Katharina Röck

Stochastic processes as surrogate models for dynamical systems in magnetic confinement fusion

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Katharina Röck

Stochastic processes as surrogate models for dynamical systems in magnetic confinement fusion

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

eingereicht am 24.01.2024 von Katharina Röck, geb. Rath

Erster Berichterstatter: Prof. Dr. Bernd Bischl
Zweiter Berichterstatter: PD Dr. Udo von Toussaint
Dritter Berichterstatter: Prof. Dr. Ulrich Stroth

Tag der Disputation: 08.05.2024

Acknowledgments

I would like to express my sincere gratitude to ...

... my supervisor Prof. Dr. Bernd Bischl for many enlightening discussions, his trust and support;

... my co-advisor PD Dr. Udo von Toussaint for his unwavering support, guidance, and ability to distill complex matters down to their essence;

... Prof. Dr. Ulrich Stroth for agreeing to act as the third reviewer for my thesis;

... Ass.Prof. Dr. Christopher Albert, who first aroused my interest in physics many years ago and who has been a true companion throughout my whole academic journey. His expertise, dedication, and patience have been invaluable;

... Prof. Dr. David Rügamer for his patience, guidance, and many helpful discussions during my PhD;

... my collaborators at MIT, Cristina Rea, Robert Granetz, and Andrew Maris, for the great collaboration and inspiring discussions.

... my colleagues at IPP and at the Department of Statistics for their support, feedback, and encouragement: Benedikt, Emilio, Frederik, Jann, Mario, Michael, Nils, Philipp, Robert, Roland, Victor;

... my parents for their endless support;

... Friederike, who was not particularly helpful in writing this thesis but who constantly reminds me that the world is filled with wonders;

... Julius, for his love, patience, and everything else during our journey together.

Summary

When designing machine learning (ML) models for scientific applications, a key point is to incorporate *a priori* domain-specific information in the model. Especially, when constructing reduced complexity models as surrogates, we need to ensure that the mathematical and physical properties of the underlying system are reflected correctly by the ML model.

The first part of this thesis focuses on physics-consistent Gaussian processes (GPs) that respect laws of physics by design. This stands in contrast to so-called physics-informed regressors that incorporate physical constraints weakly through the loss function. In scenarios where data originate from underlying linear partial differential equations (PDEs) with localized sources, the proposed model is a superposition of a Gaussian process with a specialized kernel that is constructed to exactly fulfill the homogeneous part of the PDE while a linear model is used for sources. The specialized kernel ensures an exact correspondence and physical interpretability of hyperparameters allowing insights into the underlying physical characteristics.

Physics-consistent GPs are then extended to model mappings in the phase space of Hamiltonian systems. Here, we propose a surrogate model based on multi-output GPs deploying derivative information with a matrix-valued covariance function to fully preserve the symplecticity of the Hamiltonian flow and thus conserve integrals of motion. The proposed method is related to geometric integration methods, but models the flow map with larger time steps, accelerating long-term computations. In chaotic systems, the symplectic surrogate model can not only be used for faster computations but also for early classification of chaotic versus regular trajectories, based on the calculation of Lyapunov exponents directly available from the surrogate model.

One particular challenge in applying ML models to problems in plasma physics is the lack of labeled data for training larger models. Usually, physical experiments are extremely expensive and with regard to future fusion reactors, sufficient data will not be available until operations start. The second part of this thesis treats data augmentation via robust surrogate models of multivariate time series data to mitigate this problem. We apply Student-t process regression in a state space formulation to ensure reliable uncertainty estimates despite outliers. This reduces computational complexity and allows us to use the model for high-resolution time series. We are using different approaches in this regard. One approach assumes uncorrelated input signals and induces correlations and cross-correlations via coloring transformations in a post-processing step. Another technique immediately incorporates correlations by using a multivariate Matérn kernel. Both approaches are found to be well-suited for data imputation and augmentation for multichannel time series sensor data with outliers.

Zusammenfassung

Bei der Entwicklung von Modellen, die Methoden des maschinellen Lernens (ML) für naturwissenschaftliche Anwendungen verwenden, ist es wichtig, fachspezifische Informationen und Gesetzmäßigkeiten in das Modell einzubeziehen. Insbesondere bei der Konstruktion von Surrogatmodellen mit reduzierter Komplexität muss sichergestellt werden, dass die mathematischen und physikalischen Eigenschaften des zugrundeliegenden Systems durch das ML-Modell korrekt wiedergegeben werden.

Der erste Teil der Dissertation befasst sich mit physikalisch konsistenten Gauß-Prozessen, die so konstruiert sind, dass sie physikalische Gesetze berücksichtigen. Dieser Ansatz steht im Gegensatz zu physikalisch informierten Regressoren, die physikalische Randbedingungen über die Verlustfunktion einbeziehen.

Für Daten aus zugrundeliegenden linearen partiellen Differentialgleichungen mit lokalisierten Quellen wird ein Modell vorgeschlagen, das auf einer Superposition eines Gauß-Prozesses mit spezialisierter Kovarianzfunktion, die den homogenen Teil der Differentialgleichung exakt erfüllt, und einem linearen Modell für die lokalisierten Quellen basiert. Durch die spezialisierte Kovarianzfunktion werden exakte Konsistenz und physikalische Interpretierbarkeit der Hyperparameter gewährleistet. Hierdurch werden auch Einblicke in die zugrundeliegenden physikalischen Charakteristika ermöglicht.

Hierauf aufbauend werden physikalisch konsistente Gauß-Prozesse entwickelt, um Abbildungen im Phasenraum von Hamiltonischen Systemen zu modellieren. Dafür wird ein Surrogatmodell vorgeschlagen, das auf multivariaten Gauß-Prozessen basiert und Gradienteninformationen mittels einer matrixwertigen Kovarianzfunktion verwendet, um die Symplektizität des Hamiltonschen Flusses exakt zu erhalten. Dadurch werden auch die Integrale der Bewegung erhalten. Das vorgeschlagene Modell erlaubt es, Langzeitberechnungen durch die Abbildung des Flusses mit großen Schrittweiten zu beschleunigen. Für chaotische Systeme kann das symplektische Surrogatmodell nicht nur für schnellere Berechnungen verwendet werden, sondern auf Basis der Berechnung der Lyapunovexponenten auch für eine frühzeitige Klassifizierung von chaotischen und regulären Trajektorien.

Eine besondere Herausforderung bei der Anwendung von ML-Modellen auf Probleme der Plasmaphysik ist die geringe Menge an gelabelten Daten, um größere Modelle zu trainieren. Physikalische Experimente sind in der Regel sehr teuer und im Hinblick auf zukünftige Fusionsreaktoren werden bis zu deren Inbetriebnahme nicht genügend gelabelte Daten zur Verfügung stehen. Daher beschäftigt sich der zweite Teil der Arbeit mit der Datenerweiterung mittels robuster Surrogatmodelle für multivariate Zeitreihen. Wir verwenden Student-t-Prozesse in einer Zustandsraumformulierung, um einerseits zuverlässige Unsicherheitsvorhersagen trotz Ausreißern zu gewährleisten und andererseits die Rechenkomplexität durch das Zustandsraummodell zu reduzieren. Damit ist das vorgeschlagene Modell auch für hochaufgelöste Zeitreihen anwendbar. Es werden zwei Ansätze vorgeschlagen: Ein Ansatz geht von unkorrelierten Eingangssignalen aus und induziert Korrelationen und Kreuzkorrelationen im Nachgang mittels Coloring-Transformationen. Das andere Modell induziert Korrelationen direkt durch eine multivariate Matérn Kovarianzfunktion. Beide Modelle eignen sich gut zur Datenerweiterung bzw. Datenimputation für multivariate Zeitreihen mit Ausreißern.

Contents

1 Overview

Motivation Modeling complex real-world phenomena requires physically consistent algorithms that ensure that predictions align with the fundamental laws of nature. Even though a purely data-driven model may fit observations well, it might lead to physically inconsistent and even wrong predictions if no physical knowledge is included. Hence, incorporating laws of physics in machine learning (ML) models to ensure physically plausible predictions has been an active area of research (e.g., Jin et al., 2020; Peng and Mohseni, 2016; Raissi et al., 2017; Swiler et al., 2020). Different strategies can be employed to improve ML algorithms with *a priori* information (Karniadakis et al., 2021):

(i) utilizing prior knowledge derived from observations (e.g., Fischer et al., 2003);

(ii) incorporating inductive biases via specialized alterations of the ML model architecture that ensure that predictions satisfy given mathematical or physical constraints (e.g., Jin et al., 2020; Burby et al., 2020; Greydanus et al., 2019);

(iii) by an appropriate choice of a physics-informed loss function that favors predictions corresponding to the underlying physics (e.g., Raissi et al., 2017).

Stochastic processes like Gaussian and Student-t processes (GPs and TPs) offer great flexibility in modeling complex physical systems. The probabilistic approach allows for uncertainty quantification and prior information can be incorporated through the kernel function that encodes known relationships in the data. They also allow a straightforward specification of hyperparameters as they facilitate the definition of quantities like correlation length and smoothness.

There are numerous application fields in theoretical and experimental physics of physics-consistent stochastic processes. In this thesis, we focus on problems in magnetic confinement fusion. Fusion is a highly complex process that releases energy from merging atomic nuclei. Simulating and understanding the behavior of plasma in fusion reactors, such as tokamaks and stellarators, requires accurate and physically consistent models.
In this realm, the accelerated computation of alpha particles, which are a product of the fusion reaction, is a current topic of interest in optimizing reactor designs. Alpha particles should be confined in a fusion reactor as they heat the plasma through collisions. When using fast surrogate models to estimate the loss of alpha particles, it is important that they reflect the underlying laws of physics.
Another area of application is the prediction of rapidly growing instabilities, so-called disruptions, which lead to a sudden loss of thermal and magnetic energy and are potentially harmful for the fusion reactor. To maintain a safe and stable operation, it is crucial to predict disruptions in order to start disruption mitigation measures. Progress has been achieved in disruption prediction using ML models (Rea and Granetz, 2018; Rea et al., 2019, 2020; Pau et al., 2019; Berkery et al., 2017). However, labeled experimental data is limited and expensive to obtain. Also with regard to future fusion reactor devices such as ITER or SPARC, insufficient data will be available to have a fully trained model for predicting disruptions. A carefully designed data augmentation model could

help to improve the imbalanced and restricted data situation by creating rare disruption events and thereby robustify the prediction performance of ML models.

This thesis consists of five contributing articles that focus on two strategies for incorporating physical knowledge into ML models for advancing fusion: (1) inductive bias via specialized model architectures and (2) augmenting and imputing data in a way that the correlation structure of multivariate experimental signals is retained. Chapter 2 revisits fundamental physical concepts and stochastic processes to provide a solid foundation for the contributions presented in Chapter 3.

Specialized model architectures The linear nature of GPs allows us to include physics information in the form of partial differential equations (PDEs) within the model. In Section 3.1, we present specialized kernels that exactly fulfill given homogenous PDEs with an additional linear model accounting for potential source contributions. Section 3.2 introduces structure-preserving surrogate models based on GPs. The presented approach can be employed to accelerate the computation of alpha particle losses by serving as a fast emulator for orbit tracers. This model can be combined with early orbit classification, as presented in Section 3.3, to assess whether particles are confined or lost.

Data augmentation and imputation In order to augment the training data set for a disruption predictor with data that reflect the underlying physics, we propose fast local models based on state space Student-t process regression. Section 3.4 introduces an uncorrelated model for different input dimensions and imposes correlations afterward via coloring transformations. In Section 3.5, we consider a fully multivariate model, which directly includes correlations between input dimensions and is thus also capable of reconstructing gappy data with information from other input dimensions.

2 Background

This chapter recalls the fundamentals of Hamiltonian mechanics and symplectic geometry and discusses challenges regarding geometric integration. Then, we revisit Gaussian process regression, explain hyperparameter optimization and the incorporation of derivative observations, and discuss suitable covariance functions. We consider the related Student-t processes and discuss their advantages. Finally, we explain the relation of Gaussian and Student-t processes to stochastic differential equations and introduce Bayesian filters and smoothers for sequentially solving regression problems.

2.1 Hamiltonian systems

Hamiltonian systems are ubiquitous in physics and engineering as they allow us to describe numerous dynamical systems in classical mechanics, electrodynamics, and plasma physics (Goldstein, 1980; Arnold, 1989; Lichtenberg and Lieberman, 1992; José and Saletan, 1998; Lee, 2003). Examples of Hamiltonian systems are the planetary system, the undamped pendulum, or the motion of a charged particle in an electromagnetic field. The Hamiltonian formalism developed by William Rowan Hamilton in 1834 is not only a convenient way of solving dynamical systems but has far-reaching implications. For many systems, the entire information of how the system evolves is encoded in the scalar function H. Geometric properties implied by the Hamiltonian structure led to the development of symplectic geometry. Moreover, the concept of integrability allows insights into stability and the formalism allows to include perturbations efficiently (Arnold, 1989).
In the following, Hamilton's equations of motion are defined, along with an explanation of invariants of motion and an introduction to symplectic geometry. Then, we discuss how symplectic integration schemes preserve invariants of motion. Finally, we will discuss Hamiltonian chaos and ways of distinguishing regular from chaotic trajectories.

Here, we consider an f-dimensional, autonomous system characterized by a time-invariant scalar function, the Hamiltonian $H(\boldsymbol{q}, \boldsymbol{p})$, depending on f generalized position coordinates $\boldsymbol{q}$ and f generalized momenta $\boldsymbol{p}$. The solutions of Hamilton's canonical equations of motion,

$$\dot{\boldsymbol{q}} = \frac{d\boldsymbol{q}(t)}{dt} = \nabla_{\boldsymbol{p}} H(\boldsymbol{q}(t), \boldsymbol{p}(t)) \,, \tag{2.1}$$

$$\dot{\boldsymbol{p}} = \frac{d\boldsymbol{p}(t)}{dt} = -\nabla_{\boldsymbol{q}} H(\boldsymbol{q}(t), \boldsymbol{p}(t)) \,, \tag{2.2}$$

define the system's evolution and are integral curves of the Hamiltonian vector field X_H. They are called trajectories or orbits of the system. Using $\boldsymbol{z} = (\boldsymbol{q}(t), \boldsymbol{p}(t))$, equations 2.1 and 2.2 can be rewritten as

$$X_H(\boldsymbol{z}) = \begin{pmatrix} \nabla_{\boldsymbol{p}} H(\boldsymbol{z}) \\ -\nabla_{\boldsymbol{q}} H(\boldsymbol{z}) \end{pmatrix} = \mathbf{J}^{-1} \nabla_{\boldsymbol{z}} H(\boldsymbol{z}) \,, \tag{2.3}$$

where

$$\mathbf{J}^{-1} = \begin{pmatrix} 0 & \mathbf{I} \\ -\mathbf{I} & 0 \end{pmatrix} \,, \tag{2.4}$$

with $\mathbf{I}$ being the $f \times f$ identity matrix.

2.1.1 Invariants of motion

The Hamiltonian H is constant along integral curves of X_H. More generally, the total time derivative of an arbitrary function $F(\boldsymbol{q}, \boldsymbol{p}, t)$ can be written as,

$$\frac{dF}{dt} = \frac{\partial F}{\partial t} + \sum_i \left(\frac{\partial F}{\partial q_i} \dot{q}_i + \frac{\partial F}{\partial p_i} \dot{p}_i \right) = \frac{\partial F}{\partial t} + \sum_i \left(\frac{\partial F}{\partial q_i} \frac{\partial H}{\partial p_i} - \frac{\partial F}{\partial p_i} \frac{\partial H}{\partial q_i} \right) = \frac{\partial F}{\partial t} + \{F, H\} \,, \tag{2.5}$$

where we used Hamilton's equation and the notation of a Poisson bracket $\{\cdot, \cdot\}$ (Arnold, 1989; Lichtenberg and Lieberman, 1992). If F is not explicitly dependent on t and the Poisson bracket vanishes, F commutes with the Hamiltonian and is an invariant of motion. As we are here only considering autonomous systems with time-independent Hamiltonian, and $\{H, H\} \equiv 0$, the Hamiltonian is an invariant of motion and conserved. In many Hamiltonian systems, the Hamiltonian corresponds to the energy of the system. This means that orbits with a given energy lie on a $(2f-1)$-dimensional constant-energy surface $E = H(\boldsymbol{q}, \boldsymbol{p})$.

The Hamiltonian vector field X_H is divergence-free,

$$\nabla \cdot X_H = \sum_i \left(\frac{\partial \dot{q}_i}{\partial q_i} + \frac{\partial \dot{p}_i}{\partial p_i} \right) = \sum_i \left(\frac{\partial}{\partial q_i} \frac{\partial H}{\partial p_i} - \frac{\partial}{\partial p_i} \frac{\partial H}{\partial q_i} \right) = 0 \,. \tag{2.6}$$

This is called Liouville's theorem and implies that any closed volume of phase space is conserved, although its shape may be deformed (Arnold, 1989). However, the underlying geometric structure of Hamilton's canonical equations implies much more than phase-space volume conservation or incompressibility. Hamiltonian flows are symplectic maps and preserve symplecticity. Mikhael Gromov strengthened Liouville's theorem with the non-squeezing theorem (Tao, 2006), showing that it is impossible to squeeze a ball into a cylinder of smaller radius with an Hamiltonian flow. Gromov illustrated this concept with the symplectic camel: This camel cannot pass through the eye of a needle (Stewart, 1987). This implies that the space of symplectic maps is much more restrictive with restrictions on shape and *symplectic width* than the one of volume-preserving ones.

2.1.2 Symplectic integration and generating functions

When integrating Hamilton's equations of motion (see Eqs. 2.1 and 2.2) to calculate trajectories, it is important to preserve the symplectic structure of phase space and thereby, conserve invariants of motion within fixed bounds (Hairer et al., 2006; Goldstein, 1980). If this is not the case, the numerically integrated Hamiltonian system can become dissipative, and results may show incorrect long-time behavior (Hairer et al., 2006; Abdullaev, 2006). This can be avoided by employing

canonical transformations and thereby restricting the possible maps to symplectic maps (Hairer et al., 2006). A symplectic map $g(\boldsymbol{q}, \boldsymbol{p})$ preserves the Hamiltonian character of the differential equation, and its Jacobian $g'(\boldsymbol{q}, \boldsymbol{p})$ satisfies

$$g'(\boldsymbol{q}, \boldsymbol{p})^\top \mathbf{J} g'(\boldsymbol{q}, \boldsymbol{p}) = \mathbf{J} \,, \tag{2.7}$$

meaning it is symplectic for all $\boldsymbol{q}$ and $\boldsymbol{p}$. This also holds for the Hamiltonian flow of time-invariant systems, and hence, the Hamiltonian flow is a symplectic map for each time step t (Hairer et al., 2006). Another perspective on symplectic maps is that any canonical change of coordinates is a symplectic transformation (Lichtenberg and Lieberman, 1992; Hairer et al., 2006; Goldstein, 1980). A canonical transformation from initial coordinates $(\boldsymbol{q}, \boldsymbol{p})$ to another set of canonical coordinates $(\boldsymbol{Q}, \boldsymbol{P})$ can be represented by a generating function, e.g., $F_2(\boldsymbol{q}, \boldsymbol{P}, t)$ depending on initial position coordinates $\boldsymbol{q}$ and new momentum coordinates $\boldsymbol{P}$. This leads to the following equations for the remaining coordinates $(\boldsymbol{Q}, \boldsymbol{p})$:

$$\boldsymbol{Q} = \frac{\partial F_2(\boldsymbol{q}, \boldsymbol{P}, t)}{\partial \boldsymbol{P}} \,, \tag{2.8}$$

$$\boldsymbol{p} = \frac{\partial F_2(\boldsymbol{q}, \boldsymbol{P}, t)}{\partial \boldsymbol{q}} \,, \tag{2.9}$$

and the Hamiltonian is also transformed to

$$K(\boldsymbol{Q}, \boldsymbol{P}) = H(\boldsymbol{q}, \boldsymbol{p}) + \frac{\partial F_2(\boldsymbol{q}, \boldsymbol{P}, t)}{\partial t} \,. \tag{2.10}$$

When we assume that those relations can be inverted, equations 2.8 and 2.9 give f relations for new momenta $\boldsymbol{P}$ and new positions $\boldsymbol{Q}$.

