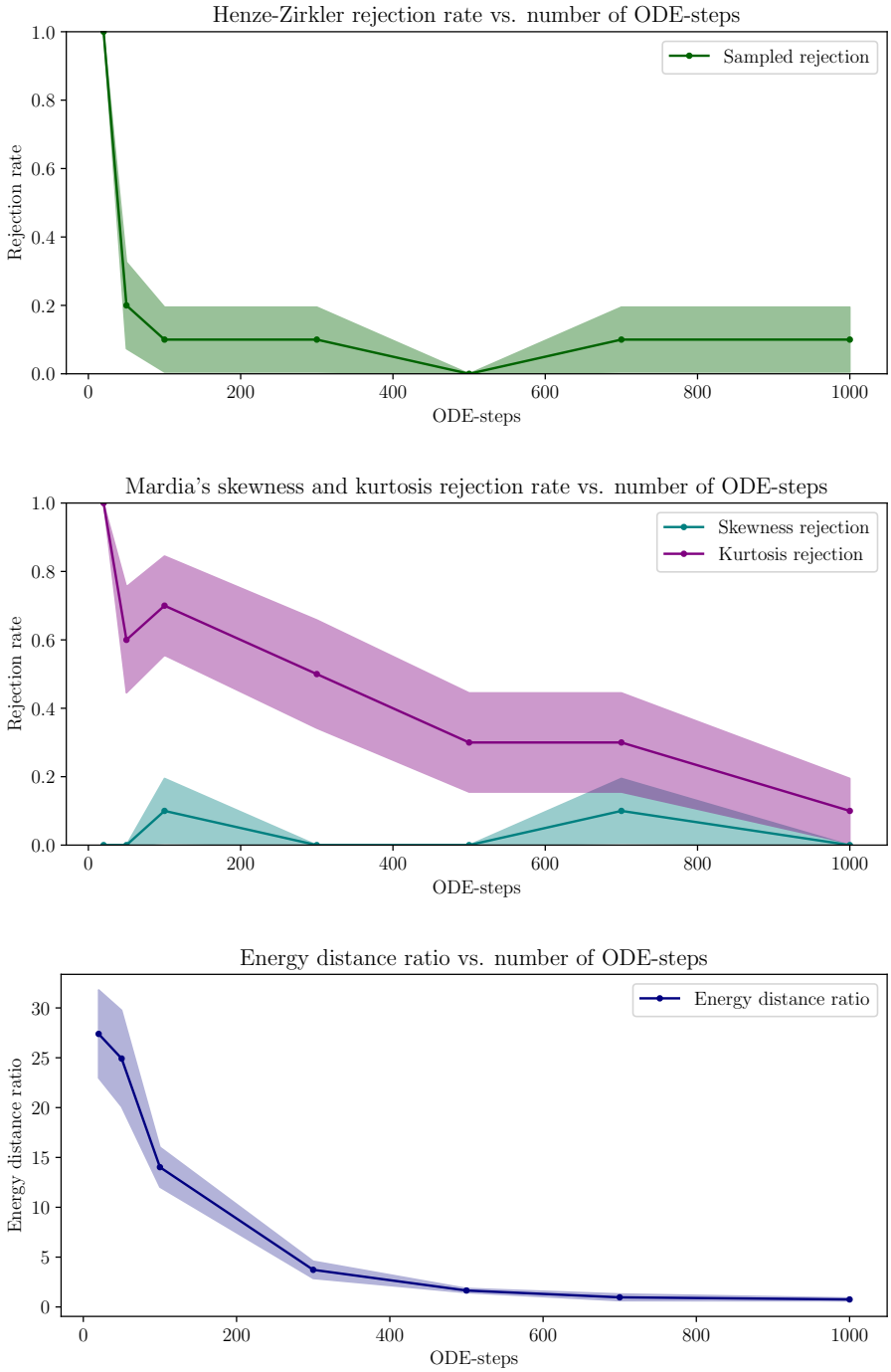


Figure 4.7: Results from inverting the multivariate TITO model, simulating a single time step forward and then reversing it. The model was evaluated on $\tau = 500$. The number of ODE steps used when simulating was increased from 20 to 1000 and the test was run 10 times for each value.

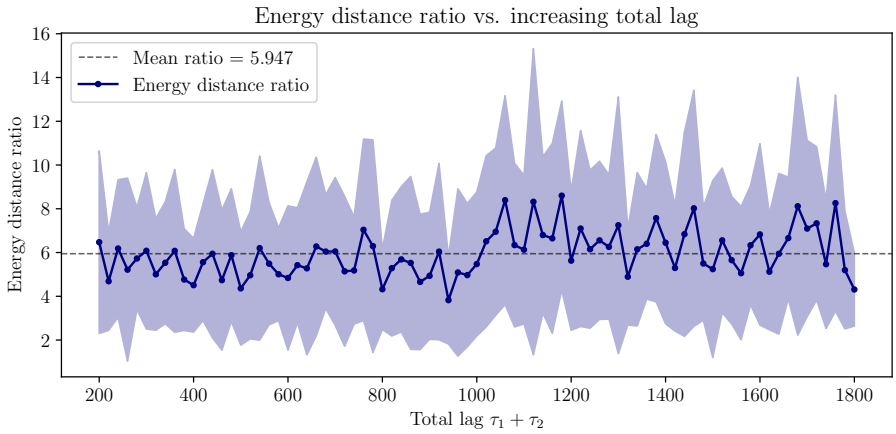


Figure 4.8: Energy distance ratio from the alternate CK-test when applied to the TITO model with varying total lag. The total lag was increased from 200 to 1800 in increments of 20 and the CK-test was run 20 times for each total lag.

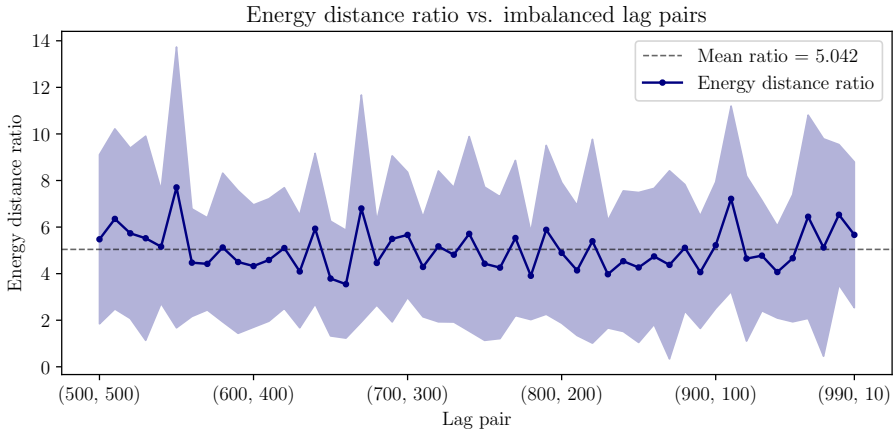


Figure 4.9: Energy distance ratio from the alternate CK-test when applied to the TITO model with varying imbalance between the two lag times. The total lag was fixed at 1000 and the difference between the two lag times was increased by 20 each step. The CK-test was run 20 times for each lag pair.

4.6 Testing untrained models

This section presents the performance of the CK-test on untrained ITO and TITO models. It will also investigate how the results of the CK-test evolves during the process of training an ITO model.