Splitting the generating function $F_2(\boldsymbol{q}, \boldsymbol{P}, t)$ into $F_2(\boldsymbol{q}, \boldsymbol{P}, t) = \boldsymbol{q} \cdot \boldsymbol{P} + F(\boldsymbol{q}, \boldsymbol{P}, t)$, we get

$$\boldsymbol{Q} = \boldsymbol{q} + \frac{\partial F(\boldsymbol{q}, \boldsymbol{P}, t)}{\partial \boldsymbol{P}} \,, \tag{2.11}$$

$$\boldsymbol{p} = \boldsymbol{P} + \frac{\partial F(\boldsymbol{q}, \boldsymbol{P}, t)}{\partial \boldsymbol{q}} \,, \tag{2.12}$$

as the first part of the split generating function corresponds to the identity transformation $(\boldsymbol{q}, \boldsymbol{p}) \mapsto (\boldsymbol{Q}, \boldsymbol{P})$. Discretizing in time with mapping time step h and setting $F(\boldsymbol{q}, \boldsymbol{P}, t) = hH(\boldsymbol{q}, \boldsymbol{P}, t)$, Eqs. 2.11 and 2.12 are equivalent to the semi-implicit symplectic Euler scheme (Hairer et al., 2006):

$$\boldsymbol{p}_{n+1} = \boldsymbol{p}_n - h\frac{\partial H(\boldsymbol{q}_n, \boldsymbol{p}_{n+1})}{\partial \boldsymbol{q}_n} \,, \tag{2.13}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + h\frac{\partial H(\boldsymbol{q}_n, \boldsymbol{p}_{n+1})}{\partial \boldsymbol{p}_{n+1}} \,, \tag{2.14}$$

where $(\boldsymbol{q}(t), \boldsymbol{p}(t))$ are equivalent to $(\boldsymbol{q}_n, \boldsymbol{p}_n)$ and $(\boldsymbol{q}(t+h), \boldsymbol{p}(t+h)) = (\boldsymbol{Q}, \boldsymbol{P})$ to $(\boldsymbol{q}_{n+1}, \boldsymbol{p}_{n+1})$. Each step of the integrator is a symplectic transformation.

It can easily be shown that the symplecticity is preserved by Eqs. 2.13 and 2.14 by differentiating with respect to $(\boldsymbol{q}_n, \boldsymbol{p}_n)$:

$$\underbrace{\begin{pmatrix} I & -h\frac{\partial^2 H}{\partial \boldsymbol{p}_n \partial \boldsymbol{p}_n} \\ 0 & I + h\frac{\partial^2 H}{\partial \boldsymbol{q}_n \partial \boldsymbol{p}_n} \end{pmatrix}}_{A} \begin{pmatrix} \frac{\partial \boldsymbol{q}_{n+1}}{\partial \boldsymbol{q}_n} & \frac{\partial \boldsymbol{q}_{n+1}}{\partial \boldsymbol{p}_n} \\ \frac{\partial \boldsymbol{p}_{n+1}}{\partial \boldsymbol{q}_n} & \frac{\partial \boldsymbol{p}_{n+1}}{\partial \boldsymbol{p}_n} \end{pmatrix} = \underbrace{\begin{pmatrix} I + h\frac{\partial^2 H}{\partial \boldsymbol{q}_n \partial \boldsymbol{p}_n} & 0 \\ -h\frac{\partial^2 H}{\partial \boldsymbol{q}_n \partial \boldsymbol{q}_n} & I \end{pmatrix}}_{B}. \tag{2.15}$$

This allows a direct calculation of $\frac{\partial(\boldsymbol{q}_{n+1}, \boldsymbol{p}_{n+1})}{\partial(\boldsymbol{q}_n, \boldsymbol{p}_n)} = \mathbf{A}^{-1}\mathbf{B}$ and straightforward matrix manipulations show that $\left(\frac{\partial(\boldsymbol{q}_{n+1}, \boldsymbol{p}_{n+1})}{\partial(\boldsymbol{q}_n, \boldsymbol{p}_n)}\right)^\top \mathbf{J} \left(\frac{\partial(\boldsymbol{q}_{n+1}, \boldsymbol{p}_{n+1})}{\partial(\boldsymbol{q}_n, \boldsymbol{p}_n)}\right) = \mathbf{J}$.

In contrast to non-symplectic integration schemes, the geometric structure of the Hamiltonian system is preserved with a symplectic integration scheme. Therefore, the results obtained are more accurate, and invariants of motion are conserved within bounds. Besides the symplectic Euler (see Eqs. 2.13 and 2.14), multiple symplectic integration schemes exist, also including higher-order schemes such as the symplectic Störmer-Verlet scheme or symplectic Runge-Kutta schemes (Hairer et al., 2006).

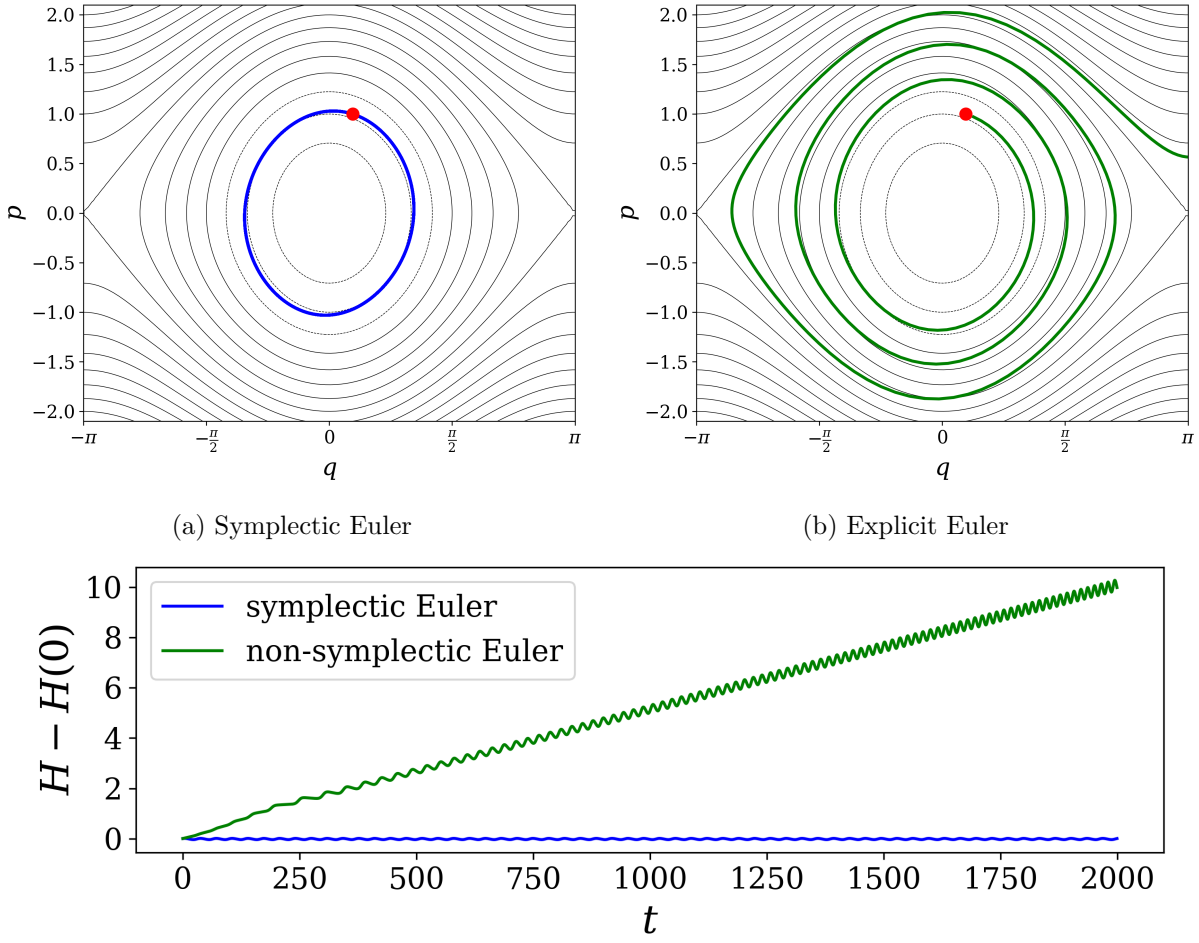
In Fig. 2.1, one pendulum orbit with initial conditions $(q, p) = (0.3, 1.0)$ in phase space and its energy are shown for two cases: calculated with a symplectic and non-symplectic integration scheme. Here, the same time step size is used for both integration schemes. While the orbit calculated using the symplectic scheme stays close to the constant energy surface, the non-symplectic trajectory gains energy and spirals outwards. When considering energy conservation for this case, the error in energy is bounded and small for the symplectic Euler method, while it grows linearly for the non-symplectic, explicit Euler (Hairer et al., 2006).

2.1.3 Hamiltonian chaos

In many Hamiltonian systems, chaos plays a vital role (Ott, 2002; Lichtenberg and Lieberman, 1992; José and Saletan, 1998; Zaslavsky, 2007). Although chaos in Hamiltonian systems is deterministic, meaning that no random elements are involved and the initial conditions fully determine the behavior of orbits, it is still not predictable (Werndl, 2009). The temporal evolution of the Hamiltonian system, e.g., the movement of charged particles in a magnetic field, is highly sensitive to the initial conditions. The chaotic system is not predictable in the sense that even tiny changes in initial conditions can lead to vastly different outcomes. In other words, although the current state uniquely defines future states, an approximation of the first does not approximately define the latter.

If an integrable[1] Hamiltonian system is perturbed and the perturbation is sufficiently small, there are regular trajectories as well as regions of stochasticity in phase space, that are well separated by regular trajectories. Due to the perturbation, the integrable system moves away from integrability and invariant tori, which confine regular trajectories, are distorted or destroyed. With increasing perturbation, more tori are destroyed, and the proportion of chaotic orbits increases.

[1] Integrability in Hamiltonian systems means that there exist enough independent Poisson commuting first integrals. For an integrable Hamiltonian system, a specific set of coordinates can be defined, so-called action-angle variables, in which the temporal evolution of the system is linear. They parametrize the motion on an invariant torus in phase space (Lichtenberg and Lieberman, 1992).



(a) Symplectic Euler

(b) Explicit Euler

(c) Energy conservation of the pendulum problem

Figure 2.1: Orbit of a pendulum with $H(q,p) = \frac{p^2}{2} - \cos(q)$ in phase space using symplectic Euler and non-symplectic, explicit Euler integration schemes with step size $h = 0.15$ for initial conditions $(q,p) = (0.3, 1.0)$ and $n = 2000$ time steps. The black contours in panels (a) and (b) depict orbits of constant energy.

A key challenge in analyzing chaotic Hamiltonian systems is the distinction between chaotic and regular orbits. A commonly used tool for doing so are Poincaré sections. We choose a lower-dimensional subspace of phase space and study the intersection of the trajectories with this surface (see Fig. 1.3.a in Lichtenberg and Lieberman (1992)). The resulting curves on this subspace allow insight into the dynamics of the underlying system as chaotic and regular orbits exhibit entirely different behavior. Especially for high-dimensional systems, Poincaré sections allow the visualization of properties of those dynamical systems.

The regular trajectories remain bounded on invariant $(2f-1)$-dimensional tori of the $2f$-dimensional phase space. Those regular trajectories are associated with the first integrals of motion. In the Poincaré plot, they appear as closed invariant curves, and when observed for a long time, the intersections densely cover the curve. Further, regular resonant trajectories exist, where all intersections with the distinct lower-dimensional subspace chosen for the Poincaré sections lie on several fixed points. An orbit near a resonant trajectory gives rise to islands that encircle the fixed points.

The stochastic or chaotic trajectories fill a finite portion of the Poincaré section but stay confined between two invariant curves stemming from regular trajectories. Depending on the strength of the perturbation, chaotic orbits might spread over the whole phase space when all invariant tori are destroyed. This is then called global stochasticity.

A very well-studied example to study chaos in Hamiltonian systems is the *standard map* (Chirikov, 1979):

$$p_{n+1} = (p_n + K\sin(q_n)) \bmod 2\pi \,, \tag{2.16}$$

$$q_{n+1} = (q_n + p_{n+1}) \bmod 2\pi \,. \tag{2.17}$$

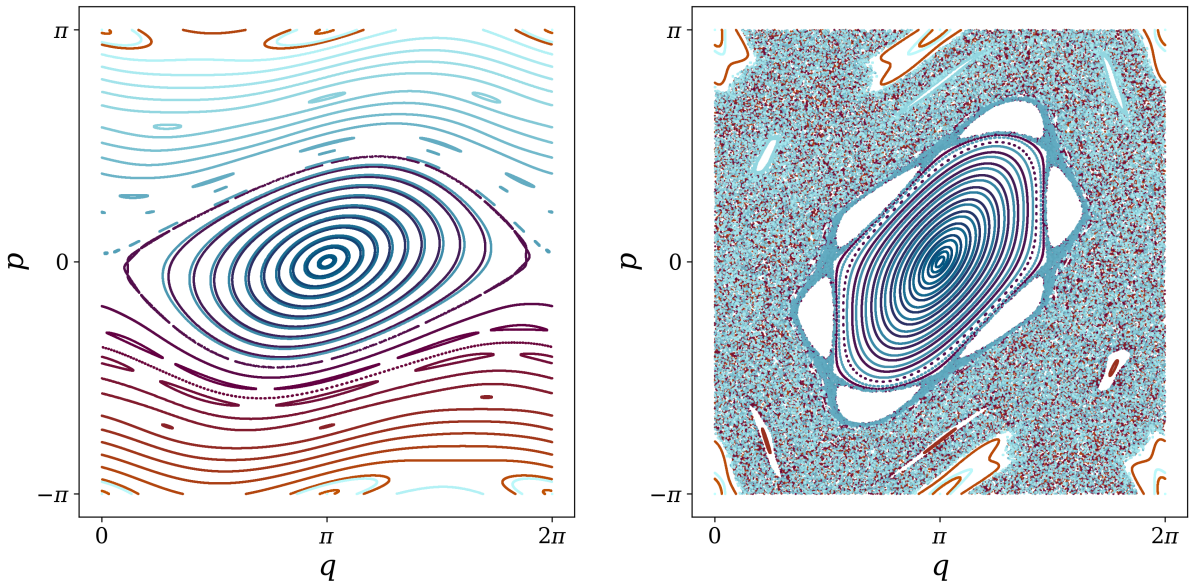
K is the stochasticity parameter giving the intensity of the perturbation. Each mapping step of the standard map corresponds to one Poincaré map of a kicked rotor with the following Hamiltonian

$$H(q,p,t) = \frac{p^2}{2} + K\cos(q) \sum_{n=-\infty}^{\infty} \delta\left(\frac{t}{T} - n\right) \,. \tag{2.18}$$

The kicked rotor is a bar fixed at one end to a frictionless pivot and subject to $T = 2\pi$-periodic kicks at the other end. The angular momentum p changes discontinuously at each kick but remains constant between the kicks. q corresponds to the angular position of the bar. The standard map allows the examination of (q,p) right after each kick. When $K = 0$, the system corresponds to a free rotator.

In Figure 2.2, the standard map is shown for different values of the stochasticity parameter K. Panel 2.2a shows a fairly regular phase space for $K = 0.5$, where small bands of stochasticity are very well separated and confined by regular regions. Initial points stay bound to a surface, which is indicated by the color scheme. In contrast, in panel 2.2b, the phase space is shown for a higher value of $K = 1.5$. There are still regular regions and island chains. However, they are surrounded by a global stochastic layer, which allows points with specific initial conditions within this "chaotic sea" to explore the whole phase space.

Distinguishing regular from chaotic orbits is a key challenge in analyzing chaotic Hamiltonian systems. Several approaches have been developed, many of which are based on the calculation of Lyapunov characteristic exponents λ_i. They measure the rate of exponential separation of



(a) $K = 0.5$ (b) $K = 1.5$

Figure 2.2: Standard map with different values of K, the stochasticity parameter. Different colours depict different initial values (q_0, p_0), where the RGB values are proportional to $(p_0, p_0 + q_0, q_0)$.

trajectories with initial conditions $\boldsymbol{z}(0) = (\boldsymbol{q}(0), \boldsymbol{p}(0))$ with perturbation $\delta\boldsymbol{z}$ over time (Eckmann and Ruelle, 1985; Benettin et al., 1980a,b; Skokos, 2009):

$$|\delta\boldsymbol{z}(T)| = \mathcal{J}^{(T)}_{\boldsymbol{z}(T)}\delta\boldsymbol{z}(0) \approx e^{T\lambda}|\delta\boldsymbol{z}(0)| \,. \tag{2.19}$$

Here, $\mathcal{J}^{(T)}_{\boldsymbol{z}(T)}$ is a time-ordered product of Jacobians $\mathcal{J}_{\boldsymbol{z}(T-1)}\mathcal{J}_{\boldsymbol{z}(T-2)}\dots\mathcal{J}_{\boldsymbol{z}(1)}\mathcal{J}_{\boldsymbol{z}(0)}$ (Eckmann and Ruelle, 1985). The spectrum of the Lyapunov characteristic exponents is then given as the logarithm of the eigenvalues of the following matrix:

$$\Lambda = \lim_{T\to\infty}[\mathcal{J}^{(T)\top}_{\boldsymbol{z}(T)}\mathcal{J}^{(T)}_{\boldsymbol{z}(T)}]^{1/(2T)} \,. \tag{2.20}$$

The maximal value of the Lyapunov characteristic exponents indicates whether orbits are of chaotic or regular nature as it gives the rate of exponential growth of an infinitesimal vector $\delta\boldsymbol{z}$. Hence, the corresponding orbit is chaotic if $\lambda_1 > 0$. The sum of the two largest Lyapunov characteristic exponents $\lambda_1 + \lambda_2$ gives the rate of exponential growth of a surface element. Generally, for a D-dimensional system, there exist D Lyapunov characteristic exponents giving the rate of growth of a D-volume element. This is equivalent to the rate of growth of the determinant of the Jacobian $\det(\mathcal{J}^{(T)}_{\boldsymbol{z}(T)})$ (Eckmann and Ruelle, 1985).

In Hamiltonian systems, the phase space is symplectic and therefore volume-preserving. Hence, the determinant of the Jacobian is constant and this implies that the Lyapunov characteristic exponents exist in additive inverse pairs, thus

$$\sum_{i}^{D} \lambda_i = 0 \,. \tag{2.21}$$

Lyapunov characteristic exponents are global in the sense that they are a measure of long-time phase space stability (in the limit $T \to \infty$). They are independent of initial conditions. Because computing the Lyapunov exponents would take a simulation over an infinite amount of time, we here use the finite time *local Lyapunov exponents* to distinguish between regular and chaotic orbits (Benettin et al., 1980a,b; Abarbanel et al., 1991). Local Lyapunov exponents depend on the phase space position $\boldsymbol{z}$ and time. They allow us to determine the predictability of a specific point in phase space for a finite time and to estimate the heterogeneity of phase space (Eckhardt and Yao, 1993; Abarbanel, 1992). These characteristics make the local Lyapunov exponents very well suited for characterizing stochastic and regular regions in Hamiltonian dynamical systems.

Another approach for orbit classification relies on box-counting fractal dimension of the set of points given by Poincaré intersections (Theiler, 1990; Albert et al., 2020). The fractal dimension depends on the number of boxes needed to cover all points in the set and is close to one in the regular case as the number of boxes grows linearly with the inverse size of the boxes. For chaotic orbits that spread over the whole phase space, the fractal dimension is well between one and two.

Contributions In Section 3.3, we use a symplectic surrogate model based on Gaussian processes with specialized covariance structure for early orbit classification based on local Lyapunov exponents (Rath et al., 2021b,a). The necessary Jacobians for several time steps are directly available from the symplectic surrogate model and can be inferred via the Hessian of the kernel function. We investigate the predictive performance of the proposed approach using the standard map with different values of the stochasticity parameter K and evaluate the distribution of the local Lyapunov exponents depending on the number of time steps necessary to classify orbits. For $K = 2.0$, where a large chaotic sea surrounds several islands of stability, we find that in the regular case, the distribution exhibits a sharp peak moving closer to zero with an increasing number of time steps. We investigate the rate of convergence of the block bias due to the finite number of mapping iterations to estimate the necessary number of mapping steps. This box bias vanishes in the limit $T \to \infty$. The distribution has a median larger than zero for chaotic orbits and is more spread out than in the regular case. In this case of $K = 2.0$, chaotic and regular regions can be distinguished more easily and also with fewer mapping iterations than in the other test case with $K = 0.9$. Here, several distinct stochastic layers are clearly separated by regular orbits. Orbits are weakly chaotic, meaning they exhibit chaotic behavior but stay close to hyperbolic fixed points. Hence, there is more variety in phase space and the transition between regular, weakly chaotic, and chaotic orbits is continuous. We use a Bayesian classifier trained on results from a reference method (generalized alignment index (Skokos et al., 2007)) to determine the probability of an orbit being regular based on the obtained local Lyapunov exponents from the surrogate model. Again, for $K = 2.0$, the rate of misclassification drops and stays constant at $\approx 1\%$ after 100 mapping iterations, while for $K = 0.9$ it remains constant at $\approx 10\%$ also after 100 mapping iterations due to the continuous transition between classes.

2.2 Gaussian process regression

Gaussian process (GP) regression is a non-parametric, Bayesian approach for representing smooth functions while allowing the incorporation of prior information and providing uncertainty measures over predictions (Rasmussen and Williams, 2005; Schölkopf and Smola, 2018; MacKay, 2003; Bishop, 2006; Murphy, 2022).
The following section introduces GPs and gives the key equations for predicting mean and covariance. The incorporation of derivative observations in the GP model and hyperparameter optimization are explained. Different covariance functions are introduced, and, finally, contributions to a specialized kernel design are presented.

When considering linear regression from a Bayesian perspective, a non-linear function $f(\boldsymbol{x})$ parameterized by weights $\boldsymbol{w}$ depends on input data $\boldsymbol{x}$ aggregated in matrix $\mathbf{X}$. We observe noisy data $y(\boldsymbol{x}) = f(\boldsymbol{x}) + \varepsilon$ and adapting the model to the observations can be described as inferring the underlying function $f(\boldsymbol{x})$ based on the available data. The posterior distribution over the weights is calculated via Bayes' rule:

$$p(\boldsymbol{w}|\boldsymbol{y}, \mathbf{X}) = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{y}|\mathbf{X})} \,, \tag{2.22}$$

where $p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w})$ is the likelihood and $p(\boldsymbol{w})$ the prior. The normalizing constant $p(\boldsymbol{y}|\mathbf{X})$ is the marginal likelihood and independent of $\boldsymbol{w}$. Assuming a prior distribution on the weights induces a prior distribution over functions. From the posterior distribution over $\boldsymbol{w}$, function values for new input values $y(\boldsymbol{x}_*)$ can be predicted.

In GP regression, the GP prior is directly postulated on the space of functions without the parameterization of $y(\boldsymbol{x})$. This allows inference on the function directly. We can think of the Gaussian process,

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \,, \tag{2.23}$$

as a generalization of a Gaussian distribution over a finite vector space to a function space of infinite dimension (Rasmussen and Williams, 2005; MacKay, 2003). While random variables from a uni- or multivariate Gaussian distribution are scalars or vectors, respectively, random variables from a Gaussian, or more generally, a stochastic process, are functions (A. Papoulis, 1991): A stochastic process is a collection of random variables $f(x) : x \in \mathcal{X}$, defined on a common probability space, indexed by elements from some set $\mathcal{X}$, the so-called index set. The finite-dimensional distributions of the stochastic process, which entail the joint distribution of a finite number of random variables from that process, satisfy the consistency property: the joint distributions of these variables remain consistent when more random variables from the same stochastic process are considered. This is also the basis for the definition of stochastic processes from the definition of a collection of all finite-dimensional marginals that are consistent (see Kolmogorov Extension Theorem in e.g. (Khoshnevisan, 2002)).

In the case of a GP, any finite number of random variables produced by the GP follows a multivariate normal distribution

$$p(\boldsymbol{f}|\boldsymbol{m}, \mathbf{K}) = \frac{1}{(2\pi)^{(n/2)}}|\mathbf{K}|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\boldsymbol{f}-\boldsymbol{m})^\top \mathbf{K}^{-1}(\boldsymbol{f}-\boldsymbol{m})\right) \,. \tag{2.24}$$

In most application cases, the mean $m(\boldsymbol{x})$ will be set to zero without any loss of generality.

Analogously to a Gaussian distribution over a finite-dimensional vector space being fully defined by its mean and covariance, a GP is fully characterized by its mean function $m(\boldsymbol{x})$,

$$m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})] \,, \tag{2.25}$$

and covariance function $k(\boldsymbol{x}, \boldsymbol{x'})$ that expresses the covariance between function values at $\boldsymbol{x}$ and $\boldsymbol{x'}$:

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \mathbb{E}[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x'}) - m(\boldsymbol{x'}))] \,. \tag{2.26}$$

The covariance function determines the entries $[\mathbf{K}]_{ij} = k(x_i, x_j)$ of the covariance matrix $\mathbf{K}$. Choices for valid covariance functions are given in section 2.2.3.

The marginalization property holds as a GP is a stochastic process and thus a collection of random variables. This consistency requirement means that if the GP defines $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{m}, \mathbf{K})$, it also specifies $y_1 \sim \mathcal{N}(m_1, \mathbf{K}_{11})$, when y_1 is a subset of $\boldsymbol{y}$ and $\mathbf{K}_{11}$ is a submatrix of $\mathbf{K}$. This means that the examination of a larger set leaves the distribution of a smaller subset unchanged.

In the context of regression, we usually work with noisy observations $\boldsymbol{y}$ of the latent function $\boldsymbol{f}$ corrupted by Gaussian noise ε with variance σ_n^2 at n training points $\boldsymbol{x}$. The input vector $\boldsymbol{x}$ is of dimension D and all n training data points are aggregated in the $(n \times D)$ design matrix $\mathbf{X}$. Here, we assume that the noise is independent for each observation y_i and not correlated. Given the training data, we want to make predictions of the latent functions at n_* new inputs $\boldsymbol{x}_*$. Similarly to the training points, test points are aggregated in the matrix $\mathbf{X}_*$. The joint prior of $\boldsymbol{y}$ and the test output $\boldsymbol{f}_* = f(\boldsymbol{x}_*)$ is given by:

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_y(\mathbf{X}, \mathbf{X}) & \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*, \mathbf{X}) & \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right) \,, \tag{2.27}$$

Here, the matrix $\mathbf{K}_y(\mathbf{X}, \mathbf{X})$ gives the covariance evaluated at the training points with an added diagonal noise matrix $\mathbf{K}_y(\mathbf{X}, \mathbf{X}) = \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}$. Due to the assumption that the noise is independent, the noise term is diagonal.

The conditional distribution of the joint Gaussian prior distribution on the observations $p(\boldsymbol{f}_* | \mathbf{X}_*, \mathbf{X}, \boldsymbol{y})$ is given as (see (Rasmussen, 2003, p. 200, Eq. A6))

$$\boldsymbol{f}_* | \mathbf{X}_*, \mathbf{X}, \boldsymbol{y} \sim \mathcal{N}(\mathbf{K}(\mathbf{X}_*, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\boldsymbol{y}, \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{K}(\mathbf{X}, \mathbf{X}_*)) \,, \tag{2.28}$$

which leads to the key predictive equations for GP regression: the predictive mean and covariance for a test point x_* are given by

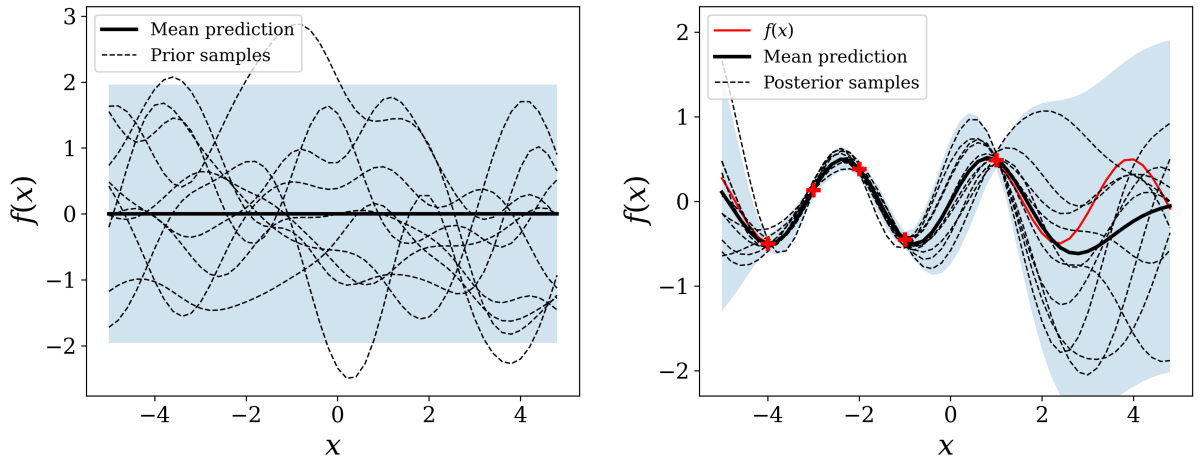
$$f(x_*) = \boldsymbol{k}_*^\top \mathbf{K}_y^{-1} \boldsymbol{y} \,, \tag{2.29}$$

$$\operatorname{cov}(f(x_*)) = \boldsymbol{k}_{**} - \boldsymbol{k}_*^\top \mathbf{K}_y^{-1} \boldsymbol{k}_* \,, \tag{2.30}$$

where $\boldsymbol{k}_* = \mathbf{K}(\mathbf{X}, \mathbf{X}_*)$ denotes the covariance between n input points and the n_* test points and $\mathbf{K}_y$ includes the noise in the covariance matrix of the input points. $\boldsymbol{k}_{**} = \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*)$ is the covariance between two test points. It is important to note that the predictive covariance only depends on the input and test points $\mathbf{X}$ and $\mathbf{X}_*$ but not on the observed values $\boldsymbol{y}$ at these input points.

There are many possibilities for choosing the covariance function $k(x, x')$ as discussed in Section 2.2.3. For now, we use the squared exponential covariance function that defines the covariances between pairs of random variables as

$$\operatorname{cov}(f(x), f(x')) = k(x, x') = \sigma_f^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right) \,, \tag{2.31}$$



(a) Prior (b) Posterior

Figure 2.3: Comparison of prior and posterior. (a) depicts 10 samples drawn from the prior distribution with a squared exponential covariance function. (b) shows 10 samples drawn from the inferred posterior distribution after observing 5 (noisy) data points (depicted as red + with 5% Gaussian noise). The hyperparameters are $l = 1.2$ and $\sigma_f^2 = 1.0$ (see Sections 2.2.2 and 2.2.3). The true underlying function $f(x) = \sin(x)\cos(x)$ is shown as a red solid line, while the estimated mean is depicted as a black solid line. The shaded regions correspond to two standard deviations at each input point x.

where σ_f and l are tunable hyperparameters.

In Figure 2.3, a comparison between the samples drawn from the prior and from the posterior is shown. In Figure 2.3a, we draw random Gaussian samples with the covariance matrix $\mathbf{K}(\mathbf{X}_*, \mathbf{X}_*)$ and zero mean. After observing measurements at training data points, the prior is conditioned on these observations, and we can calculate the predictive mean and covariance following Eqs. 2.29 and 2.30 for the same test points $\mathbf{X}_*$.

Regarding numerical stability, it is not beneficial to directly invert the covariance matrix $\mathbf{K}$ in Eqs. 2.29 and 2.30 (Murphy, 2022; Rasmussen and Williams, 2005). A better option is to use the Cholesky decomposition, which decomposes a symmetric, positive definite matrix into a product of a lower triangular matrix and its transpose:

$$\mathbf{K} = \mathbf{L}\mathbf{L}^\top . \tag{2.32}$$

The Cholesky decomposition is a common strategy for solving linear systems with symmetric positive semidefinite coefficient matrices $\mathbf{K}\boldsymbol{x} = \boldsymbol{b}$ by first solving $\mathbf{L}\boldsymbol{y} = \boldsymbol{b}$ by forward substitution and afterwards $\mathbf{L}^\top \boldsymbol{x} = \boldsymbol{y}$ by back substitution. The solution can then be written as $\boldsymbol{x} = \mathbf{L}^\top \backslash \boldsymbol{y} = \mathbf{L}^\top \backslash (\mathbf{L} \backslash \boldsymbol{b})$, where the notation of the backslash operator is used. The Cholesky decomposition has the advantage of being numerically stable ((Rasmussen, 2003, p. 202)). Its drawback is that its computational complexity is $\mathcal{O}(n^3)$ and is therefore not suitable for very big matrices. Then, the posterior mean can be calculated using $\boldsymbol{\alpha} = \mathbf{L}^\top \backslash (\mathbf{L} \backslash \boldsymbol{y})$ as

$$\mathbb{E}(f(x_*)) = \boldsymbol{k}_*^\top \mathbf{K}_y^{-1} \boldsymbol{y} = \boldsymbol{k}_*^\top \mathbf{L}^\top (\mathbf{L}^{-1} \boldsymbol{y}) = \boldsymbol{k}_*^\top \boldsymbol{\alpha} . \tag{2.33}$$

Similarly, we get the variance for each test point,

$$\mathrm{cov}(f(x_*)) = \boldsymbol{k}_{**} - \boldsymbol{k}_*^\top \mathbf{K}_y^{-1} \boldsymbol{k}_* = \boldsymbol{k}_{**} - \boldsymbol{k}_*^\top \mathbf{L}^\top \mathbf{L}^{-1} \boldsymbol{k}_* \,. \tag{2.34}$$

To draw samples $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{m}, \mathbf{K})$ from the estimated posterior, we use $\mathbf{L}$ and generate $\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. To get samples with the desired distribution, we calculate $\boldsymbol{x} = \boldsymbol{m} + \mathbf{L}\boldsymbol{u}$ (Rasmussen and Williams, 2005).

Multi-output Gaussian processes Up to now, we have only considered the modeling of scalar functions f. However, it is also possible to work with vector-valued functions, meaning we are interested in the relationship between an input space $\mathbb{R}^P$ and an output space $\mathbb{R}^D$. Multi-output GP regression is very similar to the single output case, with the only difference being that now the random variables are associated with different processes evaluated at different values of $\boldsymbol{x}$ (Alvarez et al., 2012). We take a vector-valued function $\boldsymbol{f} \sim \mathcal{GP}(\boldsymbol{m}, \mathbf{K})$, where D mean functions are stacked forming the mean function vector $\boldsymbol{m}$. $\mathbf{K}$ is a positive matrix valued function and its entries $[\mathbf{K}]_{d,d'}(\boldsymbol{x}, \boldsymbol{x}')$ correspond to the covariances between outputs $f_d(\boldsymbol{x})$ and $f_{d'}(\boldsymbol{x}')$. The predictive mean and covariance are the same as in the single output case given in Eqs. 2.29 and 2.30. Kernels for multi-output GPs will be discussed below.

2.2.1 Derivative observations

As the derivative of a Gaussian process still is a Gaussian process, the framework of GPs can easily be extended to include derivative information to improve model accuracy or enforce known constraints (Rasmussen and Williams, 2005; Solak et al., 2003; O'Hagan, 1992; Rasmussen, 2003; Eriksson et al., 2018). Additionally, it also allows for inference based on derivative information. An application case is the inference of position when data are available from sensors that measure velocity or acceleration, which are employed in extensions of the observation vector and the covariance matrix. The observation vector is augmented with derivative observations $(\boldsymbol{y}, \nabla\boldsymbol{y})$. The covariance matrix is defined as

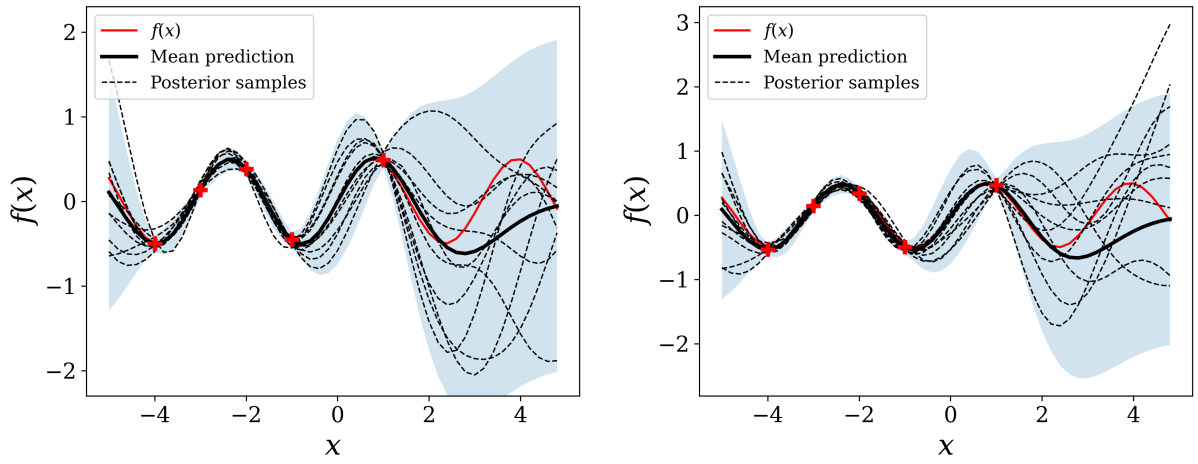
$$k^\nabla(\boldsymbol{x}, \boldsymbol{x}') = \begin{pmatrix} k(\boldsymbol{x}, \boldsymbol{x}') & \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{x}') \\ (\nabla_{\boldsymbol{x}'} k(\boldsymbol{x}, \boldsymbol{x}'))^\top & \nabla^2 k(\boldsymbol{x}, \boldsymbol{x}') \end{pmatrix} . \tag{2.35}$$

Now, the covariance matrix includes covariances between function observations y_i and derivatives $\frac{\partial y_i}{\partial x_i}$ as well as covariances between two derivative observations:

$$\mathrm{cov}\left(y_i, \frac{\partial y_j}{\partial x_j}\right) = \frac{\partial k(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial x_j}, \tag{2.36}$$

$$\mathrm{cov}\left(\frac{\partial y_i}{\partial x_i}, \frac{\partial y_j}{\partial x_j}\right) = \frac{\partial^2 k(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial x_i \partial x_j} \,. \tag{2.37}$$

Function values and derivative observations often have different noise levels, which can be accounted for in a diagonal contribution with different hyperparameters.