4.6.1 Randomly initialized univariate models

Untrained univariate ITO-models were investigated using the CK-test, where each model was tested 20 times using the lag pair $(100, 100)$. Figure 4.10 shows the results across 100 different untrained models. The D'Agostino K^2 test has an average rejection rate of 0.050, a score equal to the significance level, indicating that the untrained models are able to reconstruct a normal distribution. In contrast, the Shapiro–Wilk test has an average rejection rate of 0.937, instead indicating that the models are unable to reconstruct a normal distribution. While some models perform significantly better than the overall average, no model ever reaches a rejection rate close to the significance level. Lastly, the average energy ratio for all untrained models was 1.653, which indicates that the untrained models struggles to successfully reconstruct the noise. Although it is worth noting that some of the models have values lower than 1 and some have values of up to 4, showing a large disparity between the different random initializations of the models.

4.6.2 Randomly initialized multivariate models

Untrained models with the same architecture as the TITO model were evaluated using the multivariate CK-test. A total of 100 different models were initialized and each model was evaluated 20 times using the lag pair $(500, 500)$. Figure 4.11 presents the results, with the Henze–Zirkler test showing an average rejection rate of 0.059 across all models. This is fairly close to the significance level $\alpha = 0.05$, which suggests that the untrained models are able to accurately replicate the Gaussian distribution. Mardia's skewness test also shows a rejection rate of 0.057, which further supports this claim. The average rate of 0.071 for the kurtosis instead implies that some models struggle with the recreation. However, there is one model that performs significantly worse than the others in Mardia's kurtosis test and if this outlier is removed, then this average is reduced to 0.063, which is closer to the significance level. Lastly, the energy distance ratio shows an average of 1.094, which is close to the target ratio and therefore indicates that the models manage to recreate the original distribution.

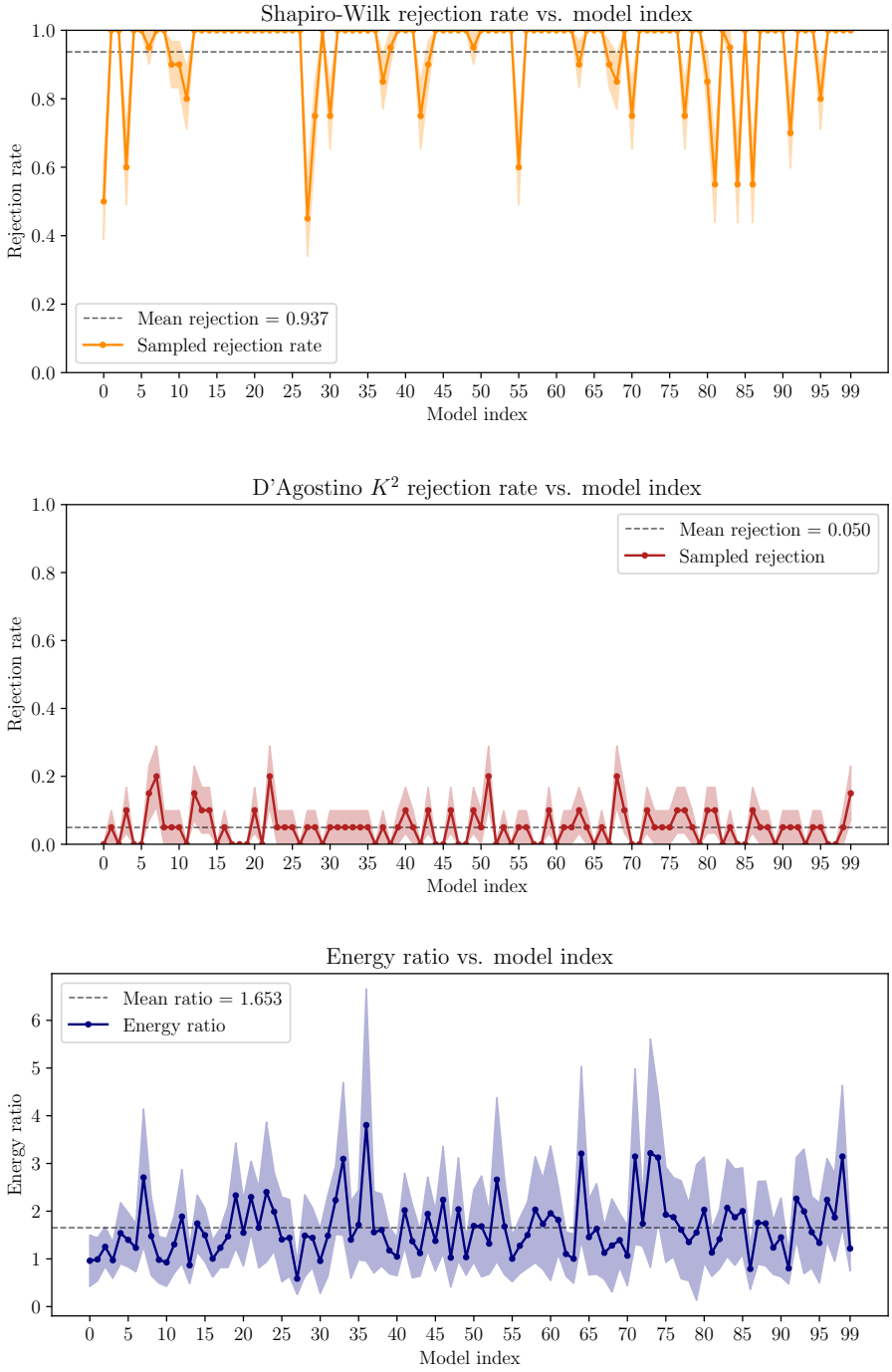


Figure 4.10: CK-test results for untrained univariate ITO models. Each untrained model was tested 20 times on the lag pair $(100, 100)$.

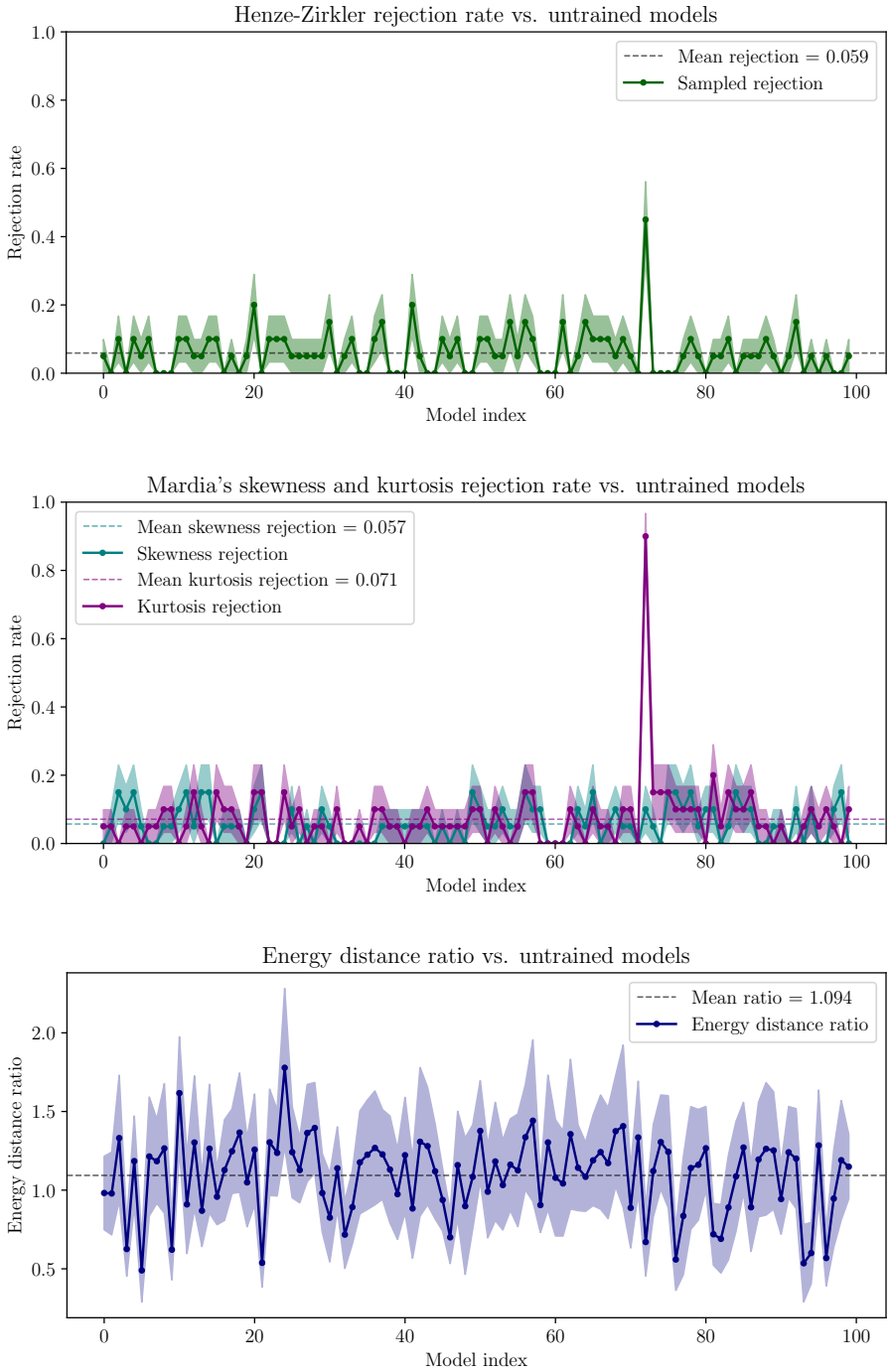


Figure 4.11: CK-test results for untrained multivariate TITO models. Each untrained model was tested 20 times on the lag pair $(500, 500)$.

4.6.3 Evaluation of the vector fields

To compare the typical magnitude of the transport by the vector fields, the RMS value was evaluated for both trained and untrained ITO and TITO models, with results shown in Table 4.1. For both ITO and TITO, the trained model has a larger mean RMS value than the corresponding untrained model, indicating that training increases the average magnitude of the vector field. The ratios between the trained and untrained models further show that the standard deviation of the RMS values increases significantly after training. The significantly smaller standard deviation of the untrained vector fields indicates a more uniform transport when compared to the trained vector fields.

Table 4.1: Vector field RMS statistics for trained and untrained ITO and TITO models. The ratio is computed as trained divided by untrained.

Model	mean RMS	max RMS	std RMS
Untrained ITO	0.2721	0.2785	0.0035
Trained ITO	1.2316	1.4477	0.1950
ITO ratio	4.53	5.20	55.71
Untrained TITO	2.0064	2.0977	0.0438
Trained TITO	3.9000	10.3447	2.5441
TITO ratio	1.94	4.93	58.08

4.6.4 Evaluation of univariate models during training

A randomly initialized model was trained and evaluated using the CK-test. This was done 10 times every epoch and each time the test was run 20 times. The model was trained for 100 epochs and the results from the first 40 epochs can be seen in Figure 4.12. The results show how the model starts performing worse on both the D'Agostino K^2 test and the energy distance ratio when training begins compared to the untrained results, but then the performance improves as the training continues. For the Shapiro–Wilk test, this initial decline in performance doesn't occur and the rejection rate continually improves as the training goes on. Both the D'Agostino K^2 test and the Shapiro–Wilk test seems to converge to a stable rejection rate after approximately 10 epochs, while the energy distance ratio already converges after just a single epoch.

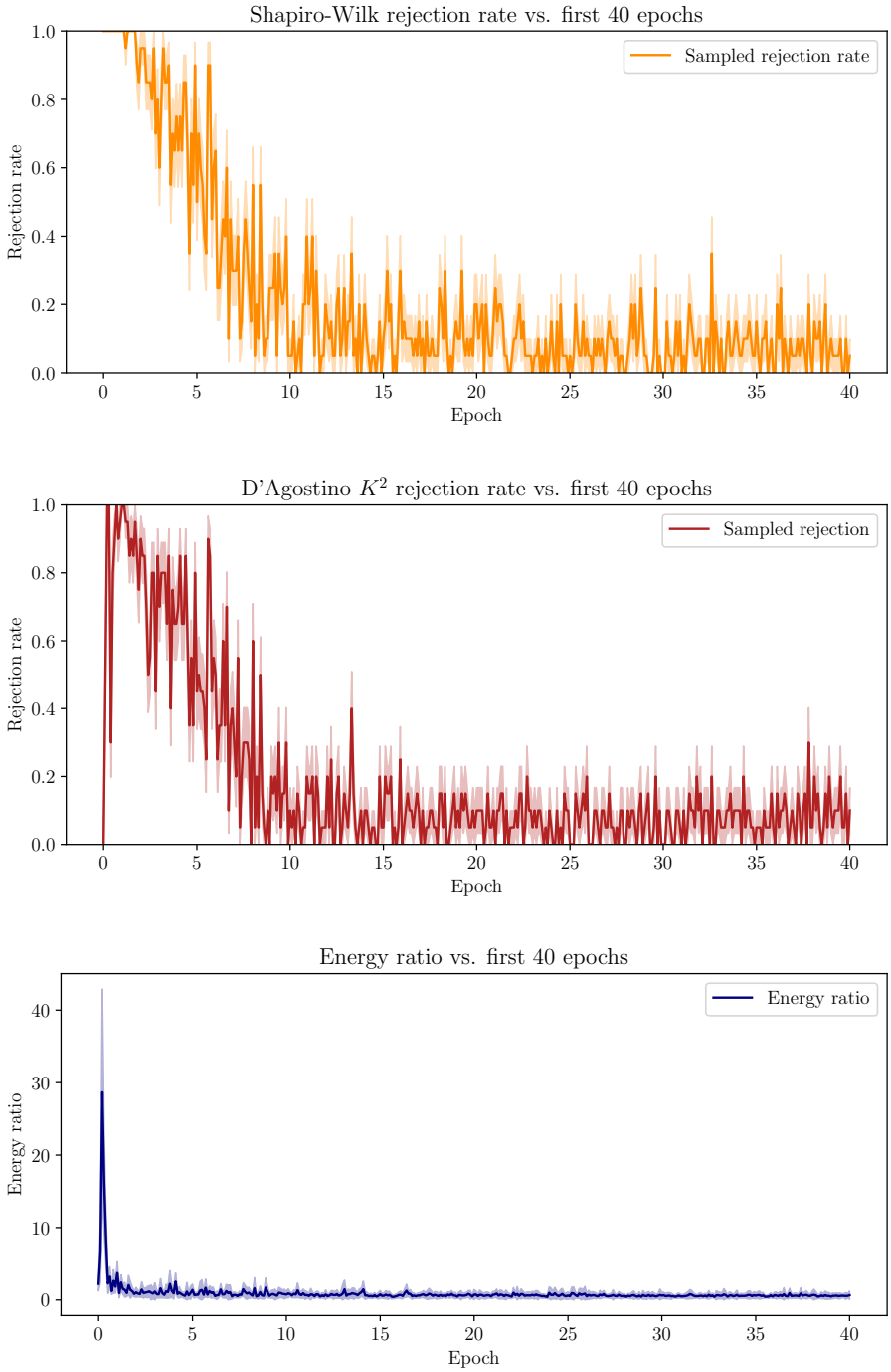


Figure 4.12: CK-test results for the lag pair $(100, 100)$ during the first 40 epochs of training a univariate ITO model. The CK-test was run 20 times at each evaluation point.