(a) Posterior

(b) Posterior with derivative observations

Figure 2.4: Comparison of posterior prediction (a) without and (b) with derivative observations. Both panels show the predicted mean (black solid line) along with 10 samples drawn from the estimated posterior distribution after observing 5 (noisy) data points (depicted as red + with 5% Gaussian noise). The hyperparameters are $l = 1.2$ and $\sigma_f^2 = 1.0$. The true underlying function $f(x) = \sin(x)\cos(x)$ is shown as a red solid line. The shaded regions correspond to two standard deviations at each input point x.

In Figure 2.4, the same test case as in Figure 2.3 is depicted with the difference that now derivative information at the observation points are included. The prediction is improved as the observations of the derivatives have a constraining effect.

2.2.2 Hyperparameter optimization

Learning in GP regression means adapting the hyperparameters $\boldsymbol{\theta}$ of the covariance function, typically having a length scale l, noise variance σ_n^2 and amplitude parameter σ_f^2, that controls the vertical scale of the function. These hyperparameters usually have significant effects on the predictions from the model. Finding the optimal values of the hyperparameters is based on the maximization of the marginal likelihood

$$p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{\theta}) = \int p(\boldsymbol{y}|\boldsymbol{f}, \mathbf{X}) p(\boldsymbol{f}|\mathbf{X}, \boldsymbol{\theta}) d\boldsymbol{f} \,. \tag{2.38}$$

The prior is Gaussian $p(\boldsymbol{f}|\mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{f}|\mathbf{0}, \mathbf{K})$, where we assume a zero mean function. The likelihood is a factorized Gaussian $p(\boldsymbol{y}|\boldsymbol{f}, \mathbf{X}) = \prod_{i=1}^{n} \mathcal{N}(y_i|f_i, \sigma_n^2) = \mathcal{N}(\boldsymbol{f}, \sigma_n^2 \mathbf{I})$. Then the marginal log-likelihood is given by

$$\log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^\top \mathbf{K}_y^{-1} \boldsymbol{y} - \frac{1}{2}\log|\mathbf{K}_y| - \frac{n}{2}\log(2\pi) \,, \tag{2.39}$$

where the covariance matrix $\mathbf{K}$ implicitly depends on the hyperparameters $\boldsymbol{\theta}$. The first term depends on the observations $\boldsymbol{y}$ and is the data fit. The second term penalizes the model complexity and the last term is a normalization constant. Optimization of the hyperparameters is a trade-off

between the first two terms: the data fit is good for small values of the length scale l. However, the model complexity is high as $\mathbf{K}$ is close to being diagonal. In contrast, the data fit is worse for large length scales as the model is less flexible, but there is less penalization in the model complexity term.

Using the Cholesky decomposition of the covariance matrix, we also get a simplified form for the marginal likelihood,

$$\log p(\boldsymbol{y}|\mathbf{X},\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^\top\boldsymbol{\alpha} - \sum_{i=1}^{n}\log L_{ii} - \frac{n}{2}\log(2\pi)\,, \tag{2.40}$$

where $\boldsymbol{\alpha}$ again is $\mathbf{L}^\top\backslash(\mathbf{L}\backslash\boldsymbol{y})$. When using gradient-based methods for the optimization of the marginal likelihood, its gradients are needed:

$$\frac{\partial}{\partial\theta_j}\log p(\boldsymbol{y}|\mathbf{X},\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{y}^\top\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}\mathbf{K}_y^{-1}\boldsymbol{y} - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}\right) \tag{2.41}$$

$$= \frac{1}{2}\mathrm{tr}\left((\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - \mathbf{K}_y^{-1})\frac{\partial\mathbf{K}_y^{-1}}{\partial\theta_j}\right)\,. \tag{2.42}$$

The computational complexity is still dominated by the inversion of the covariance matrix, as the computational complexity for calculating the gradients is $\mathcal{O}(n^2)$ for each hyperparameter. Hence, the use of a gradient-based optimizer is recommended. However, the marginal log-likelihood is generally a non-convex function, meaning optimization is non-trivial.

2.2.3 Covariance functions

The dynamics of a GP are entirely defined by the choice of the covariance function k, the kernel of the stochastic process. The covariance function expresses the similarity between function values evaluated at two data points:

$$\mathrm{cov}(f(x), f(x')) = k(x, x'). \tag{2.43}$$

A valid covariance function produces a covariance matrix $\mathbf{K}$ with entries $[\mathbf{K}]_{ij} = k(x_i, x_j)$ that is positive semidefinite (PSD) for all possible entries of x_i (Rasmussen and Williams, 2005). Below some choices of covariance functions are discussed and random samples drawn from GP priors with different covariance functions are shown in Fig. 2.5.

A widely used covariance function is the squared exponential covariance function (Fig. 2.5a),

$$k_{\mathrm{SE}}(x, x') = \sigma_f^2\,\exp\left(-\frac{(x-x')^2}{2l^2}\right)\,, \tag{2.44}$$

where l and σ_f are hyperparameters defining the characteristic length scale and the vertical scale of variations, respectively. This kernel is very smooth as it is infinitely differentiable. The squared exponential covariance function is a stationary covariance function depending on $\boldsymbol{x} - \boldsymbol{x}'$ as it is invariant to translations in the input space.

Another choice is represented by a family of Matérn kernel functions (Fig. 2.5c) of the following form

$$k_{\mathrm{M}}(\tau) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{\tau}{l} \right)^{\nu} K_{\nu} \left(\sqrt{2\nu} \frac{\tau}{l} \right) , \tag{2.45}$$

where $\tau = x - x'$ and $K_{\nu}(\cdot)$ is the modified Bessel function of the second kind. The hyperparameters ν, l and σ_f^2 control the smoothness, length scale, and vertical magnitude, respectively. Higher values of ν correspond to smoother functions. In the limit $\nu \to \infty$, we obtain the squared exponential kernel (Eq. 2.44). For small values of ν, the Matérn kernel gives rise to rougher and more wiggly functions, which are well suited for modeling data with local fluctuations without decreasing the overall length scale. In Fig. 2.5c, random functions from a GP with a Matérn covariance function are shown. We set $\nu = 3/2$, where the covariance function becomes

$$k_{\mathrm{M},\nu=3/2}(\tau) = \left(1 + \frac{\sqrt{3}\tau}{l} \right) \exp \left(-\frac{\sqrt{3}\tau}{l} \right) . \tag{2.46}$$

Furthermore, an interesting covariance function is the periodic kernel (Fig. 2.5d) capturing periodic structures in the data,

$$k_{\mathrm{per}}(\tau) = \sigma_f^2 \exp \left(-\frac{2\sin^2(\frac{\pi\tau}{p})}{l^2} \right) , \tag{2.47}$$

where p is the periodicity, σ_f^2 and l are the magnitude and length scale, respectively.

The linear kernel can be obtained from linear regression,

$$k_{\mathrm{lin}}(x, x') = \sigma_b^2 + \sigma_f^2 x x' , \tag{2.48}$$

where σ_b^2 gives the offset at zero. This is not a stationary kernel, but it is invariant to rotations of coordinates around the origin.

The combination of valid kernel functions again results in valid kernel functions, e.g., via multiplication (Fig. 2.5i) (Rasmussen and Williams, 2005; Bishop, 2006),

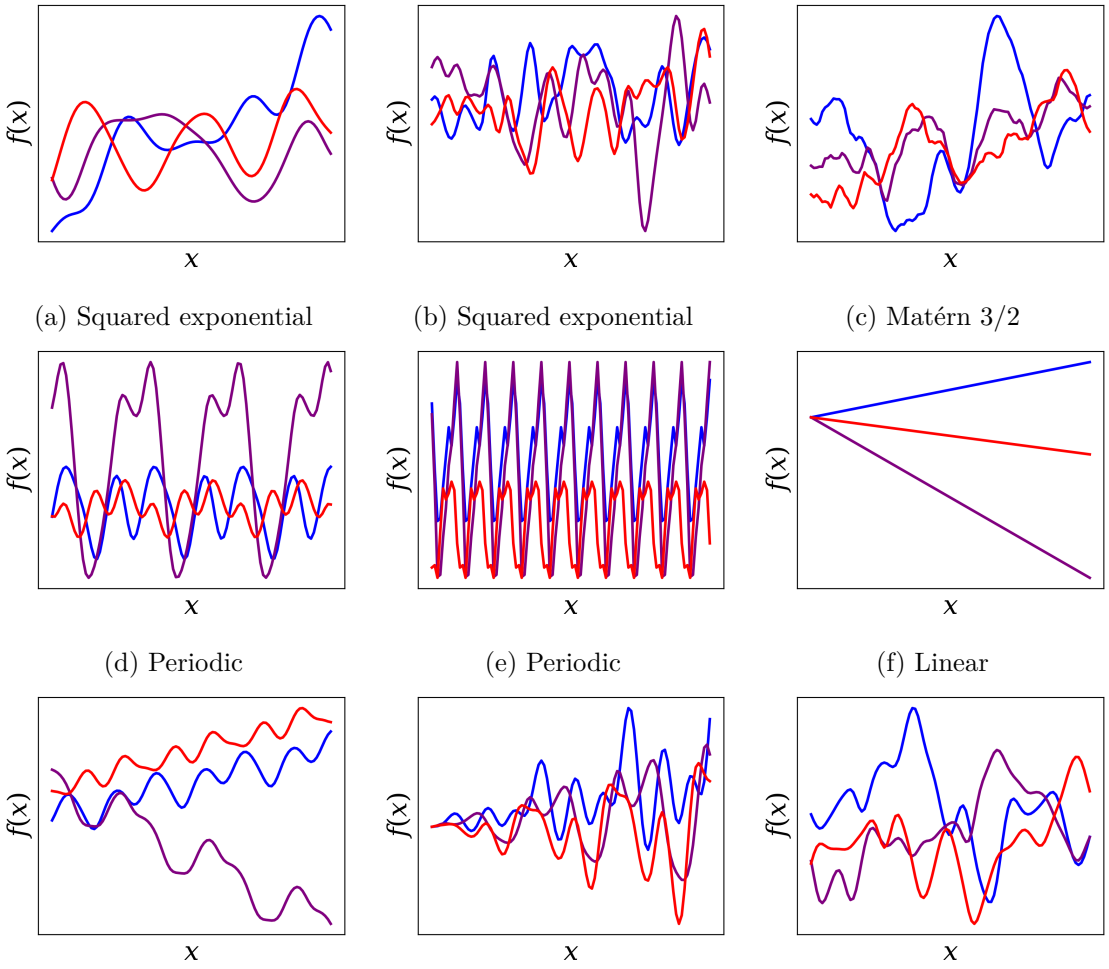
$$k(x, x') = k_1(x, x') \cdot k_2(x, x') , \tag{2.49}$$

or addition (Fig. 2.5g)

$$k(x, x') = k_1(x, x') + k_2(x, x') . \tag{2.50}$$

This allows to combine features of each kernel, e.g., periodicity with a linear trend.

Fig. 2.5 shows some examples of random functions sampled from GPs with different covariance functions. When comparing Fig. 2.5a and b, the influence of the length scale l is immediately recognizable: while $l = 1.0$ in panel (a) and the resulting samples are smooth and vary slowly along x, the samples in panel (b) with $l = 0.2$ are more wiggly. Larger length scales imply that even distant points have a higher degree of correlation and the covariance matrix has off-diagonal elements that decrease more slowly with distance. The GP is then less sensitive to local fluctuations and might fail to capture patterns in the data. In contrast, smaller values of l lead to faster decaying off-diagonal elements in the covariance matrix and result in a flexible model that might even be sensitive to noise in the data.



(a) Squared exponential (b) Squared exponential (c) Matérn 3/2

(d) Periodic (e) Periodic (f) Linear

(g) Periodic + linear (h) Periodic · linear (i) Locally periodic

Figure 2.5: Examples of valid covariance functions and their combinations. Three samples drawn from GPs with different covariance functions are shown in each panel. (a) and (b) show both samples using a squared exponential kernel, but with different length scales l ((a): $l = 1$, (b): $l = 0.2$). The smaller length scale produces samples that are more wiggly and change faster. In (c), samples using a Matérn kernel with $\nu = 3/2$ and $l = 1.0$ are shown. (d) and (e) depict samples using periodic covariance functions with different periodicity p ((d): $p = \pi$, (e): $p = 1.0$). GPs with a linear covariance function allow to generate samples shown in (f). This kernel is particularly useful in combination with other kernels. In (g), the covariance function is a periodic kernel with an added linear trend. When multiplying a periodic with a linear kernel, we get samples with increasing amplitude when moving from the origin as shown in (h). A locally periodic kernel (i), whose periodicity can change over the input space, combines a periodic kernel with a squared exponential kernel via multiplication.

Covariance functions for multi-output Gaussian process regression When working with vector-valued functions of dimension D, covariance functions have to be adapted to work in this setting (Alvarez et al., 2012). When we assume that there are no correlations between the processes and thus make the multiple outputs independent of each other, important information might be lost. This situation corresponds to a covariance matrix $\mathbf{K}$ built from block matrices where the diagonal blocks correspond to the covariance matrices from single output GPs and the off-diagonal blocks equal to zero, implying there is no cross-correlation between the output processes. There are several approaches to incorporate cross-correlations, e.g., based on separable kernels like the intrinsic co-regionalization model (Alvarez et al., 2012) or via process convolutions where the different output channels are assumed to share one underlying noise process but are subject to different convolutions (Boyle and Frean, 2004).

2.3 Contributions towards specialized kernel design

2.3.1 Constrained Gaussian processes

When working with data that are known to fulfill a given differential equation, it is beneficial to construct the GP in a way such that the predictions satisfy laws of physics by construction (Swiler et al., 2020; Raissi et al., 2017; Särkkä, 2011). When dealing with a linear PDE, the respective linear operator $\mathcal{L}$ can be included naturally in GP regression as a GP subject to $\mathcal{L}$ is still a GP (Adler, 2010, Sec. 2.2):

$$\mathcal{L}f(\boldsymbol{x}) \sim \mathcal{GP}(\mathcal{L}_{\boldsymbol{x}} m(\boldsymbol{x}), \mathcal{L}_{\boldsymbol{x}} \mathbf{K}(\boldsymbol{x}, \boldsymbol{x}') \mathcal{L}_{\boldsymbol{x}'}^{\top}) \,. \tag{2.51}$$

As differentiation and integration are linear operators, they can be included seamlessly in GP regression as well (Särkkä, 2011; Raissi et al., 2017; Mendes and da Costa Júnior, 2012; Cockayne et al., 2017).

Contributions In the contribution presented in Section 3.1, we derive specialized kernels for given linear differential equations with vanishing or localized sources (Albert and Rath, 2020). The method is based on the superposition of a GP with a specialized kernel exactly fulfilling the homogeneous part of the differential equation and a linear model for point source contributions. The construction of specialized kernels for homogeneous differential equations is based on fundamental solutions via Mercer's theorem. The superimposed model for localized sources is constructed via a linear model over fundamental solutions. Specialized kernels are derived for Laplace's equation, the heat equation, and the Helmholtz equation. The hyperparameters of the kernels have a direct correspondence with the parameters of the underlying equations. For the application case of the Helmholtz equation, source position and strength are estimated. We compare the presented kernels to a squared exponential kernel using the same number of training data points. Results from the squared exponential kernel do not satisfy the underlying equations exactly and are not as stable and accurate as those produced by the specialized kernels of our approach. Additionally, the parameters of the specialized kernels allow physical interpretability, e.g., diffusivity and wave number.

2.3.2 Incorporating symplectic geometry

Other work regarding symplecticity in machine learning Recently, several attempts have been made to incorporate symplectic geometry in machine learning models. Besides multiple approaches based on GPs, there have also been several attempts to incorporate symplecticity into neural networks (e.g. (Greydanus et al., 2019; Finzi et al., 2020; Jin et al., 2020; Burby et al., 2020; Duruisseaux et al., 2023; Cranmer et al., 2020; Toth et al., 2019; Chen et al., 2019; Brantner et al., 2023; Brantner and Kraus, 2023)).

Hamiltonian neural networks (HNNs) model a Hamiltonian H based on observations of phase points (and their derivatives) by using a loss function based on Hamilton's equations and then solve the resulting differential equations (Greydanus et al., 2019). In contrast to HNNs, SympNets are neural networks that directly learn the flow map as opposed to the Hamiltonian of the system (Jin et al., 2020). A SympNet is a composition of many, relatively simple layers. Each of these layers enforces symplecticity separately. As the composition of symplectic maps is again symplectic, the entire SympNet also has this property.

HénonNets are similar in architecture, their input-output mapping is a canonical symplectic map and they are designed to learn Poincaré maps of symplectic systems (Burby et al., 2020; Duruisseaux et al., 2023). In contrast to traditional field-line following, modeling Poincaré maps is orders of magnitude faster.

Using GPs for Hamiltonian systems has the advantage that uncertainties present in the training data can be considered. Several approaches aim to learn directly from (noisy) state trajectories (Tanaka et al., 2022; Ross and Heinonen, 2023) by placing a GP prior over the Hamiltonian H. The Hamiltonian GP can directly be embedded in a symplectic integrator (Ensinger et al., 2023). In Offen and Ober-Blöbaum (2022), symplectic shadow integration is proposed where an inverse modified Hamiltonian is learned from data to compensate for discretization errors. Another approach is based on the direct identification of the Hamiltonian from data via derivatives of the flow map (Bertalan et al., 2019).

Contributions In the contribution presented in Section 3.2, we introduce Symplectic Gaussian Processes (SympGPR) that are based on learning the generating function of the flow map of the underlying Hamiltonian system (Rath et al., 2021b). In contrast to traditional numerical integration schemes that approximate orbits based on the knowledge of the Hamiltonian H, the training data for the GP are given orbit data over a (possibly large) mapping time step, which might even be a full Poincaré section. Based on the given observations, SympGPR should then find an approximation of the flow map without the knowledge of H. Once the flow map is learned, it can be applied subsequently to calculate the dynamics of the model over many periods. As the mapping time step where the orbit data is given does not have to be small compared to the dynamics of the model under investigation, SympGPR can be used as a fast surrogate model for orbit tracing. Depending on the complexity of the Hamiltonian system under investigation, it is even possible to directly interpolate Poincaré sections without the calculation of the full orbit.