5

Conclusion

5.1 Discussion

This section will discuss the findings from applying the developed CK-test for both the univariate and multivariate case, as well as the results from when the test was applied on untrained models.

5.1.1 Evaluation of the univariate CK-test

The results shown in Section 4.2 indicate that the one-dimensional ITO model passes the CK-test since it performs well for all three different metrics, and thus, most likely, follows the semigroup property. This also indicates that the model is self consistent since the test requires both forward and backward sampling to be consistent in order to pass it. The test does, however, not give any clear indication of how well a model is trained or how well it can perform the intended purpose of the model. For the ITO model, the objective is to learn the dynamics of a one-dimensional potential. However, two models with substantially different dynamical performance can still achieve similar results on the CK-test; an example of this can be found in Appendix A.4.

Furthermore, the one-dimensional CK-test produces results that are interpretable and the choice of metrics is highly flexible, since any normality or distribution test can be used. One important aspect of the test is, however, that the current CK-test does not expect element-wise reconstruction, only distributional reconstruction. This is because, as illustrated in Figure 3.3, the sampling from x_1 to x_2 introduces fresh Gaussian noise z_0^2, which makes it improbable to perfectly reconstruct z_0^1. Consequently, metrics that compare the reconstructed and base noise element-wise, such as the mean squared error, should be used with care and can be hard to interpret. A low value of such a metric may indicate that the reverse sampling happens to reconstruct samples close to the original noise, but this is stronger than what is required by the semigroup property. Conversely, a high element-wise error does not necessarily imply that the CK test has failed, as long as the reconstructed noise follows the same distribution as the base noise.

5.1.2 Evaluation of the multivariate CK-test

When the multivariate CK-test is applied to the TITO model, both the Henze–Zirkler test and Mardia's test show rejection rates above 0.9. This is observed for both the total lag experiment and the lag imbalance experiment, as shown in Figures 4.4 and 4.5. The mean energy ratio is also high for both experiments, further indicating that the TITO model does not reconstruct a distribution consistent with normality under this test.

In order to evaluate whether there was an issue with the reverse sampling, the experiment of sampling forward a single step, followed by reversed sampling was conducted. The results presented in Figure 4.6 shows that the model fails to accurately reconstruct a Gaussian distribution. While the rejection rate of Mardia's skewness test is very low, the remaining metrics show a complete rejection, which would indicate that there is an issue with the reverse sampling. However, when evaluating the reverse sampling across a different number of ODE steps, as seen in Figure 4.7, there is a clear increase in performance as the number of steps is increased. This would indicate that the reverse sampling is not inaccurate, but rather sensitive to numerical integration errors, which can occur as an effect of using too few ODE steps in Euler's Method [17]. Although the reverse sampling improves with a higher number of ODE steps, the CK-test was found to be unaffected by the increased number of steps. The complete results from the experiment of applying the CK-test while varying the number of ODE steps can be found in Appendix A.3.2. Since it shows no improvement, the issue of the TITO model failing the test does not seem to be due to improper reverse sampling.

The TITO model has previously been shown to be able to satisfy the semigroup property [7], which raises the question of why the developed CK-test rejects the TITO model. The alternate implementation of the CK-test was implemented in hopes of gaining further insight as to what the cause could be. The results from this test presented in Section 4.5 shows that the TITO model continues to fail the CK-test, albeit with a slightly better energy distance ratio than the original method.

5.1.3 Possible explanations for the TITO results

The underlying cause as to why the TITO model consistently fails both the original CK-test and the alternate CK-test could be due to several reasons. The simplest explanation is that the tested TITO model does not satisfy the semigroup property under the evaluated conditions, but the conclusions presented in the paper that the TITO model originates from would suggest that the model should be able to satisfy the semigroup property [7]. This means that the alternate CK-test is expected to be able to reproduce the dynamics seen in the TITO paper to a reasonable extent, however since the model fails the test, this does not seem to be the case. One factor that could contribute to this contradiction is the fitting when implementing the TICA projection. The lag time used in order to determine slow processes was chosen somewhat arbitrarily based on an analysis of the implied timescales of the MD trajectory, but might benefit from further investigation. Fine-tuning the TICA fit may affect the results of the CK-test. Alternatively, using features other than

torsion angles might also affect the results. Overall, it may be important to further explore the implementation of the alternate CK-test before making any definitive statements regarding the implications of its performance.

Furthermore, the original CK-test that was developed in this paper evaluates the model's ability to achieve a full latent space reconstruction, whereas the conclusion in the TITO paper is drawn from evaluating the slower dynamics in molecular space [7]. As a result, small perturbations of the system in the molecular space, that don't significantly affect the slower dynamics, might have a larger impact after being propagated through the reverse sampling and into the latent space. Therefore, the developed CK-test is a more strict and noise-sensitive test, which might explain why the TITO model fails the test. This may also indicate that the developed test is too strict for it's purpose, especially if the TITO model is mainly used for modeling slower dynamics. It may therefore be useful to evaluate the CK-test on sub-spaces, in order to evaluate the model's slower dynamics, before drawing any firm conclusions on whether the model satisfies the semigroup property.

5.1.4 The CK-test on untrained models

Both Figures 4.10 and 4.11 show that several of the untrained models manage to get good results for most of the CK-test metrics, which may seem quite unusual. We believe that a major reason that untrained models can have a good performance on the CK-test, while being terrible at predicting the dynamics, is that their vector fields have a much smaller typical magnitude than those of trained models. In other words, the untrained models do not significantly transport the base noise, but instead act approximately like identity maps. Such models can trivially satisfy the semigroup property, since applying the identity transition twice is equivalent to applying it once over the combined time interval, which is shown as an equation in Appendix A.5.

The results from the vector field magnitude experiment, shown in Section 4.6.3, further support this interpretation. The untrained models have smaller average vector field magnitudes than the trained models, indicating that they move the samples less during sampling. In addition, the standard deviation of the vector field magnitude is much smaller for the untrained models, suggesting that their transport is more uniform.

This also further supports the claim that satisfying the semigroup property and passing the CK-test does not have a strong correlation to the actual performance of the model, and should therefore not be used as an indication of how "good" a model is at its intended purpose.

5.1.4.1 The ITO model during training

As shown in Figure 4.12, the ITO model initially shows a decrease in CK-test performance during very early training compared to its untrained state. However, after only a few epochs the model starts to improve and achieves better performance compared to the untrained model. This indicates that the ITO model can learn to

satisfy the semigroup property as it learns the underlying dynamics. Furthermore, if the untrained model's good performance is explained by them acting approximately like an identity map, then it is reasonable that CK-test performance decreases during the initial training stage. During early training, the vector field of the model begins to transport samples away from the base distribution, but has not yet learned sufficiently accurate and self-consistent dynamics.

5.2 Summary

From the initial research questions of this project, this thesis developed and evaluated a latent-space Chapman–Kolmogorov test and investigated its ability to assess the physical consistency of ITO models. First, a univariate test was developed and evaluated on a one-dimensional model trained on dynamics from a Prinz potential. This showed that the CK-test was applicable to the one-dimensional ITO model and indicated that the model satisfied the semigroup property. The one-dimensional test was then also used to show that the semigroup property can be learned alongside the underlying dynamics, rather than a property that is inherent to the model architecture.

Expanding on the one-dimensional CK-test, a multivariate version of the test was developed and applied to a three-dimensional TITO model. Evaluating the model in latent space allowed the use of multivariate statistical metrics that would not be directly applicable in molecular space and provided a framework for evaluating CK consistency in higher-dimensional systems. While the test could be implemented and applied to the TITO model, the results were less conclusive than in the one-dimensional setting. This highlighted both the challenges of evaluating the semigroup property in more complex molecular systems and limitations of the proposed test that warrant further investigation.

Overall, this thesis demonstrates the possibility of implementing a Markovianity test on ITO models using the CK equation. Applying the test across different scales and problems is possible, but may require different metrics and careful interpretation. The results also indicate that the semigroup property can be learned during training, but that CK-test performance alone does not fully describe how accurately the model has learned the target dynamics. Therefore, the test should be viewed as a tool for assessing consistency rather than as a complete measure of model quality.

5.2.1 Future work

While the overall goal of the project was achieved and the corresponding research questions answered, there exist areas that would benefit from further exploration.

The multivariate CK-test may benefit from being evaluated on a variety of molecules instead of just using one test molecule. This could supply further insight into the reliability and robustness of the test and show a more general view of the model's performance across different molecular systems. Additionally, the test could be evaluated for larger peptides and proteins in order to determine if it remains applicable at those scales.

It could also prove insightful to evaluate the CK-test across several different models, both univariate and multivariate. Similarly to using more molecules, applying the test to more models could provide a greater confidence in the results and uncover any possible limitations that the test may have.

Lastly, further investigation into the implementation of the TICA projection could be valuable. A more systematic study of the parameters used when fitting the TICA may increase the confidence in the CK-test results. Furthermore, exploring whether projections onto slow dynamical modes could somehow be incorporated into the original CK-test may improve both the performance and reliability of the test.

Bibliography

[1] M. E. Tuckerman, *Statistical Mechanics: Theory and Molecular Simulation*. Oxford, United Kingdom: Oxford University Press, 2010.

[2] K. Bonneau et al., "Breaking the barriers of molecular dynamics with deep-learning: Opportunities, pitfalls, and how to navigate them," *WIREs Computational Molecular Science*, vol. 16, no. 1, 2026, e70064. DOI: https://doi.org/10.1002/wcms.70064.

[3] S. Olsson, "Generative molecular dynamics," *Current Opinion in Structural Biology*, vol. 96, p. 103 213, Feb. 2026, ISSN: 0959-440X. DOI: 10.1016/j.sbi.2025.103213. [Online]. Available: http://dx.doi.org/10.1016/j.sbi.2025.103213.

[4] T. J. Sargent and J. Stachurski, *Continuous-Time Markov Chains*. 2020, Lecture notes. [Online]. Available: https://continuous-time-mcs.quantecon.org/.

[5] R. M. Blumenthal and R. K. Getoor, *Markov Processes and Potential Theory*. New York: Academic Press, 1968.

[6] M. Schreiner, O. Winther, and S. Olsson, "Implicit transfer operator learning: Multiple time-resolution models for molecular dynamics," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36, Curran Associates, Inc., 2023, pp. 36 449–36 462. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/7274ed909a312d4d869cc328ad1c5f04-Paper-Conference.pdf.

[7] J. V. Diez, M. Schreiner, and S. Olsson, "Transferable generative models bridge femtosecond to nanosecond time-step molecular dynamics," *Science Advances*, vol. 12, no. 15, eaed2333, 2026. DOI: 10.1126/sciadv.aed2333.

[8] P. Antoniadis, B. Pavesi, S. Olsson, and O. Winther, *Protein language model embeddings improve generalization of implicit transfer operators*, 2026. DOI: 10.48550/ARXIV.2602.11216. [Online]. Available: https://arxiv.org/abs/2602.11216.

[9] J. V. Diez, M. J. Schreiner, O. Engkvist, and S. Olsson, "Boltzmann priors for implicit transfer operators," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=pRCOZllZdT.

[10] P. Langevin, "Sur la théorie du mouvement brownien," *Comptes Rendus de l'Académie des Sciences (Paris)*, vol. 146, pp. 530–533, 1908.

[11] F. Noé and C. Clementi, "Kinetic distance and kinetic maps from molecular dynamics simulation," *Journal of Chemical Theory and Computation*, vol. 11, no. 10, pp. 5002–5011, 2015. DOI: 10.1021/acs.jctc.5b00553.

[12] S. Klus et al., "Data-driven model reduction and transfer operator approximation," *Journal of Nonlinear Science*, vol. 28, no. 3, pp. 985–1010, 2018. DOI: 10.1007/s00332-017-9437-7.

[13] L. Banh and G. Strobel, "Generative artificial intelligence," *Electronic Markets*, vol. 33, no. 1, p. 63, 2023, ISSN: 1422-8890. DOI: 10.1007/s12525-023-00680-1.

[14] Z. Zhang, J. Zhang, X. Zhang, and W. Mai, "A comprehensive overview of generative ai (gai): Technologies, applications, and challenges," *Neurocomputing*, vol. 632, p. 129 645, 2025, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2025.129645.

[15] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 3964–3979, Nov. 2021, ISSN: 1939-3539. DOI: 10.1109/tpami.2020.2992934.

[16] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, *Normalizing flows for probabilistic modeling and inference*, 2021. arXiv: 1912.02762 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1912.02762.

[17] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," *CoRR*, vol. abs/1806.07366, 2018. [Online]. Available: http://arxiv.org/abs/1806.07366.

[18] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, *Flow matching for generative modeling*, 2023. arXiv: 2210.02747 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2210.02747.

[19] A. Tong et al., "Improving and generalizing flow-based generative models with minibatch optimal transport," *Transactions on Machine Learning Research*, 2024, Expert Certification, ISSN: 2835-8856. [Online]. Available: https://openreview.net/forum?id=CD9Snc73AW.

[20] M. S. Albergo and E. Vanden-Eijnden, "Building normalizing flows with stochastic interpolants," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=li7qeBbCR1t.

[21] X. Liu, C. Gong, and qiang liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=XVjTT1nw5z.

[22] N. Hangst, T. M. Wendt, and S. J. Rupitsch, "Heuristic methods for checking the normality of measurement data with graphical and numerical tests," in *2025 IEEE Sensors Applications Symposium (SAS)*, 2025, pp. 1–6. DOI: 10.1109/SAS65169.2025.11105185.

[23] D. S. Moore, G. P. McCabe, and B. A. Craig, *Introduction to the Practice of Statistics*, 8th ed. New York: W. H. Freeman and Company, 2014, ISBN: 978-1-4641-5893-3.

[24] M. L. Rizzo and G. J. Székely, "Energy distance," *WIREs Computational Statistics*, vol. 8, no. 1, pp. 27–38, 2016. DOI: https://doi.org/10.1002/wics.1375.

[25] B. Ebner and N. Henze, "Tests for multivariate normality—a critical review with emphasis on weighted L^2-statistics," *TEST*, vol. 29, no. 4, pp. 845–892, 2020, ISSN: 1863-8260. DOI: 10.1007/s11749-020-00740-0.

[26] H. C. Thode, *Testing for Normality* (Statistics: Textbooks and Monographs). CRC Press, 2002.

[27] N. M. Razali and Y. B. Wah, "Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests," *Journal of Statistical Modeling and Analytics*, 2011. [Online]. Available: https://www.nrc.gov/docs/ML1714/ML17143A100.pdf.