SympGPR is a multi-output GP that includes derivative observations of the generating function and is equipped with a particular covariance structure. The covariance function is constructed so that the learned flow map is guaranteed to be symplectic. Depending on the choice of the kernel, two methods are presented: a product kernel results in an accurate implicit scheme, whereas a

sum kernel gives a fast explicit scheme. The latter case corresponds to a separable Hamiltonian, which leads to the symplectic Euler becoming fully explicit. Although the generating function F is not observed directly, it can be inferred from the derivative observations ∇F. For general Hamiltonian systems, F is approximately hH (up to a constant) for sufficiently small mapping steps h, and thus, the Hamiltonian can also be inferred by the SympGPR.

The method is tested on several non-chaotic Hamiltonian systems: flow maps of the pendulum with different mapping time steps and Poincaré sections of the perturbed pendulum and the Hénon-Heiles system. Finally, chaotic systems, such as the standard map and magnetic field lines in a tokamak with non-axisymmetric perturbations are investigated. Despite the small number of training data points, SympGPR can capture the dynamics in phase space in all test cases. Also, in chaotic systems, the onset of chaos and accelerator mode islands are reproduced correctly. Investigations on the diffusion of chaotic orbits in the standard map show that the obtained diffusion rates with SympGPR match the theoretical predictions. For higher energies in the Hénon-Heiles system, the generating function becomes multi-valued, and therefore, future states are not predictable without further measures. For the test case of magnetic field line tracing, we present a split SympGPR, where the Poincaré map for a full turn 2π is split into several sub-steps to ensure that the generating function is sufficiently smooth and unique. For each of the m sub-steps, we introduce an independent SympGPR map that represents a leap of $2\pi/m$ of the full Poincaré map. This produces more stable and reliable results, but comes with higher computational costs in training as four independent GPs have to be trained and m-times more operations have to be carried out when applying the SympGPR map. However, due to the small number of training data points (≈ 100) the necessary CPU time for evaluating 2000 Poincaré maps is still reduced significantly when compared to a symplectic Euler integrator. However, splitting is not always possible as for the Hénon-Heiles system, no additional surfaces for splitting can be identified. One possibility to approach the non-unique generating function is to consider an unwinding transformation of the generating function to allow a unique prediction by the GP. Compared to alternative methods based on symplectic neural networks, the number of training data points needed for training the SympGPR is considerably smaller and SympGPR is competitive with respect to run time and accuracy.

As outlined in Section 2.1.3, where the contribution of Section 3.3 is summarized, the symplectic surrogate model allows the direct calculation of the Jacobians needed for the estimation of Lyapunov exponents, which are then used for orbit classification (Rath et al., 2021a).

2.4 Student-*t* process regression

GP regression struggles when dealing with data with many outliers or heavy noise tails. In these cases, working with Student-t process regression instead can be beneficial (Shah et al., 2014; Roth et al., 2017). Student-t processes (TPs) generalize GPs with an additional hyperparameter ν controlling the distribution's kurtosis. Hence, depending on ν, outliers and heavier noise are assigned higher probability by the posterior distribution. While the estimated posterior covariance of a GP solely relies on the input data points $\mathbf{X}$, a TP takes the scattering of observations into account when estimating the covariance. This also leads to more realistic uncertainty estimates in case of severe outliers. TP regression shares the advantages of GP regression: the computational complexity is the same and there is a closed-form expression for the posterior distribution.

A Student-t process is defined similarly to a GP. Let f be a Student-t process with degrees of freedom $\nu > 2$, mean $\boldsymbol{\mu}$, and covariance matrix $\mathbf{K}$:

$$f \sim \mathcal{TP}(\nu, \boldsymbol{\mu}, \mathbf{K}) \,. \tag{2.52}$$

Any finite collection of functions follows a joint multivariate Student-t distribution, where the multivariate density function is given by

$$\mathrm{MVT}(\boldsymbol{y}|\boldsymbol{\mu}, \mathbf{K}, \nu) = \frac{\Gamma\left(\frac{\nu+n}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \frac{1}{((\nu-2)\pi)^{\frac{n}{2}}} |\mathbf{K}|^{-1/2} \left(1 + \frac{1}{\nu-2}(\boldsymbol{y}-\boldsymbol{\mu})^\top \mathbf{K}^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\right)^{-\frac{\nu+n}{2}} \,. \tag{2.53}$$

In TP regression, the posterior mean and covariance are given by

$$\mathbb{E}[f(t_*)] = \boldsymbol{k}_*^\top \mathbf{K}_y^{-1} \boldsymbol{y} \,, \tag{2.54}$$

$$\mathrm{cov}(f(t_*)) = \frac{\nu - 2 + \boldsymbol{y}^\top \mathbf{K}_y^{-1} \boldsymbol{y}}{\nu - 2 + n} (\boldsymbol{k}_{**} - \boldsymbol{k}_*^\top \mathbf{K}_y^{-1} \boldsymbol{k}_*) \,. \tag{2.55}$$

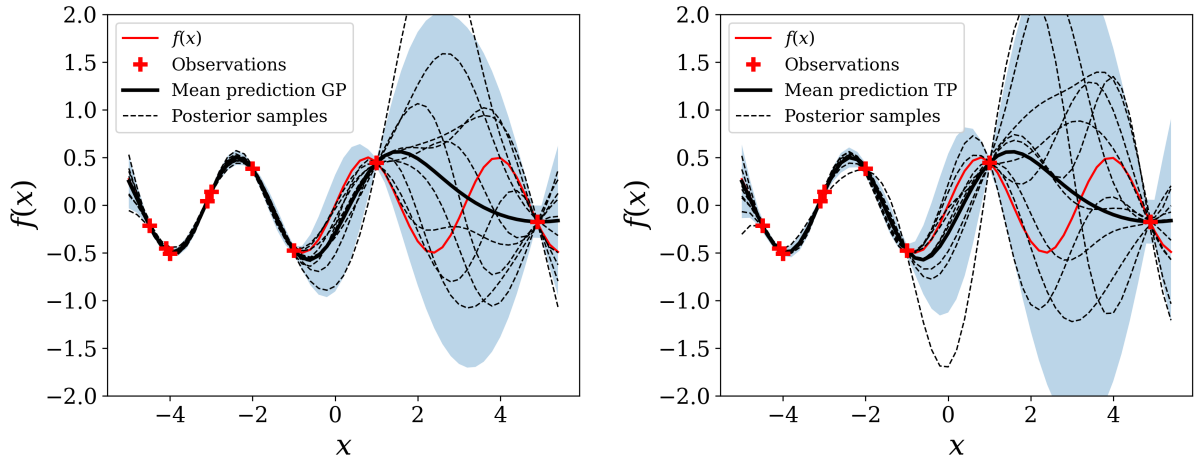
The posterior mean is equivalent to the posterior mean estimated by GP regression (Eq. 2.30), assuming the same hyperparameters. However, the posterior covariance differs from the posterior covariance estimated by a GP (Eq. 2.30) in the leading term and depends on the observations $\boldsymbol{y}$. If the observed data come from an underlying Gaussian distribution, then the squared Mahalanobis distance $\boldsymbol{y}^\top \mathbf{K}_y^{-1} \boldsymbol{y}$ is distributed like a χ^2-distribution with mean $|n|$ (Slotani, 1964). Hence, in this case, the posterior covariance estimated by Eq. 2.55 is approximately the same as estimated with GP regression. However, when the observed data scatter significantly more or less than expected under a Gaussian assumption, the estimated posterior covariance is larger or lower using a TP. The smaller ν is, the bigger is the difference of the estimated covariance compared to GP regression. For increasing ν, the effect becomes smaller and in the limit $\nu \to \infty$, TP regression is equivalent to GP regression (Tracey and Wolpert, 2018).

In Fig. 2.6, a comparison between GP and TP regression for a synthetic test case with 9 observations is shown. For illustration purposes the same hyperparameters ($l = 1.2, \sigma_f^2 = 1.0, \sigma_n^2 = 0.05$) for the squared exponential kernel are used for both regression models. The additional hyperparameter ν in the TP is set to 2.7. As both models use the same hyperparameters, the estimated mean is identical. However, the TP estimates a larger covariance as the observed data scatter more than expected under a Gaussian assumption.

The hyperparameters $\boldsymbol{\theta}$ of the model can be optimized by minimizing the negative log-likelihood,

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) &= -\log\, p(\boldsymbol{y}|\boldsymbol{\theta}, \nu, \mathbf{X}) \\ &= \frac{n}{2}\log((\nu-2)\pi) + \frac{1}{2}\log(|\mathbf{K}|) - \log\,\Gamma\left(\frac{\nu+n}{2}\right) + \log\,\Gamma\left(\frac{\nu}{2}\right) + \frac{\nu+n}{2}\log\left(1 + \frac{\beta}{\nu-2}\right) \,, \end{aligned} \tag{2.56}$$

where $\beta = (\boldsymbol{y}-\boldsymbol{\mu})^\top \mathbf{K}^{-1}(\boldsymbol{y}-\boldsymbol{\mu})$. Since the TP marginal likelihood (Eq. 2.56) differs from the marginal likelihood coming from a GP (Eq. 2.42), different hyperparameters are to be expected.



(a) Gaussian process regression

(b) Student-t process regression

Figure 2.6: Comparison between (a) Gaussian process and (b) Student-t process regression. Both models use the same hyperparameters. Both panels show the true underlying function $f(x) = \sin(x)\cos(x)$ (red solid line), the mean (black solid line) predicted by the respective model along with 10 samples drawn from the inferred posterior distribution after observing 9 noisy data points (depicted as red + with 5% Student-t distributed noise with $\nu = 3$). The shaded regions correspond to two standard deviations at each input point x.

In Rath et al. (2022) in Fig. 1 a synthetic test case with outliers is shown to illustrate the behavior of GPs and TPs in the presence of outliers. The different marginal likelihoods lead to different estimations of the hyperparameters and allow a more robust prediction of the mean using the TP.

2.5 State space formulation

A challenge when working with GPs is the computational complexity in the traditional formulation that scales cubically with the number of observations. However, several approaches exist to reduce the computational complexity by using, e.g., inducing variables in sparse GPs (Hensman et al., 2013; Csató and Opper, 2002), or representing the covariance matrix in Toeplitz form (Zhang et al., 2005).

There is also a different perspective on GPs. It is well established that solutions to linear stochastic differential equations (SDEs) are always GPs (Särkkä, 2013; Särkkä and Solin, 2019; O'Hagan, 1978). This perspective offers new possibilities, especially in analyzing time-series data and solving regression problems using methods from signal processing for solving SDEs, e.g., Kalman filter and Rauch-Tung-Striebel smoother (Särkkä, 2013).

To use Bayesian filtering and smoothing to solve the filtering problem, the GP regression problem has to be reformulated as a time-invariant linear SDE. For several classes of stationary covariance functions, this transformation can be done analytically without any approximations and hence allows a representation of the GP regression as a solution to an m-th order linear SDE. For covariance functions that do not possess a rational spectral density, a simple Taylor series approximation is sufficient to approximate the covariance function (Hartikainen and Sarkka, 2010). The computational complexity scales as $\mathcal{O}(nm^3)$, where n is the number of observations and m the state dimensionality of the linear SDE. As m is typically very small (usually less than 10) and hence the scaling is linear in the number of observations, the state space approach is also very well suited for high-resolution multivariate time series and can also be extended to spatio-temporal models (Wilkinson et al., 2020; Särkkä et al., 2013; Särkkä and Hartikainen, 2012).

The following section will discuss stochastic differential equations and their spectral density. Then, the reformulation of a GP regression problem into its corresponding state space form is explained. The discrete form of linear SDEs is introduced, followed by a short introduction to Bayesian filters and smoothers.

2.5.1 Introduction to Bayesian filtering and smoothing

An m-th order linear SDE is given by

$$a_0 f(t) + a_1 \frac{df(t)}{dt} + \cdots + a_m \frac{d^m f(t)}{dt^m} = w(t)\,, \tag{2.57}$$

where $w(t)$ is a zero-mean white noise process[2] and $a_0, \ldots, a_m$ are known constants. The solution $f(t)$ is a GP, as $w(t)$ is Gaussian, and under linear operations, a GP stays a GP (Adler, 2010, Sec. 2.2).

Rewriting Eq. 2.57 in its state space form, defining a vector-valued function $\boldsymbol{f}(t) = (f(t), df(t)/dt, \ldots, d^{m-1}f(t)/dt^{m-1})$, we get

[2] A white noise process $w(t)$ is usually modeled as Gaussian and $w(t)$ and $w(t')$ are uncorrelated for all $t \neq t'$. The spectral density of a white noise process is constant over all frequencies (Särkkä and Solin, 2019).

$$\frac{d\boldsymbol{f}(t)}{dt} = \underbrace{\begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -a_0 & -a_1 & \dots & -a_{m-1} \end{pmatrix}}_{\mathbf{F}} \boldsymbol{f}(t) + \underbrace{\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}}_{\mathbf{L}} w(t) \,. \tag{2.58}$$

We are still only observing the noise-corrupted values y_k of the first component of $\boldsymbol{f}(t_k)$ at times t_k and can therefore define a measurement model,

$$y_k = \underbrace{\begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}}_{\mathbf{H}} \boldsymbol{f}(t_k) + \varepsilon_k \,, \tag{2.59}$$

where ε_k is i.i.d. Gaussian noise $\varepsilon_k \sim \mathcal{N}(0, \sigma_n^2)$. Combining Eqs. 2.58 and 2.59 allows us to formulate a linear state space model with a linear measurement model

$$\frac{d\boldsymbol{f}(t)}{dt} = \mathbf{F}\boldsymbol{f}(t) + \mathbf{L}\boldsymbol{w}(t) \,, \tag{2.60}$$

$$y_k = \mathbf{H}\boldsymbol{f}(t_k) + \varepsilon_k \,, \tag{2.61}$$

where $k = 1, \dots, T$. Here, $\boldsymbol{f}(t) \in \mathbb{R}^m$ contains m stochastic processes. The feedback matrix $\mathbf{F} \in \mathbb{R}^{m \times m}$ and the noise effect matrix $\mathbf{L} \in \mathbb{R}^{m \times s}$ define the model. Here, we allow a vector of white noise processes $\boldsymbol{w}(t)$ with a spectral density matrix $\mathbf{Q_c} \in \mathbb{R}^{s \times s}$ (Solin, 2016). Generally, the driving white noise process can be multi-dimensional, which generalizes the class of linear time-invariant SDEs (Solin, 2016). For a one-dimensional white noise process $w(t)$ that we use in the following, the spectral density matrix becomes a constant q_c.

To find the corresponding covariance function $C(t)$ of the SDE, we take the Fourier transform of Eq. 2.57 to calculate the spectral density of the process, which is the square of the absolute value of the Fourier transform $\hat{f}(i\omega)$ of $\boldsymbol{f}(t)$:

$$S(\omega) = |\hat{f}(i\omega)|^2 = \hat{f}(i\omega)\hat{f}(-i\omega) \,. \tag{2.62}$$

For general time-invariant linear SDEs of the form given in Eqs. 2.60 and 2.61, we get

$$S(\omega) = \mathbf{H}(\mathbf{F} + i\omega\mathbf{I})^{-1}\mathbf{L}q_c\mathbf{L}[(\mathbf{F} + i\omega\mathbf{I})^{-1}]^\top\mathbf{H}^\top \,. \tag{2.63}$$

The Wiener-Khinchin theorem states that the inverse Fourier transform of the spectral density is the covariance function:

$$C(t, t') = \mathcal{F}^{-1}[S(\omega)] = \frac{1}{2\pi}\int S(\omega)\exp(i\omega t)d\omega \,. \tag{2.64}$$

We now assume that we have a given GP regression problem with covariance function $k(t, t')$ that we want to transform into its corresponding state space form. This means that we want the output $f(t)$ of the m-th order time-invariant linear SDE to have a specific covariance function $C(t, t')$. Hence, we have to find matrices $\mathbf{F}$ and $\mathbf{L}$ and the spectral density of the driving white noise process q_c such that $f(t)$, i.e. the first component of $\boldsymbol{f}(t)$, has the desired covariance function

$C(t, t')$. The spectral density of the covariance function (i.e. the Fourier transform of $C(t)$) must be of the form (Särkkä et al., 2013)

$$S(\omega) = \frac{m\text{-th order polynomial in } \omega^2}{n\text{-th order polynomial in } \omega^2} , \tag{2.65}$$

where $m < n$. If this is not the case, e.g., for the squared exponential covariance function, an approximation with, e.g., Taylor series expansion of the denominator, is needed to get the rational form (Hartikainen and Sarkka, 2010). Then, we need to find a stable[3] transfer function $G(i\omega)$ via spectral factorization[4] of the form

$$G(i\omega) = \frac{b_m(i\omega)^m + b_{m-1}(i\omega)^{m-1} + \cdots + b_1(i\omega) + b_0}{a_n(i\omega)^n + a_{n-1}(i\omega)^{n-1} + \cdots + a_1(i\omega) + a_0} , \tag{2.66}$$

where again $m < n$ and $a_n \neq 0$. This transfer function is needed to rewrite the spectral density as

$$S(\omega) = G(i\omega) q_c G(-i\omega) . \tag{2.67}$$

To build the matrices that define the dynamics $\mathbf{F}$ and $\mathbf{L}$, we bring the transfer function $G(i\omega)$ into the *controller canonical form* (Glad and Ljung, 2000):

$$\frac{d\boldsymbol{f}(t)}{dt} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \dots & 1 \\ -a_0 & -a_1 & \dots & \dots & -a_n \end{pmatrix} \boldsymbol{f}(t) + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} w(t) . \tag{2.68}$$

$$y(t) = \begin{pmatrix} b_0 & b_1 & \dots & b_{n-1} & b_n \end{pmatrix} \boldsymbol{f}(t) . \tag{2.69}$$

For illustration of the procedure outlined above, we use the class of Matérn covariance functions given as

$$k(\tau) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{\tau}{l} \right)^{\nu} K_\nu \left(\sqrt{2\nu} \frac{\tau}{l} \right) , \tag{2.70}$$

where $\tau = t - t'$, $K_\nu(\cdot)$ is the modified Bessel function of the second kind and parameters ν, l, σ control the smoothness, length scale, and magnitude, respectively. We first compute the spectral density by using the Fourier transform of $k(\tau)$:

$$S(\omega) = \mathcal{F}[k(\tau)] = \sigma^2 \frac{2\pi^{1/2}\Gamma(\nu + 1/2)}{\Gamma(\nu)} \lambda^{2\nu} (\lambda^2 + \omega^2)^{-\nu+1/2} , \tag{2.71}$$

where $\lambda = \sqrt{2\nu}/l$. Here, the spectral density is of the desired rational form (Eq. 2.65). The spectral density can be factored as

$$S(\omega) \propto (\lambda + i\omega)^{-(p+1)} (\lambda - i\omega)^{-(p+1)} , \tag{2.72}$$

[3] A stable transfer function $G(i\omega)$ has all poles in the upper half of the complex plane (Särkkä et al., 2013).

[4] One possibility for spectral factorization is the computation of the roots of the numerator and denominator polynomials of $S(\omega)$ that will appear in complex conjugate pairs (Särkkä et al., 2013).

where $\nu = p + 1/2$.