[28] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3-4, pp. 591–611, Dec. 1965, ISSN: 0006-3444. DOI: 10.1093/biomet/52.3-4.591.

[29] P. H. Westfall, "Kurtosis as peakedness, 19052014. r.i.p.," *The American Statistician*, vol. 68, no. 3, pp. 191–195, 2014, PMID: 25678714. DOI: 10.1080/00031305.2014.917055.

[30] R. D'Agostino and E. S. Pearson, "Tests for departure from normality. empirical results for the distributions of b2 and b1," *Biometrika*, vol. 60, no. 3, pp. 613–622, 1973, ISSN: 00063444, 14643510. Accessed: Mar. 26, 2026. [Online]. Available: http://www.jstor.org/stable/2335012.

[31] F. J. Anscombe and W. J. Glynn, "Distribution of the kurtosis statistic b2 for normal samples," *Biometrika*, vol. 70, no. 1, pp. 227–234, 1983, ISSN: 00063444, 14643510. Accessed: Mar. 28, 2026. [Online]. Available: http://www.jstor.org/stable/2335960.

[32] A. C. Rencher and W. F. Christensen, *Methods of Multivariate Analysis*, 3rd ed. John Wiley & Sons, 2012. [Online]. Available: https://ebookcentral.proquest.com/lib/chalmers/detail.action?docID=875890.

[33] N. Henze, "Invariant tests for multivariate normality: A critical review," *Statistical Papers*, vol. 43, no. 4, pp. 467–506, 2002, ISSN: 1613-9798. DOI: 10.1007/s00362-002-0119-6.

[34] N. Henze and B. Zirkler, "A class of invariant consistent tests for multivariate normality," *Communications in Statistics - Theory and Methods*, vol. 19, no. 10, pp. 3595–3617, 1990. DOI: 10.1080/03610929008830400.

[35] K. V. Mardia, "Measures of multivariate skewness and kurtosis with applications," *Biometrika*, vol. 57, no. 3, pp. 519–530, 1970, ISSN: 00063444, 14643510. Accessed: May 13, 2026.

[36] J.-H. Prinz et al., "Markov models of molecular kinetics: Generation and validation," *The Journal of Chemical Physics*, vol. 134, no. 17, p. 174 105, May 2011. DOI: 10.1063/1.3565032.

[37] M. Hoffmann et al., "Deeptime: A python library for machine learning dynamical models from time series data," *Machine Learning: Science and Technology*, vol. 3, no. 1, p. 015 009, 2021. DOI: 10.1088/2632-2153/ac3de0.

[38] J. Viguera Diez and S. Olsson, *Mdqm9-nc dataset*, Zenodo, Feb. 2024. DOI: 10.26434/chemrxiv-2023-sx61w.

[39] J. Viguera Diez, S. Romeo Atance, O. Engkvist, and S. Olsson, "Generation of conformational ensembles of small molecules via surrogate model-assisted molecular dynamics," *Machine Learning: Science and Technology*, vol. 5, no. 2, p. 025 010, Apr. 2024, ISSN: 2632-2153. DOI: 10.1088/2632-2153/ad3b64. [Online]. Available: http://dx.doi.org/10.1088/2632-2153/ad3b64.

[40] L. Molgedey and H. G. Schuster, "Separation of a mixture of independent signals using time delayed correlations," *Phys. Rev. Lett.*, vol. 72, pp. 3634–3637, 23 Jun. 1994. DOI: 10.1103/PhysRevLett.72.3634.

[41] G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé, "Identification of slow molecular order parameters for markov model construction," *The Journal of Chemical Physics*, vol. 139, no. 1, p. 015 102, Jul. 2013, ISSN: 0021-9606. DOI: 10.1063/1.4811489.

[42] M. Tancik et al., *Fourier features let networks learn high frequency functions in low dimensional domains*, 2020. arXiv: 2006.10739 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2006.10739.

A

Appendix 1

A.1 Prinz potential landscape

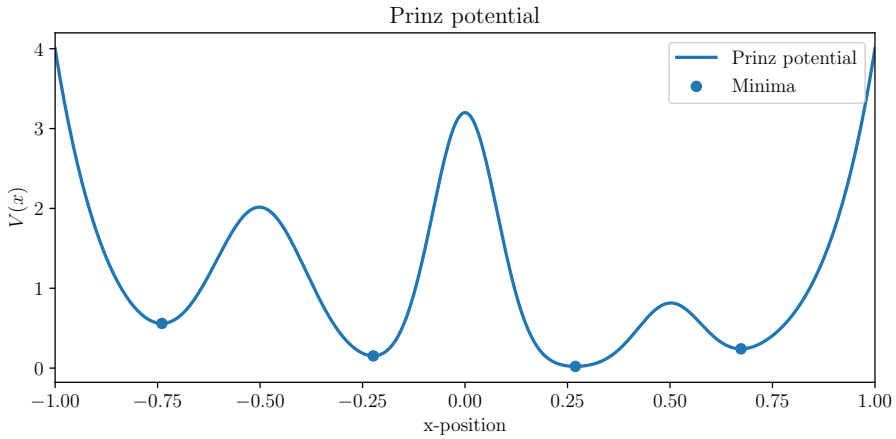


Figure A.1: The one-dimensional Prinz potential used to generate the training trajectories for the ITO model. The blue dots indicate the location of the minima.

The Prinz potential used by the one-dimensional ITO model is shown in Figure A.1. The landscape consists of four metastable states separated by energy barriers of varying height, giving rise to transitions on multiple timescales [36].

A.2 Ultra Vector Field architecture

The vector field used in the CFM model was parameterized by a neural network called the *Ultra Vector Field*. It takes the interpolation time t, the current interpolation state x_t, the conditioning position x, and the lag time τ as input. The learned vector field is written as

$$v_\theta(t, x_t \mid x, \tau), \tag{A.1}$$

where θ denotes the trainable network parameters. The output is a scalar velocity that determines how x_t is transported during the learned flow.

Before being passed to the network, the coordinate inputs x and x_t are encoded using Gaussian Fourier features [42], which for a scalar input z can be expressed as:

$$\gamma(z) = [\sin(2\pi z W), \cos(2\pi z W)] \tag{A.2}$$

where W is a Gaussian random matrix. The lag time is normalized by the maximum lag used during training. These encoded quantities are concatenated and passed through a fully connected neural network with residual blocks, where the interpolation time t is used to scale and shift the hidden representation in each block. Finally, a linear output layer maps the hidden representation to the scalar velocity.

A.3 Additional TITO results using more ODE steps

A.3.1 Results from reverse sampling TITO with 1000 ODE steps

Figure A.2 shows the reverse sampling experiment when using 1000 ODE steps when sampling. The average rejection rate of Mardia's skewness and kurtosis test are both close to the significance level of 0.05. The Henze-Zirkler test shows a slightly higher rejection, but is still close to the significance level. The higher variance is possibly partly due to the fact that the test was only run 5 times per total lag, which means that a single rejection raises the rate to 0.2. The energy distance ratio is lower than the target of 1. This all suggests that the model is able to reverse properly when using 1000 ODE steps.

A.3.2 Results from varying ODE steps in multivariate CK-test

Figure A.3 shows how varying the number of ODE steps when sampling affects the CK-test metrics. Both the Henze-Zirkler test and Mardia's skewness and kurtosis test remain constant regardless of the number of ODE steps and also at a high average rejection rate. The energy distance ratio shows a high variance across the number of ODE steps, but it never manages to achieve a value close to the target of 1. Overall, the multivariate CK-test seem to be unaffected by the increase in ODE steps.

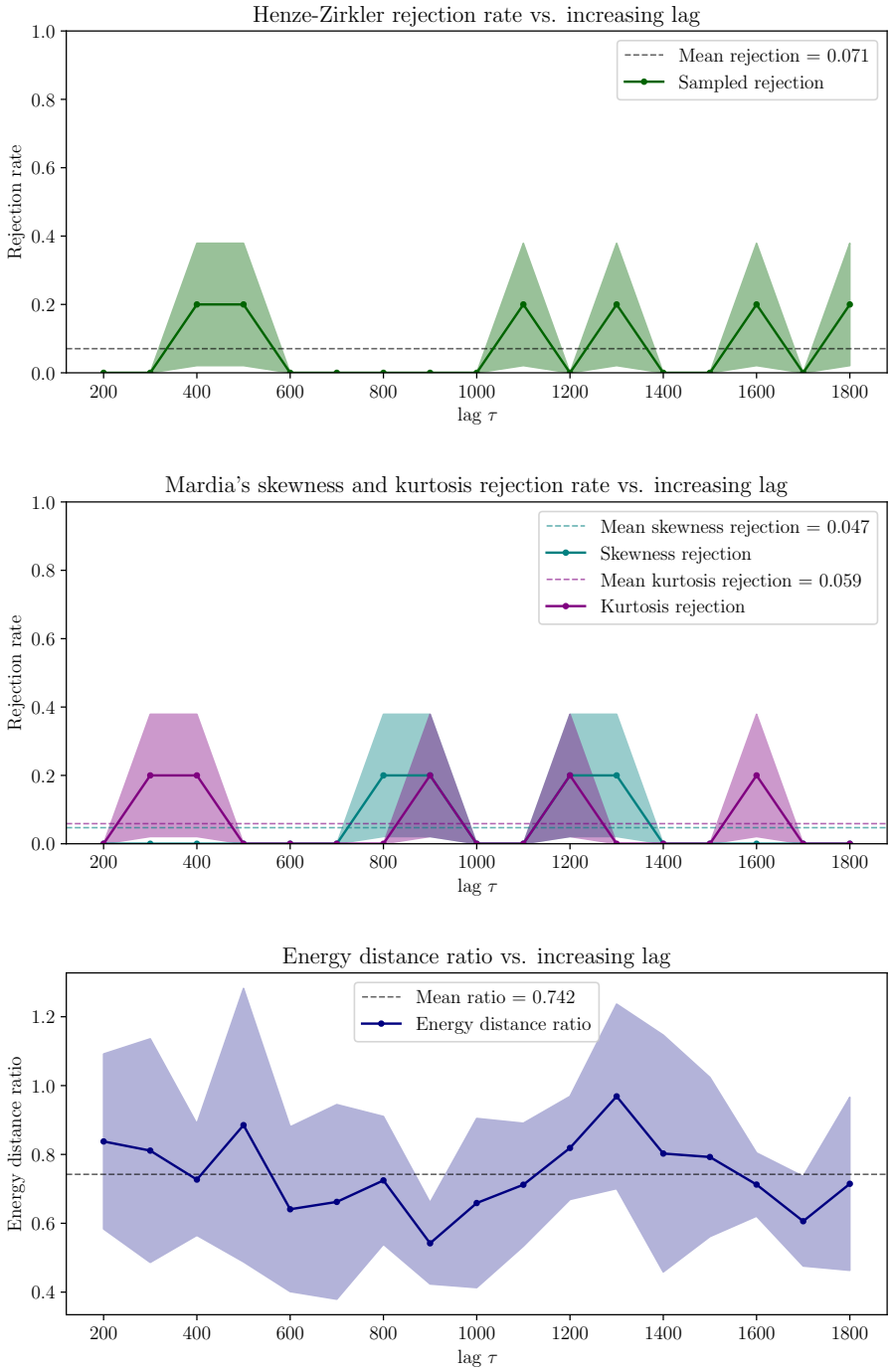


Figure A.2: Results from inverting the multivariate TITO model, simulating a single time step forward and then reversing it. The time lag used was increased from 200 to 1800 in increments of 100 and the test was run 5 times for each total lag.

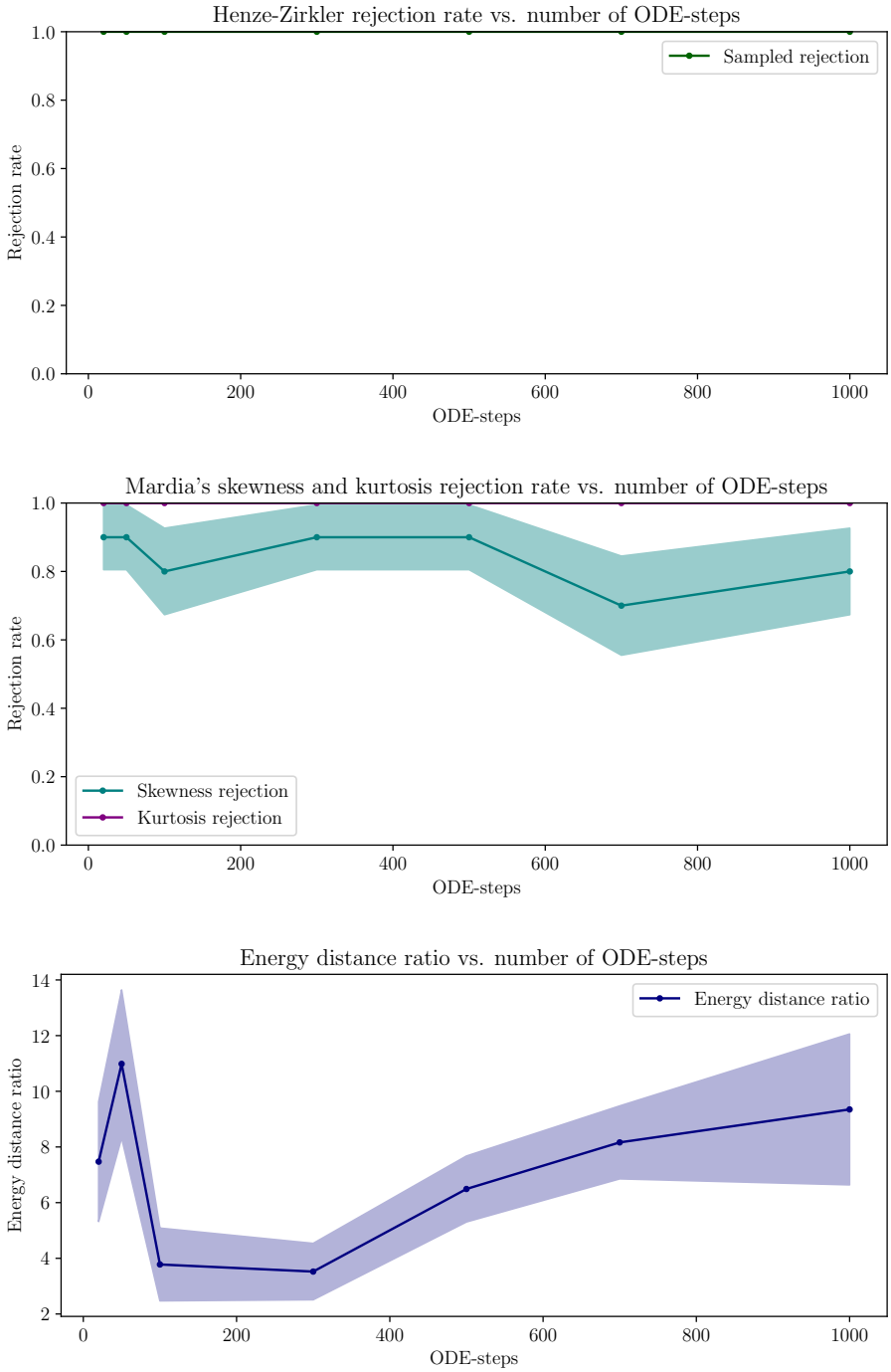


Figure A.3: Results from inverting the multivariate TITO model, simulating a single time step forward and then reversing it. The model was evaluated on the lag pair $(500, 500)$. The number of ODE steps used when simulating was increased from 20 to 1000 and the test was run 10 times for each value.