The next step is to use spectral factorization $S(\omega) = G(i\omega)q_c G(-i\omega)$ to find a stable transfer function $G(i\omega)$ and the corresponding spectral density q_c of the driving white noise process, which are in the case of a Matérn function

$$G(i\omega) = (\lambda + i\omega)^{-(p+1)} , \tag{2.73}$$

and

$$q_c = \frac{2\sigma^2 \pi^{1/2} \lambda^{(2p+1)} \Gamma(p+1)}{\Gamma(p+1/2)} . \tag{2.74}$$

Using Eqs. 2.68 and 2.69) we get for the current application case of the Matérn covariance function with $p = 1$ ($\nu = 3/2$):

$$\frac{d\boldsymbol{f}(t)}{dt} = \underbrace{\begin{pmatrix} 0 & 1 \\ -\lambda^2 & -2\lambda \end{pmatrix}}_{\mathbf{F}} \boldsymbol{f}(t) + \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\mathbf{L}} w(t) , \tag{2.75}$$

where the vector valued function $\boldsymbol{f}(t)$ contains the state $f(t)$ and its first derivative $df(t)/dt$. In this case, the state dimensionality m is equal to 2. The spectral density q_c of the white noise process reduces to $q_c = 4\lambda^3\sigma^2$.

The continous-time linear SDE (Eqs. 2.60 and 2.61) is equivalent to the following discrete-time system:

$$\begin{aligned} \boldsymbol{f}(t_{k+1}) &= \mathbf{A}_k \boldsymbol{f}(t_k) + \boldsymbol{q}_k , & (2.76) \\ y_k &= \mathbf{H}_k \boldsymbol{f}(t_k) + \varepsilon_k , & (2.77) \end{aligned}$$

with initial state $\boldsymbol{f}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_0)$ and $\boldsymbol{q}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$. $\mathbf{A}_k$ is the discrete transition matrix between t_k and t_{k+1} and $\mathbf{Q}_k$ is the discrete process noise covariance matrix given by

$$\begin{aligned} \mathbf{A}_k &= \boldsymbol{\Phi}(\Delta t_k) , & (2.78) \\ \mathbf{Q}_k &= \int_0^{\Delta t_k} \boldsymbol{\Phi}(\Delta t_k - \tau) \mathbf{L}\, \mathbf{Q_c}\, \mathbf{L}^\top \boldsymbol{\Phi}(\Delta t_k - \tau)^\top d\tau , & (2.79) \end{aligned}$$

where we use the matrix exponential $\boldsymbol{\Phi}(\tau) = \exp(\mathbf{F}\tau)$ and $\Delta t_k = t_{k+1} - t_k$. The initial state covariance $\mathbf{P}_0$ is the steady-state covariance $\mathbf{P}_\infty$ given by the solution of the Lyapunov equation:

$$\frac{d\mathbf{P}_\infty}{dt} = \mathbf{F}\mathbf{P}_\infty + \mathbf{P}_\infty \mathbf{F}^\top + \mathbf{L}\, \mathbf{Q_c}\, \mathbf{L}^\top = 0 . \tag{2.80}$$

When $\mathbf{P}_\infty$ is known, the process noise covariance matrix can be calculated as (Solin, 2016)

$$\mathbf{Q}_k = \mathbf{P}_\infty - \mathbf{A}_k \mathbf{P}_\infty \mathbf{A}_k^\top . \tag{2.81}$$

The continuous-time and discrete-time system distributions coincide at t_k.

This discrete-time system can now be solved by estimating the joint posterior distribution of the hidden states $\boldsymbol{f}_{0:T}$ given all observed measurements $\boldsymbol{y}_{1:T}$ (Särkkä, 2013):

$$p(\boldsymbol{f}_{0:T}|\boldsymbol{y}_{1:T}) = \frac{p(\boldsymbol{y}_{1:T}|\boldsymbol{f}_{0:T})p(\boldsymbol{f}_{0:T})}{p(\boldsymbol{y}_{1:T})}\,, \tag{2.82}$$

where $p(\boldsymbol{f}_{0:T})$ is the prior distribution, $p(\boldsymbol{y}_{1:T}|\boldsymbol{f}_{0:T})$ is the likelihood of the measurements and $p(\boldsymbol{y}_{1:T})$ is the normalization. With an increasing number of measurements, the dimensionality of the full posterior distribution also increases, making its full calculation for each time step computationally infeasible. Therefore, we restrict the computation to marginal distributions given by the Bayesian filter and smoother:

- Filtering distribution $p(\boldsymbol{f}_k|\boldsymbol{y}_{1:k}) = \mathcal{N}(\boldsymbol{m}_{k|1:k}, \mathbf{P}_{k|1:k})$: marginal distribution of $\boldsymbol{f}_k$ taking the current and previous measurements into account.
- Prediction distribution $p(\boldsymbol{f}_{k+n}|\boldsymbol{y}_{1:k}) = \mathcal{N}(\boldsymbol{m}_{k+n|1:k}, \mathbf{P}_{k+n|1:k})$: prediction of a future state $\boldsymbol{f}_{k+n}$.
- Smoothing distribution $p(\boldsymbol{f}_k|\boldsymbol{y}_{1:T}) = \mathcal{N}(\boldsymbol{m}_{k|1:T}, \mathbf{P}_{k|1:T})$: marginal distribution of $\boldsymbol{f}_k$ given measurements $\boldsymbol{y}_{1:T}$ with $T > k$.

The state space model is defined by a prior probability distribution of the hidden state $\boldsymbol{f}_0$, a dynamic model that describes the dynamics of the systems via a transition probability distribution $p(\boldsymbol{f}_k|\boldsymbol{f}_{k-1})$ and the measurement model giving the conditional probability of the measurement given the hidden state $p(\boldsymbol{y}_k|\boldsymbol{f}_k)$. For linear Gaussian filtering problems, the applicable Bayesian filter and smoother are the Kalman filter and the RTS smoother, providing a closed-form solution of the distribution (Särkkä, 2013; Särkkä and Solin, 2019).

The Kalman filter recursively solves state space models where the dynamic model and the measurement model are linear Gaussian as in Eqs. 2.76 and 2.77. We assume an initial Gaussian distribution $\boldsymbol{f}_0 = \mathcal{N}(\boldsymbol{m}_0, \mathbf{P}_0)$. For each $k = 1, \ldots, T$, the *prediction step* is calculated using

$$\begin{aligned} \boldsymbol{m}_k^- &= \mathbf{A}_{k-1}\boldsymbol{m}_{k-1}\,, & (2.83)\\ \mathbf{P}_k^- &= \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^\top + \mathbf{Q}_{k-1}\,. & (2.84) \end{aligned}$$

If a measurement is available for time step t_k, the prediction is updated via the *update step*:

$$\begin{aligned} \boldsymbol{v}_k &= \boldsymbol{y}_k - \mathbf{H}_k\boldsymbol{m}_k^-\,, & (2.85)\\ \mathbf{S}_k &= \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^\top + \mathbf{R}_k\,, & (2.86)\\ \mathbf{K}_k &= \mathbf{P}_k^-\mathbf{H}_k^\top\mathbf{S}_k^{-1}\,, & (2.87)\\ \boldsymbol{m}_k &= \boldsymbol{m}_k^- + \mathbf{K}_k\boldsymbol{v}_k\,, & (2.88)\\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top\,. & (2.89) \end{aligned}$$

Here, the innovation mean $\boldsymbol{v}_k$ is estimated by using the difference between the observation $\boldsymbol{y}_k$ and the Kalman filter prediction $\boldsymbol{m}_k^-$. Similarly, the innovation covariance $\mathbf{S}_k$ is calculated with $\mathbf{R}$, the observation noise. $\mathbf{K}_k$ is the Kalman gain used to update the mean and covariance prediction. Whenever no measurement is available, we skip the update step.

The corresponding smoothing algorithm is the RTS smoother, which computes the smoothing distribution recursively backwards, taking all measurements into account. Here, we start with the filter output at $k = T$, initialize the smoother with $(\boldsymbol{m}_T, \mathbf{P}_T)$ and move backwards to $k = 0$:

$$\begin{aligned}
\boldsymbol{m}_{k+1}^- &= \mathbf{A}_k \boldsymbol{m}_k \,, && (2.90)\\
\mathbf{P}_{k+1}^- &= \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^\top + \mathbf{Q}_k \,, && (2.91)\\
\mathbf{G}_k &= \mathbf{P}_k \mathbf{A}_k^\top [\mathbf{P}_{k+1}^-]^{-1} \,, && (2.92)\\
\boldsymbol{m}_k^s &= \boldsymbol{m} + \mathbf{G}_k (\boldsymbol{m}_{k+1}^s - \boldsymbol{m}_{k+1}^-) \,, && (2.93)\\
\mathbf{P}_k^s &= \mathbf{P}_k + \mathbf{G}_k (\mathbf{P}_{k+1}^s - \mathbf{P}_{k+1^-}) \mathbf{G}_k^\top \,, && (2.94)
\end{aligned}$$

where $\mathbf{G}_k$ is the smoother gain.

While the prediction of the Kalman filter is an optimal estimate of the underlying state at each time step with respect to the available information, the solution provided by the RTS smoother is conditioned on all observations. This allows a refinement of the prediction from the Kalman filter, which is subject to fluctuations, especially when working with noisy data. The RTS smoother also considers future observations and this allows more stable and smoothed estimates as underlying trends can be distinguished from random fluctuations. The smoothing solution is identical to the prediction of mean and covariance using GP regression with the same covariance function and hyperparameters used for building the state space model.

To include the prediction of test points $f(t_*)$, they are included in the filtering algorithm and the smoothing algorithm. The prediction for mean and covariance at the test point t_* is then calculated by using the smoother output:

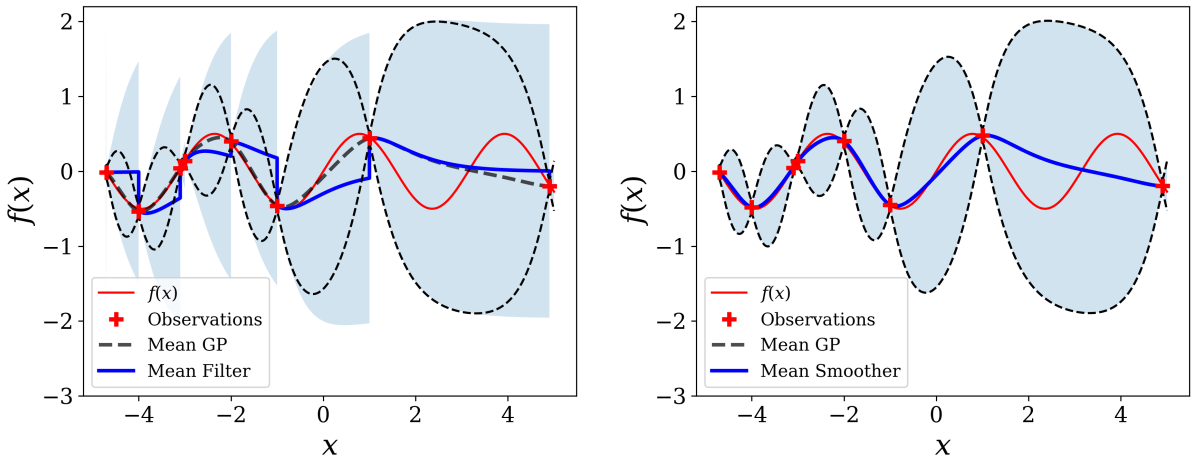
$$\begin{aligned}
f(t_*) &= \mathbf{H} \boldsymbol{m}_*^s \,, && (2.95)\\
\mathrm{cov}[f(t_*)] &= \mathbf{H} \mathbf{P}_*^s \mathbf{H}^\top \,. && (2.96)
\end{aligned}$$

In contrast to traditional GP regression that scales cubically with the number of observations, the Kalman filter and RTS smoother allow inference in linear time.

In Fig. 2.7, the predictive filtering and smoothing distributions for a synthetic test case in direct comparison with usual GP regression are shown. Here, a Matérn kernel with $\nu = 3/2$ and hyperparameters $l = 1$, $\sigma_f^2 = 1$, $\sigma_n = 0.05$ is used. The covariance function with the same hyperparameters is transformed using its spectral density to build the corresponding linear SDE model (see Eq. 2.75). In Fig. 2.7a, one can see that the filtering solution is a sequential solution that is corrected whenever a measurement is observed. The marginal variance grows until there is a new observation and is corrected with the innovation covariance, taking the observation noise into account. The smoother solution depicted in Fig. 2.7b coincides with the solution obtained via GP regression up to machine precision.

Similar to GP regression, the parameters $\boldsymbol{\theta}$ of the state space model have to be estimated. Here, we minimize the negative log-likelihood, where all necessary computations are available as a side product in the filtering algorithm (Särkkä, 2013):

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_k \left(-\frac{1}{2} \log |2\pi \mathbf{S}_k| - \frac{1}{2} \boldsymbol{v}_k^\top \mathbf{S}_k^{-1} \boldsymbol{v}_k \right) \,. \qquad (2.97)$$



(a) Filtering solution

(b) Smoothing solution

Figure 2.7: Comparison between usual GP regression and the sequential solution via Kalman filter (a) and RTS smoother (b) for 8 observations with 5% Gaussian noise from the underlying function $f(x) = \sin(x)\cos(x)$. The 95% confidence band calculated by Kalman filtering and RTS smoothing is depicted as a shaded area, while the results coming from GP regression are shown as dashed lines.

For optimization with gradient information, the corresponding gradients can be computed straightforwardly by differentiating every equation in the Kalman filter algorithm. Then these get evaluated in the optimization routine, besides the standard Kalman filter equations.

2.5.2 Filtering and smoothing for TP regression

Similar to GPs (as described in Section 2.5), there also exists a state space formulation for Student-t processes (Solin and Särkkä, 2015). Based on the analytical transformation of certain classes of covariance functions into solutions of m-th order linear SDEs, a state space Student-t process can be constructed in the form of Eq. 2.60 and 2.61. Here, a scale mixture of state space form SDEs is needed. This is achieved by setting the spectral density to $\gamma\mathbf{Q}_c$ and using the initial state $\boldsymbol{f}(0) \sim \mathcal{N}(\mathbf{0}, \gamma\mathbf{P}_0)$, where γ is a random variable distributed according to the inverse gamma distribution $\gamma \sim \mathcal{IG}(\nu/2, (\nu-2)/2)$. This can be understood when defining a Student-t process as a scale mixture of Gaussians: we consider a Gaussian with mean $\boldsymbol{\mu}$ and covariance $\mathbf{K}$ and scale the covariance of the Gaussian with γ. Then, the scale mixture form of the probability density becomes equivalent to the Student-t density. This gives the same result as placing an inverse Wishart process prior on the kernel function of a GP (Shah et al., 2014).

The sequential inference algorithm is very similar to the Kalman filter and RTS smoother with the difference of the scaling factor. However, an additional step in the filter update is necessary to estimate the scaling factor γ_k. Below, differences to the Kalman filter and RTS smoother are highlighted in red.

The filtering prediction distribution gives the next state based on previous observations,

$$\boldsymbol{m}_k^- = \mathbf{A}_{k-1}\boldsymbol{m}_{k-1}\,, \tag{2.98}$$
$$\mathbf{P}_k^- = \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^\top + \gamma_{k-1}\mathbf{Q}_{k-1}\,. \tag{2.99}$$

The filter update is needed whenever there is a measurement at time step t_k:

$$\boldsymbol{v}_k = \boldsymbol{y}_k - \mathbf{H}_k\boldsymbol{m}_k^-\,, \tag{2.100}$$
$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^\top + \mathbf{R}_k\,, \tag{2.101}$$
$$\gamma_k = \frac{\gamma_{k-1}}{\nu_k - 2}(\nu_{k-1} - 2 + \boldsymbol{v}_k^\top\mathbf{S}_k^{-1}\boldsymbol{v}_k)\,, \tag{2.102}$$
$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^\top\mathbf{S}_k^{-1}\,, \tag{2.103}$$
$$\boldsymbol{m}_k = \boldsymbol{m}_k^- + \mathbf{K}_k\boldsymbol{v}_k\,, \tag{2.104}$$
$$\mathbf{P}_k = \frac{\gamma_k}{\gamma_{k-1}}(\mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top)\,. \tag{2.105}$$

The filter is initialized using $\nu_0 = \nu$ and $\gamma_0 = 1$. ν is updated whenever there is a filter update with $\nu_k = \nu_{k-1} + 1$. The RTS smoother becomes

$$\boldsymbol{m}_{k+1}^- = \mathbf{A}_k\boldsymbol{m}_k\,, \tag{2.106}$$
$$\mathbf{P}_{k+1}^- = \mathbf{A}_k\mathbf{P}_k\mathbf{A}_k^\top + \gamma_k\mathbf{Q}_k\,, \tag{2.107}$$
$$\mathbf{G}_k = \mathbf{P}_k\mathbf{A}_k^\top[\mathbf{P}_{k+1}^-]^{-1}\,, \tag{2.108}$$
$$\boldsymbol{m}_k^s = \boldsymbol{m} + \mathbf{G}_k(\boldsymbol{m}_{k+1}^s - \boldsymbol{m}_{k+1}^-)\,, \tag{2.109}$$
$$\mathbf{P}_k^s = \frac{\gamma_n}{\gamma_k}(\mathbf{P}_k + \mathbf{G}_k(\mathbf{P}_{k+1}^s - \mathbf{P}_{k+1^-})\mathbf{G}_k^\top)\,. \tag{2.110}$$

The marginal log-likelihood is directly available from the filtering algorithm and can be used to optimize the hyperparameters $\boldsymbol{\theta}$:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{k=1}^{n}\left[\frac{1}{2}\log((\nu-2)\pi) + \frac{1}{2}\log(|\mathbf{S}_k|) + \log\Gamma\left(\frac{\nu_{k-1}}{2}\right)\right.$$
$$\left. -\log\Gamma\left(\frac{\nu_k}{2}\right) + \frac{1}{2}\log\left(\frac{\nu_{k-1}-2}{\nu-2}\right) + \frac{\nu_k}{2}\log\left(1 + \frac{\boldsymbol{v}_k^\top\mathbf{S}_k^{-1}\boldsymbol{v}_k}{\nu_{k-1}-2}\right)\right]\,, \tag{2.111}$$

where $\boldsymbol{v}_k$ and $\mathbf{S}_k$ are the innovation mean and covariance.

Equivalently to GP and TP regression, Kalman and Student-t filters give the same mean prediction when using the same hyperparameters. They differ, however, in the estimation of the state covariance.

Contributions Section 3.4 and 3.5 present the application of TP regression for data augmentation for disruption prediction and data imputation in plasma diagnostics (Rath et al., 2022, 2023). In both cases, we work with very few labeled multivariate time series measurements that we aim to augment to produce a comprehensive training database for training large machine learning models for disruption prediction. In addition to the small training data set, there are different causes for disruptions, making the data set highly imbalanced. In Rath et al. (2022), we use state space Student-t process regression via Bayesian filtering to overcome challenges posed by possible outliers and noise in the training data set. The state space formulation reduces the computational complexity allowing inference in linear time, and the model is thus also usable for high-resolution time series. Here, we work with a multi-output state space model that, in the first step, neglects signal interdependencies. All dimensions use a Matérn 3/2 kernel but have their own set of hyperparameters to handle dynamics on different time scales. Due to the limited training data, hyperparameter optimization for all dimensions is practically unfeasible. However, in a post-processing step, signal correlations and cross-correlations are introduced via coloring transformations.

Our data augmentation approach is based on working with several fast local models – one for each disruption class with similar operating conditions – that estimate the posterior distribution from a small set of multivariate time series. We apply the model to three different disruption types and use 5 signals with which we want to produce data for training a neural network for disruption prediction. In general, the method is not limited to a specific number or types of signal. After a data preprocessing step, where signals from different measurements are aligned according to their end time and missing data points are interpolated linearly, each local model is trained by minimizing the negative log-likelihood and the predictive posterior is estimated using a Student-t filter and corresponding smoother. From the estimated posterior for each class, we draw samples and perform coloring transformations to account for signal interdependencies. Different methods from time series analysis, statistics, and clustering are used to assess the quality of the generated data and evaluate whether the generated samples are indistinguishable from the original data to be used in a comprehensive training database. For all three test cases, we find that the artificially generated data resembles the original data sufficiently well within the scope of the used metrics.

The contributing article presented in Section 3.5 focuses on imputing gappy data due to sensor failures or non-converging calculation routines. While this issue was addressed in Rath et al. (2022) via linear interpolation, we now use the correlations between input signals to reconstruct missing data points. Here, we include correlations directly in the surrogate model. Again, we use a state space Student-t process but employ a Matérn cross-covariance kernel. Two synthetic test cases are considered to evaluate the performance, where the latter is inspired by two correlated flux loop signals during an edge localized mode. Two input signals are considered in both test cases, and measurements are removed from one signal to artificially create missing values. We compare the performance of the proposed dependent model with an independent model. In both cases, the predicted mean of the signal with missing values is better captured and more accurately reproduced by the dependent model as correlations are taken into account. This model can also be used for data augmentation to generate quasi-realistic training data. Its performance is evaluated by assessing the distributions of generated and original data using metrics from time series analysis and statistics.

3 Contributions

3.1 Gaussian Process Regression for Data Fulfilling Linear Differential Equations with Localized Sources

Main novelty:
The paper introduces Gaussian processes with specialized kernels to exactly fulfill linear partial differential equations with localized sources.

Contributing article:
Albert, C. G. and Rath, K. (2020). Gaussian Process Regression for Data Fulfilling Linear Differential Equations with Localized Sources. *Entropy*, 22(2)

Author contributions:
Christopher Albert devised the project, implemented the code and drafted the initial version. Katharina Rath added the hyperparameter optimization, comparisons to existing methods and the estimations of the source positions and helped to finalize the manuscript.

entropy

MDPI

Article

Gaussian Process Regression for Data Fulfilling Linear Differential Equations with Localized Sources †

Christopher G. Albert [1,*] and Katharina Rath [1,2]

[1] Max-Planck-Institut für Plasmaphysik, Boltzmannstr. 2, 85748 Garching, Germany; katharina.rath@ipp.mpg.de
[2] Department of Statistics, Ludwig-Maximilians-Universität München, Ludwigstr. 33, 80539 Munich, Germany
* Correspondence: albert@alumni.tugraz.at
† This paper is an extended version of our paper published in MaxEnt 2019.

Received: 15 December 2019; Accepted: 24 January 2020; Published: 27 January 2020

Abstract: Specialized Gaussian process regression is presented for data that are known to fulfill a given linear differential equation with vanishing or localized sources. The method allows estimation of system parameters as well as strength and location of point sources. It is applicable to a wide range of data from measurement and simulation. The underlying principle is the well-known invariance of the Gaussian probability distribution under linear operators, in particular differentiation. In contrast to approaches with a generic covariance function/kernel, we restrict the Gaussian process to generate only solutions of the homogeneous part of the differential equation. This requires specialized kernels with a direct correspondence of certain kernel hyperparameters to parameters in the underlying equation and leads to more reliable regression results with less training data. Inhomogeneous contributions from linear superposition of point sources are treated via a linear model over fundamental solutions. Maximum likelihood estimates for hyperparameters and source positions are obtained by nonlinear optimization. For differential equations representing laws of physics the present approach generates only physically possible solutions, and estimated hyperparameters represent physical properties. After a general derivation, modeling of source-free data and parameter estimation is demonstrated for Laplace's equation and the heat/diffusion equation. Finally, the Helmholtz equation with point sources is treated, representing scalar wave data such as acoustic pressure in the frequency domain.

Keywords: Gaussian process regression; physics-informed methods; kernel methods; field reconstruction; source localization; partial differential equations; meshless methods

1. Introduction

The larger context of the present work is the goal to construct reduced complexity models as emulators or surrogates that retain mathematical and physical properties of the underlying system. In recent terminology, such approaches are examples of "physics informed machine learning". Similar to usual numerical models, the aim here is to represent infinite systems by exploiting finite information in some optimal sense. In the spirit of structure preserving numerics, one tries to move errors to the "right place" to retain laws such as conservation of mass, energy, or momentum. Here, we treat data known to fulfill a given linear differential equation. This article is an extended version of a conference paper [1] presented at the MaxEnt workshop 2019. The revised text adds hyperparameter optimization, results for the heat equation and detailed comparisons to existing methods.

This article deals with Gaussian process (GP) regression on data with additional information known in the form of linear, generally partial differential equations (PDEs). An illustrative example is

the reconstruction of an acoustic sound pressure field and source parameters from discrete microphone measurements. GPs, a special class of random fields, are used in a probabilistic rather than a stochastic sense: estimate a fixed but unknown field from possibly noisy local measurements. Uncertainties in this reconstruction are modeled by a normal distribution.

Using GPs to fit data from PDEs has been a topic of research for some time, especially in the field of geostatistics [2]. A general analysis for deterministic source densities including a number of important properties is given by [3]. In these earlier works GPs are usually referred to as "Kriging" and covariance functions/kernels as "covariograms". A number of more recent works from various fields [4–8] use the linear operator of a PDE to relate the kernels of source and response field. One of the two is usually modeled by a generic squared exponential kernel. Although the authors of [4,6,7] use such a kernel for the response field and a kernel modified by a differential operator for the source field, [5] models the source field by a generic kernel and applies the inverse (integral) operator to obtain a kernel for the measured response. In contrast to the present approach such methods are suited best for source fields that are non-vanishing across the whole domain. In terms of deterministic numerical methods, one could say that these approaches with volumetric charge densities correspond to meshless variants of the finite element method (FEM).

The approach in the present work instead relies on Gaussian processes that generate *exact* solutions of the homogeneous part of the differential equation [9–11]. This is efficient for problems with mostly source-free domains and requires specialized kernels where possible singularities (virtual sources) are moved outside the domain of interest. In particular, boundary conditions on a finite domain can be either supplied or reconstructed in this fashion. Localized internal point sources are then superimposed as a linear model, using again fundamental solutions in the free field. One can thus interpret this approach as a probabilistic variant of a procedure related to the boundary element method (BEM), known as the *method of fundamental solutions (MFS)* or regularized BEM [12–14]. As in the BEM, the MFS also builds on fundamental solutions, but allows to place sources outside the boundary rather than localizing them on a layer. Thus, the MFS avoids singularities in boundary integrals of the BEM, while retaining a similar ratio of numerical effort and accuracy for smooth solutions. To the best of the author's knowledge, the probabilistic variant of the MFS via GPs has first been introduced by [9] to solve the boundary value problem of the Laplace equation and dubbed *Bayesian boundary elements estimation method ((BE)^{2}M)*. This work also provides a detailed treatment of kernels for the 2D Laplace equation. A more extensive and general treatment of the Bayesian context as well as kernels and their connection to fundamental solutions is available in [10] under the term *probabilistic meshless methods (PMM)*.

Although the authors of [9] treat boundary data of a the homogeneous Laplace equation and the authors of [10] provides a detailed mathematical foundation, the present work aims to extend the recent work on added point sources in [11], unify the derivation of specialized kernels, and demonstrate usefulness in applications. First, a general derivation is given on how to model PDE data by superposing a GP and a linear model for localized sources. Then, the construction of kernels for the homogeneous part of partial differential equations via according fundamental solutions is described in general. Finally, concrete application examples are given for Laplace/Poisson, heat/diffusion and Helmholtz equation for which the derivation of several kernels is presented. Performance is compared to regression with a generic squared exponential kernel, including hyperparameter optimization in all cases. For the Helmholtz equation we estimate strength and positions of sources by nonlinear optimization.

2. GP Regression for Data from Linear PDEs

Gaussian process (GP) regression [15] is a tool to represent and update incomplete information on scalar fields $u(\mathbf{x})$, i.e., a real number u depending on a (multidimensional) independent variable $\mathbf{x}$

(the more general case of complex valued fields and vector fields is left open for future investigations in this context). A GP with mean $m(\mathbf{x})$ and covariance function or kernel $k(\mathbf{x}, \mathbf{x}')$ is denoted as

$$u(\mathbf{x}) \sim \mathcal{G}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{1}$$

The choice of an appropriate kernel $k(\mathbf{x}, \mathbf{x}')$ restricts realizations of (1) to respect regularity properties of $u(\mathbf{x})$ such as continuity or characteristic length scales. Often regularity of u does not appear by chance, but rather reflects an underlying law. We will exploit such laws in the construction and application of GPs describing u for the case described by linear (partial) differential equations:

$$\hat{L}u(\mathbf{x}) = q(\mathbf{x}). \tag{2}$$

where $\hat{L}$ is a linear differential operator and $q(\mathbf{x})$ is a source term. In the laws of physics, dimensions of $\mathbf{x}$ usually consist of space and/or time. Physical scalar fields u include, e.g., electrostatic potential Φ, temperature T, or pressure p. Corresponding laws include Gauss' law of electrostatics for Φ with weighted Laplacian $\hat{L} = \varepsilon\Delta$, thermodynamics for T with heat/diffusion operator $\hat{L} = \frac{\partial}{\partial t} - D\Delta$ and frequency-domain acoustics for p with Helmholtz operator $\hat{L} = \Delta + k_0^2$. These operators contain free parameters, namely, permeability ε, wavenumber k_0, and diffusivity D, respectively. While ε may be absorbed inside q in a uniform material model of electrostatics, estimation of parameters such as D or k_0 is useful for material characterization.

Consider first the source-free (homogeneous) case

$$\hat{L}u_h(\mathbf{x}) = 0. \tag{3}$$

An unknown field $u_h(\mathbf{x})$ that fulfills (3) shall be modeled by the Gaussian process

$$u_h(\mathbf{x}) \sim \mathcal{G}(0, k(\mathbf{x}, \mathbf{x}')). \tag{4}$$

Application of a linear operator $\hat{L}$ yields a modified Gaussian process

$$\hat{L}u_h(\mathbf{x}) \sim \mathcal{G}(0, \hat{L}k(\mathbf{x}, \mathbf{x}')\hat{L}'), \tag{5}$$

where $\hat{L}'$ acts from the right side with respect to $\mathbf{x}'$. In order to fulfill (3) we require (5) to vanish identically, i.e., yield a deterministic zero. Consequently, the kernel $k(\mathbf{x}, \mathbf{x}')$ needs to satisfy

$$\hat{L}k(\mathbf{x}, \mathbf{x}')\hat{L}' = 0. \tag{6}$$

A discussion on derivation of such kernels is found in Section 2.

For the general case (2), with unknown source density $q(\mathbf{x})$, we introduce a linear model

$$q(\mathbf{x}) = \sum_i \varphi_i(\mathbf{x}) q_i = \boldsymbol{\varphi}^T(\mathbf{x})\mathbf{q}, \tag{7}$$

with basis functions $\varphi_i(\mathbf{x})$ and a normally distributed prior

$$\mathbf{q} \sim \mathcal{N}(\mathbf{q}_0, \Sigma_q), \tag{8}$$

with mean $\mathbf{q}_0$ and prior covariance Σ_q for coefficients q_i representing source strengths.

For a particulary solution $u_p(\mathbf{x})$ fulfilling the inhomogeneous Equation (2) with source model (8), a linear model induced by the operator $\hat{L}$ follows as

$$u_p(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T\mathbf{q}, \text{ with } \hat{L}h_i(\mathbf{x}) = \varphi_i(\mathbf{x}). \tag{9}$$

Here, coefficients q_i remain the same as in (8) and new basis functions $h_i(\mathbf{x})$ fulfil the differential equation with source density $\varphi_i(\mathbf{x})$. In case of point monopole sources $\varphi_i(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_i^q)$ placed at positions $\mathbf{x}_i^q$, each $h_i(\mathbf{x})$ represents a fundamental solution evaluated for the respective source, so

$$h_i(\mathbf{x}) = G(\mathbf{x}, \mathbf{x}_i^q), \tag{10}$$

where $G(\mathbf{x}, \mathbf{x}_i^q)$ is a Green's function for operator $\hat{L}$. In the remaining work with localized sources we take this approach. As $G(\mathbf{x}, \mathbf{x}_i^q)$ is usually only available for simple geometries and boundary conditions the discussed linear model alone is limited in its application. We can however represent much more general fields by a superposition of a locally source-free background $u_h(\mathbf{x})$ and point source contributions $u_p(\mathbf{x})$. Boundary conditions induced by external sources are then covered by $u_h(\mathbf{x})$, and internal sources entering $u_p(\mathbf{x})$ are treated via simple free-field Green's functions. Following the technique of [16] discussed in [15] (Chapter 2.7), the superposition $u(\mathbf{x}) = u_h(\mathbf{x}) + u_p(\mathbf{x})$ of the GP $u_h(\mathbf{x})$ and the linear model $u_p(\mathbf{x})$ is distributed according to the Gaussian process

$$u(\mathbf{x}) \sim \mathcal{G}(\mathbf{h}(\mathbf{x})^T \mathbf{q}_0, k(\mathbf{x}, \mathbf{x}') + \mathbf{h}(\mathbf{x})^T \Sigma_q \mathbf{h}(\mathbf{x}')). \tag{11}$$

We will now verify that (11) indeed models the original differential Equation (2) correctly, thereby generalizing the analysis for a deterministic source density in [3]. With $\hat{L}k(\mathbf{x}, \mathbf{x}')\hat{L}' = 0$, we obtain

$$\hat{L}u(\mathbf{x}) \sim \mathcal{G}(\hat{L}\mathbf{h}(\mathbf{x})^T \mathbf{q}_0, \hat{L}\mathbf{h}(\mathbf{x})^T \Sigma_q \mathbf{h}(\mathbf{x}')\hat{L}') = \mathcal{G}(\boldsymbol{\varphi}(\mathbf{x})^T \mathbf{q}_0, \boldsymbol{\varphi}(\mathbf{x})^T \Sigma_q \boldsymbol{\varphi}(\mathbf{x})). \tag{12}$$

This is indeed the GP representing the linear source model (8) that we assumed and yields a consistent representation of $u(\mathbf{x})$ and $q(\mathbf{x})$ inside (2).

Using the limit of a vague prior with $\mathbf{q}_0 = 0$ and $|\Sigma_q^{-1}| \to 0$, i.e., minimum information / infinite prior covariance [15,16], posteriors for mean $\bar{u}$ and covariance matrix $\mathrm{cov}(u, u)$ based on given training data $\mathbf{y} = u(X) + \sigma_\mathrm{n}$ with measurement noise variance σ_n^2 are

$$\bar{u}(X_\star) = K_\star^T K_y^{-1}(\mathbf{y} - H^T \tilde{\mathbf{q}}) + H_\star^T \tilde{\mathbf{q}} = K_\star^T K_y^{-1} \mathbf{y} + R^T \tilde{\mathbf{q}}, \tag{13}$$

$$\mathrm{cov}(u(X_\star), u(X_\star)) = K_{\star\star} - K_\star^T K_y^{-1} K_\star + R^T (H K_y^{-1} H^T)^{-1} R. \tag{14}$$

where $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N)$ contains the training points and $X_\star = (\mathbf{x}_{\star 1}, \mathbf{x}_{\star 2}, \ldots, \mathbf{x}_{\star N_\star})$ the evaluation or test points. Functions of X and $X^\star$ are to be understood as vectors or matrices resulting from evaluation at different positions, i.e., $\bar{u}(X_\star) \equiv (\bar{u}(\mathbf{x}_{\star 1}), \bar{u}(\mathbf{x}_{\star 2}), \ldots, \bar{u}(\mathbf{x}_{\star N_\star}))$ is a tuple of predicted expectation values. The matrix $K \equiv k(X, X)$ is the covariance of the training data with entries $K_{ij} \equiv k(\mathbf{x}_i, \mathbf{x}_j)$. Entries of the predicted covariance matrix for u evaluated at the test points $\mathbf{x}_{\star i}$ are $\mathrm{cov}(u(X_\star), u(X_\star))_{ij} \equiv \mathrm{cov}(u(\mathbf{x}_{\star i}), u(\mathbf{x}_{\star j}))$. Furthermore, $K_y \equiv K + \sigma_\mathrm{n}^2 I$, $K_\star \equiv k(X, X_\star)$, $K_{\star\star} \equiv k(X_\star, X_\star)$, $R \equiv H_\star - H K_y^{-1} K_\star$, and entries of H are $H_{ij} \equiv h_i(\mathbf{x}_j)$, $H_{\star ij} \equiv h_i(\mathbf{x}_{\star j})$. Posterior mean and covariance of source strengths are given from the linear model [16] in the limit of a vague prior,

$$\tilde{\mathbf{q}} = (H K_y^{-1} H^T)^{-1} H K_y^{-1} \mathbf{y}, \tag{15}$$

$$\mathrm{cov}(\mathbf{q}, \mathbf{q}) = (H K_y^{-1} H^T)^{-1}. \tag{16}$$

In the absence of sources, the matrix R vanishes, and (13) and (14) reduce to posteriors of a GP with zero prior mean and are directly used to model homogeneous solutions $u_h(\mathbf{x})$ of (3).