A.4 Results from moderately trained ITO model

Figure A.5 shows the results from the CK-test applied to an ITO model with the same architecture as the one used in the rest of the thesis, but only trained for 20 epochs instead of 1000. The results are only slightly worse compared to the CK-test performance of the model that is trained for 1000 epochs, but still good enough to be considered to pass the CK-test. However, Figure A.4 shows the sampled dynamics of both models, illustrating that the model trained for 20 epochs has not yet learned the correct dynamics. In particular, it underestimates the two larger wells while overestimating the two smaller ones. This indicates that good CK-test performance does not necessarily imply that the model has learned the underlying dynamics accurately.

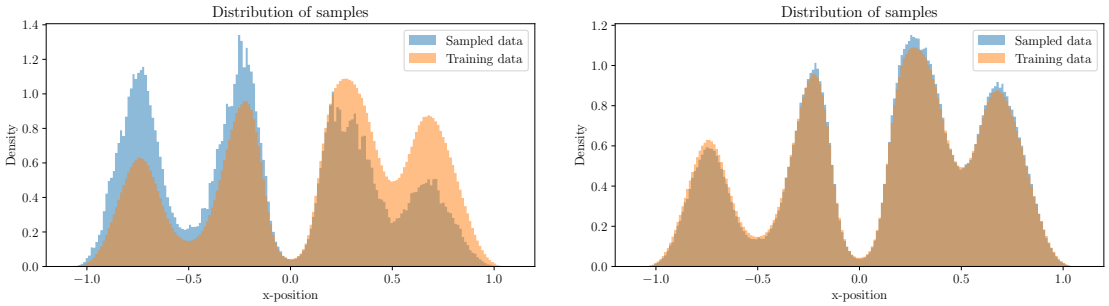


Figure A.4: The sampled distribution after 1000 steps and a batch size of 4096 for two ITO models with the same architecture but different number of training epochs. Left was trained for 20 epochs and right for 1000.

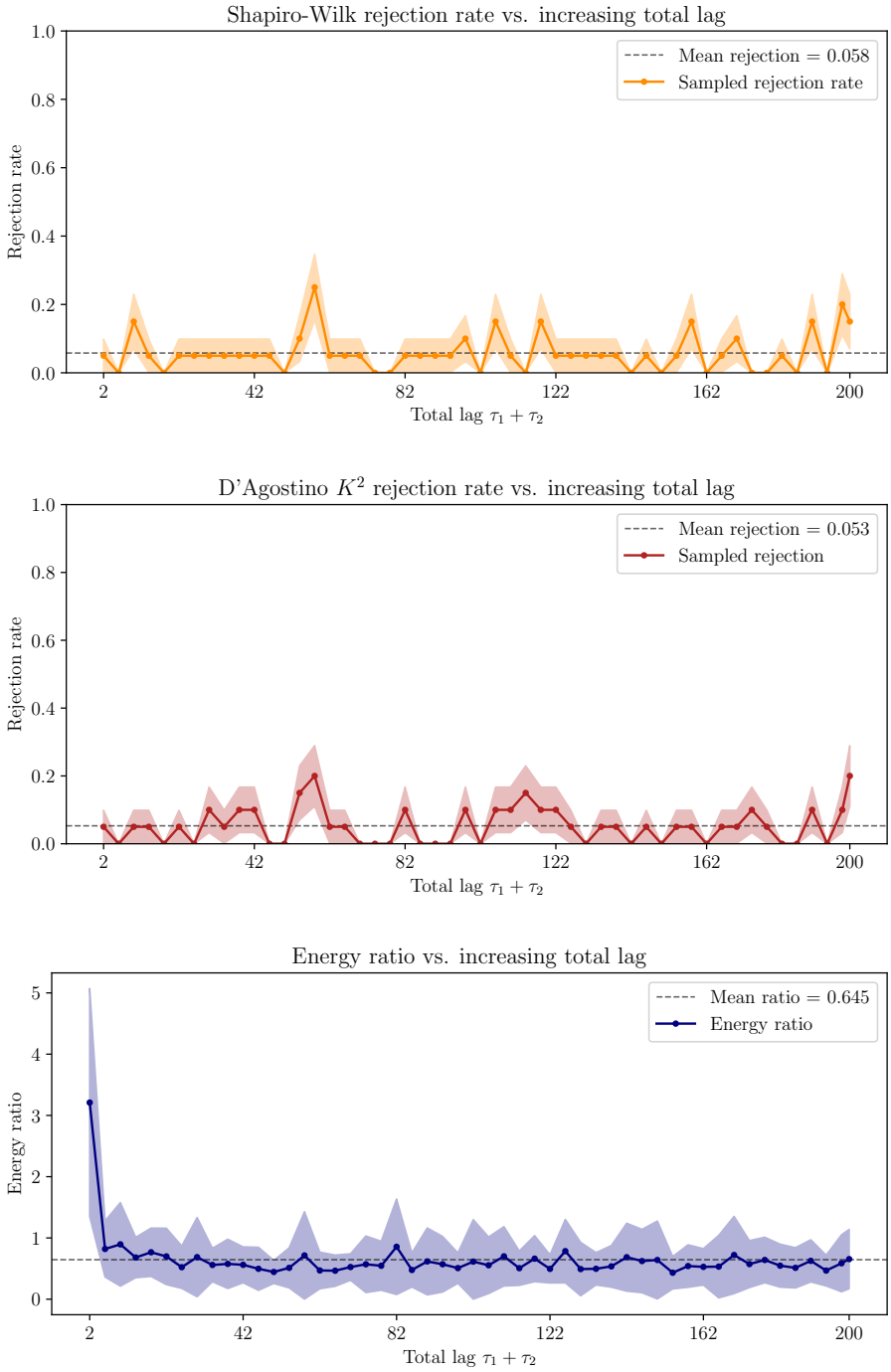


Figure A.5: CK-test results for the univariate ITO-model with varying total lag time for a model trained for 20 epochs. The total lag was increased from 2 to 200 in increments of 4 and the CK-test was run 20 times for each total lag.

A.5 Identity map CK-equation

For an identity map, the transition kernel is given by:

$$P_t(x, A) = \mathbf{1}_A(x) \qquad P_t(x, dy) = \delta_x(dy)$$

where δ_x denotes the Dirac measure at x and where $\mathbf{1}_A(x)$ is the indicator function, equal to one if $x \in A$ and zero otherwise. Substituting this into the Chapman–Kolmogorov equation (2.3) gives:

$$P_{t+s}(x, A) = \mathbf{1}_A(x) = \int \delta_x(dy)\, \mathbf{1}_A(y) = \int P_t(x, dy)\, P_s(y, A) \tag{A.3}$$

which satisfies the equation.