Construction of Kernels for Homogeneous PDEs

For the representation of solutions $u_h(\mathbf{x})$ of homogeneous differential Equations (3), the weight-space view ([15] Chapter 2.1) of Gaussian process regression is useful. There the kernel k

is represented via a tuple $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots)$ of basis functions $\phi_i(\mathbf{x})$ that underlie a linear regression model

$$u(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^T \mathbf{w} = \sum_i \phi_i(\mathbf{x}) w_i. \tag{17}$$

Bayesian inference starting from a Gaussian prior with covariance matrix Σ_p for weights $\mathbf{w}$ yields a Mercer kernel

$$k(\mathbf{x}, \mathbf{x}') \equiv \boldsymbol{\phi}^T(\mathbf{x}) \Sigma_\mathrm{p} \boldsymbol{\phi}(\mathbf{x}') = \sum_{i,j} \phi_i(\mathbf{x}) \Sigma_\mathrm{p}^{ij} \phi_j(\mathbf{x}'). \tag{18}$$

The existence of such a representation is guaranteed by Mercer's theorem in the context of reproducing kernel Hilbert spaces (RKHS) [14]. More generally one can also define kernels on an uncountably infinite number of basis functions in analogy to (17) via

$$f(\mathbf{x}) = (\hat{\phi} w)(\mathbf{x}) = \langle \phi(\mathbf{x}, \boldsymbol{\zeta}), w(\boldsymbol{\zeta}) \rangle = \int \phi(\mathbf{x}, \boldsymbol{\zeta})\, w(\boldsymbol{\zeta})\, \mathrm{d}\boldsymbol{\zeta}, \tag{19}$$

where $\hat{\phi}$ is a linear operator acting on elements $w(\boldsymbol{\zeta})$ of an infinite-dimensional weight space parametrized by an auxiliary index variable $\boldsymbol{\zeta}$, that may be multidimensional. We represent $\hat{\phi}$ via an inner product $\langle \phi(\mathbf{x}, \boldsymbol{\zeta}), w(\boldsymbol{\zeta}) \rangle$ in the respective function space given by an integral over $\boldsymbol{\zeta}$. The infinite-dimensional analog to the prior covariance matrix is a prior covariance operator $\hat{\Sigma}_\mathrm{p}$ that defines the kernel as a bilinear form

$$k(\mathbf{x}, \mathbf{x}') \equiv \left\langle \phi(\mathbf{x}, \boldsymbol{\zeta}), \hat{\Sigma}_\mathrm{p} \phi(\mathbf{x}', \boldsymbol{\zeta}') \right\rangle \equiv \int \phi(\mathbf{x}, \boldsymbol{\zeta}) \Sigma_\mathrm{p}(\boldsymbol{\zeta}, \boldsymbol{\zeta}') \phi(\mathbf{x}', \boldsymbol{\zeta}')\, \mathrm{d}\boldsymbol{\zeta}\, \mathrm{d}\boldsymbol{\zeta}'. \tag{20}$$

Kernels of the form (20) are known as convolution kernels. Such a kernel is at least positive semidefinite, and positive definiteness follows in the case of linearly independent basis functions $\phi(\mathbf{x}, \boldsymbol{\zeta})$ [14].

For treatment of PDEs, the possible choices of index variables in Equation (18) or Equation (20) include separation constants of analytical solutions, or the frequency variable of an integral transform. In accordance with [10], using basis functions that satisfy the underlying PDE, a probabilistic meshless method (PMM) is constructed. In particular, if $\boldsymbol{\zeta}$ parameterizes positions of sources, and $\phi(\mathbf{x}, \boldsymbol{\zeta}) = G(\mathbf{x}, \boldsymbol{\zeta})$ in (20) is chosen to be a fundamental solution/Green's function $G(\mathbf{x}, \boldsymbol{\zeta})$ of the PDE, one may call the resulting scheme a *probabilistic method of fundamental solutions (pMFS)*. In [10], sources are placed across the whole computational domain, and the resulting kernel is called *natural*. Here, we will instead place sources in the exterior to fulfill the homogeneous interior problem, as in the classical MFS [12–14]. Technically, this is also achieved by setting $\Sigma_\mathrm{p}(\boldsymbol{\zeta}, \boldsymbol{\zeta}') = 0$ for either $\boldsymbol{\zeta}$ or $\boldsymbol{\zeta}'$ lies in the interior. For discrete sources localized at $\boldsymbol{\zeta} = \boldsymbol{\zeta}_i$ one obtains again discrete basis functions $\phi_i(\mathbf{x}) = G(\mathbf{x}, \boldsymbol{\zeta}_i)$ for (18).

3. Application Cases

Here, the general results described in the previous sections are applied to specific equations. First, a specialized kernel fulfilling the given linear differential equation is constructed according to (18), and second, numerical experiments on physical examples are performed comparing the specialized kernel to a squared exponential kernel. Regression is performed based on values measured at a set of sampling points $\mathbf{x}_i$ and may also include optimization of hyperparameters $\boldsymbol{\theta}$ appearing as auxiliary variables inside the kernel $k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$. The optimization step is, as usually, performed such that the marginal likelihood of the GP is maximized (maximum likelihood or ML values). In the Bayesian sense, this corresponds to a maximum a-posteriori (MAP) estimate for a flat prior. Accordingly, $\boldsymbol{\theta}_\mathrm{ML}$ is fixed rather than providing a joint probability distribution function including $\boldsymbol{\theta}$ as random variables. We note that depending on the setting this choice may lead to underestimation of uncertainties in the reconstruction of $u(\mathbf{x})$, in particular for sparse, low-quality measurements.

3.1. Laplace's Equation in Two Dimensions

First, we explore construction of kernels fulfilling (5) for a homogeneous problem in a finite and infinite dimensional index space, depending on the mode of separation. Consider Laplace's equation:

$$\Delta u(\mathbf{x}) = 0. \tag{21}$$

In contrast to the Helmholtz equation, Laplace's equation has no scale, i.e., permits all length scales in the solution. In the 2D case using polar coordinates the Laplacian becomes

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u(r,\theta)}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u(r,\theta)}{\partial \theta^2} = 0. \tag{22}$$

A well-known family of solutions for this problem based on the separation of variables is

$$u(r,\theta) = r^{\pm m} e^{\pm i m\theta}, \tag{23}$$

with separation constant m, leading to real-valued combinations

$$r^m \cos(m\theta),\ r^m \sin(m\theta),\ r^{-m}\cos(m\theta),\ r^{-m}\sin(m\theta). \tag{24}$$

As our aim is to work in bounded regions, we discard the solutions with negative exponent that diverge at $r = 0$. Choosing a diagonal prior that weights sine and cosine terms equivalently [9] and introducing a length scale ℓ as a free parameter we obtain a kernel according to (18) with

$$\begin{aligned} k(\mathbf{x},\mathbf{x}';\ell,\sigma_m) &= \sum_{m=0}^{\infty}\left(\frac{rr'}{\ell^2}\right)^m \sigma_m^2\left(\cos(m\theta)\,\cos(m\theta') + \sin(m\theta)\,\sin(m\theta')\right) \\ &= \sum_{m=0}^{\infty}\left(\frac{rr'}{\ell^2}\right)^m \sigma_m^2 \cos\left(m(\theta-\theta')\right). \end{aligned} \tag{25}$$

A flat prior $\sigma_m^2 = \sigma_u^2$ for all polar harmonics and a characteristic length scale ℓ as another hyperparameter yields

$$k(\mathbf{x},\mathbf{x}';\ell,\sigma_u) = \sigma_u^2 \frac{1-\frac{rr'}{\ell^2}\cos(\theta-\theta')}{1-2\frac{rr'}{\ell^2}\cos(\theta-\theta')+\frac{(rr')^2}{\ell^4}} = \sigma_u^2 \frac{1-\frac{\mathbf{x}\cdot\mathbf{x}'}{\ell^2}}{1-2\frac{\mathbf{x}\cdot\mathbf{x}'}{\ell^2}+\frac{|\mathbf{x}|^2|\mathbf{x}'|^2}{\ell^4}}. \tag{26}$$

This kernel is not stationary, but isotropic around a fixed coordinate origin. Introducing a mirror point $\bar{\mathbf{x}}'$ with polar angle $\bar{\theta}' = \theta'$ and radius $\bar{r}' = \ell^2/r'$ we notice that (26) can be written as

$$k(\mathbf{x},\mathbf{x}';\ell,\sigma_u) = \sigma_u^2 \frac{|\bar{\mathbf{x}}'|^2 - \mathbf{x}\cdot\bar{\mathbf{x}}'}{(\mathbf{x}-\bar{\mathbf{x}}')^2}, \tag{27}$$

making a dipole singularity apparent at $\mathbf{x} = \bar{\mathbf{x}}'$. In addition, k is normalized to 1 at $\mathbf{x} = 0$. Choosing $\ell > R_0$ larger than the radius R_0 of a circle centered in the origin and enclosing the computational domain, we have $\bar{r}' > \ell^2/\ell = \ell > R_0$. Thus, all mirror points and the according singularities are moved outside the domain. This behavior is illustrated in Figure 1 where computing the covariance kernel with respect to point $\mathbf{x}' = (0.8, 0)$ leads to distances > 1 everywhere inside the unit circle.

Choosing a slowly decaying $\sigma_m^2 = \sigma_u^2/m$, excluding $m = 1$ and adding a constant term yields a logarithmic kernel instead [9] with

$$k(\mathbf{x}, \mathbf{x}'; \ell, \sigma_u) = \sigma_u^2 \left(1 - \frac{1}{2} \ln \left(1 - 2\frac{\mathbf{x} \cdot \mathbf{x}'}{\ell^2} + \frac{|\mathbf{x}|^2 |\mathbf{x}'|^2}{\ell^4}\right)\right)$$
$$= \sigma_u^2 \left(1 - \ln \left(\frac{|\mathbf{x} - \bar{\mathbf{x}}'|}{|\bar{\mathbf{x}}'|}\right)\right). \tag{28}$$

Instead of a dipole singularity that expression features a monopole singularity at $\mathbf{x} - \bar{\mathbf{x}}'$ that is again avoided by placing it outside the domain for any pair of $\mathbf{x}$ and $\mathbf{x}'$ (Figure 1).

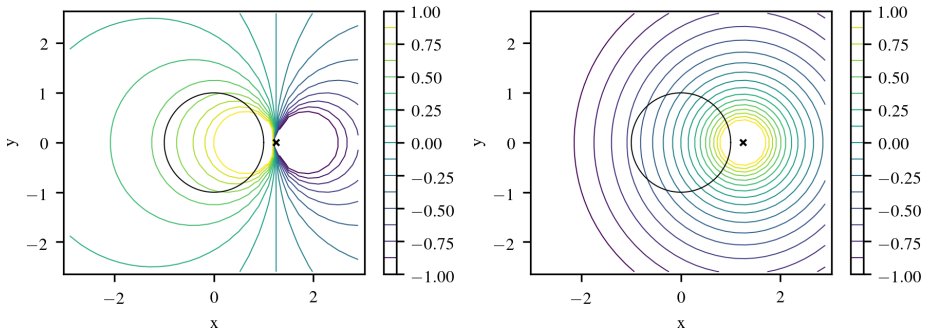


Figure 1. Kernels $k(\mathbf{x}, \mathbf{x}')$ evaluated at $\mathbf{x} = (x, y)$ and $\mathbf{x}' = (0.8, 0)$. **Left**: dipole response of (27), **right**: monopole response of (28). Singularities are moved outside the domain of interest.

Using instead Cartesian coordinates x, y to separate the Laplacian provides harmonic functions like

$$u(x, y) = e^{\pm\kappa x} e^{\pm i\kappa y}. \tag{29}$$

Here, all solutions yield finite values at $x = 0$, so we do not have to exclude any of them a priori. Introducing, again, a diagonal covariance operator in (20) and taking the real part yields

$$k(\mathbf{x}, \mathbf{x}'; \sigma^2(\kappa)) = \int \varphi(\mathbf{x}, \kappa) \sigma^2(\kappa) \varphi(\mathbf{x}', \kappa) \, \mathrm{d}\kappa = \mathrm{Re} \int_{-\infty}^{\infty} \sigma^2(\kappa) e^{\kappa(x \pm x')} e^{i\kappa(y \pm y')} \, \mathrm{d}\kappa. \tag{30}$$

Setting $\sigma^2(\kappa) \equiv e^{-2\kappa^2}$ and choosing a characteristic length scale ℓ together with a possible rotation angle θ_0 of the coordinate frame yields the kernel

$$k(\mathbf{x}, \mathbf{x}'; \ell, \theta_0, \sigma_u) = \frac{\sigma_u^2}{2} \mathrm{Re} \exp \left(\frac{((x + x') \pm i(y - y'))^2 e^{i2\theta_0})}{\ell^2}\right). \tag{31}$$

Other sign combinations do not yield a positive definite kernel – similar to the polar kernel (27) before we couldn't obtain an fully stationary expression that depends only on differences between coordinates of $\mathbf{x}$ and $\mathbf{x}'$.

For demonstration purposes we consider an analytical solution to a boundary value problem of Laplace's equation on a square domain Ω with corners at $(x, y) = (\pm 1, \pm 1)$. The reference solution is

$$u_{\mathrm{ref}}(x, y) = \frac{1}{2} \left(e^y \cos(x) + e^{2x} \cos(2y)\right) \tag{32}$$

and depicted in the upper left of Figure 2 together with the extension outside the boundaries. This figure also shows results from a GP fitted based on data with artificial noise of $\sigma_{\mathrm{n}} = 0.1$ measured at 8 points using kernel (27) with optimized maximum-likelihood (ML) values for hyperparameters

ℓ and σ_u but fixed σ_n. Inside Ω the solution is represented with errors below 5%. This is also reflected in the error predicted by the posterior variance of the GP that remains small in the region enclosed by measurement points. The analogy in classical analysis is the theorem that the solution of a homogeneous elliptic equation is fully determined by boundary values.

In comparison, a reconstruction using a generic squared exponential kernel

$$k(\mathbf{x}, \mathbf{x}'; \ell, \sigma_u) = \sigma_u^2 \exp\left(\frac{-(\mathbf{x} - \mathbf{x}')^2}{2\ell^2}\right) \tag{33}$$

yields a much worse approximation quality in Figures 2 and 3. This is in contrast to earlier investigations [1] where a fixed length scale hyperparamter $\ell = 2$ was used. Although the specialized GP with kernel (27) could identify this length scale during hyperparameter optimization, using a generic kernel (33) leads to an underestimation of ℓ and requires twice the number of training points to achieve a similar fit quality and profits from scattered training points, as it has no information about the nature of the boundary value problem (Figures 4 and 5).

In addition, the posterior covariance of that reconstruction is not able to capture the vanishing error inside the enclosed domain due to given boundary data. More severely, in contrast to the specialized GP, the posterior mean $\bar{u}$ does not satisfy Laplace's equation $\Delta\bar{u} = 0$ exactly. This leads to a violation of the classical result that (differences of) solutions of Laplace's equation may not have extrema inside Ω, showing up in the difference to the reconstruction in Figures 3 and 4. This kind of error is quantified by computation of the reconstructed charge density $\bar{q} = \Delta\bar{u}$. This is fine if data from Poisson's equation $\Delta u = q$ with distributed charges should be fitted instead. However, to keep $\Delta u = 0$ exact in Ω, one requires more specialized kernels such as (27).

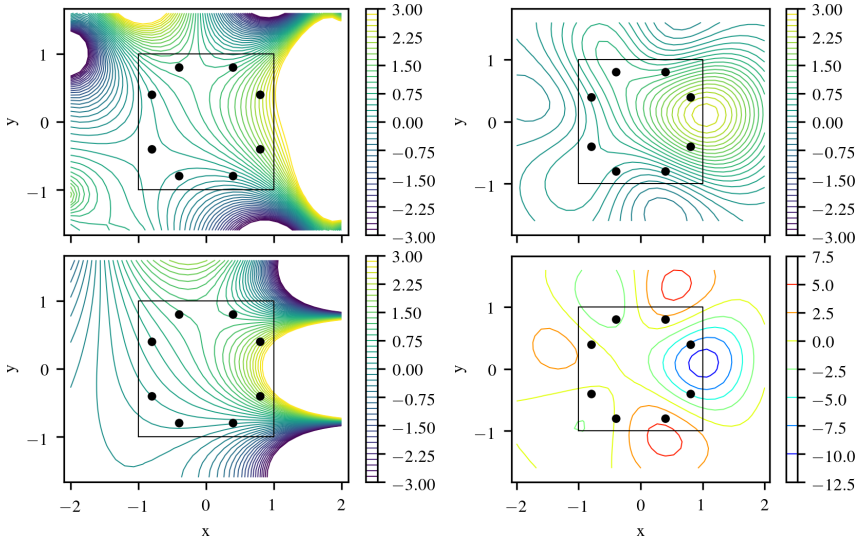


Figure 2. GP reconstruction of Laplace's equation with specialized locally source-free Mercer kernel (27) (**top left**) and generic squared exponential kernel (**top right**). Sources lie outside the black square region and 8 measurement positions are marked by black dots. Reference analytical solution (**bottom left**). Source density $\bar{q} = \Delta\bar{u}$ of prediction via a generic squared exponential kernel (**bottom right**).

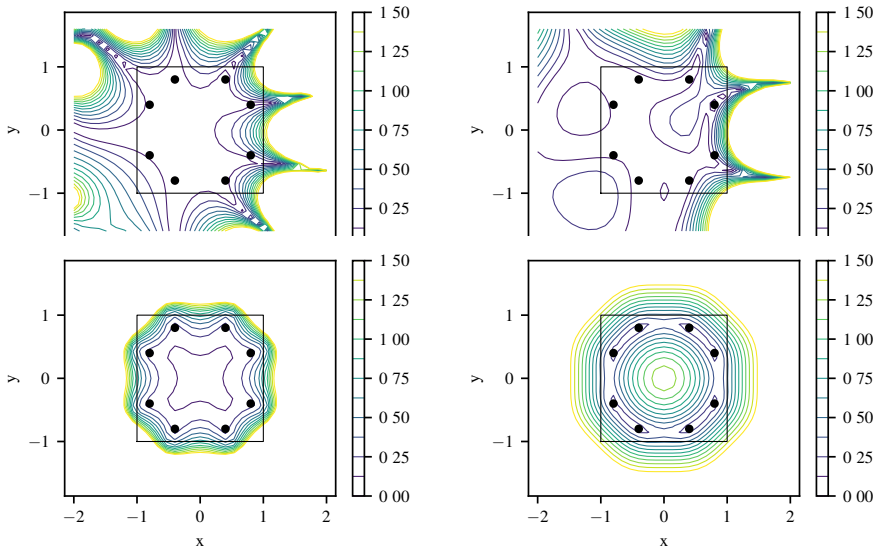


Figure 3. Absolute error (**top left**) and predicted 95% confidence interval (**bottom left**) with specialized locally source-free Mercer kernel (27) in comparison to absolute error (**top right**) and predicted 95% confidence interval (**bottom right**) with generic squared exponential kernel for 8 training points.

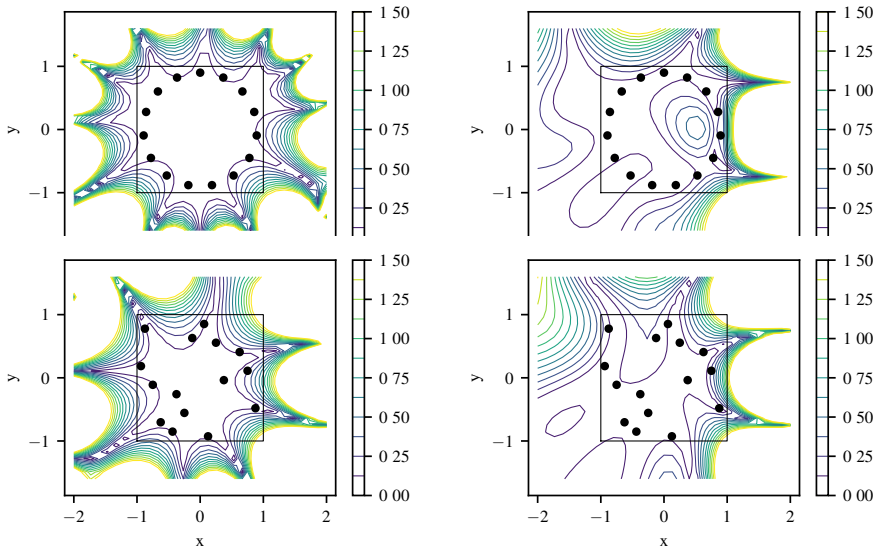


Figure 4. Absolute error as in Figure 3 for 15 training points on a circle (**top**) and for quasi-random points (**bottom**). As the generic squared exponential kernel does not fulfill the given differential equation, even for a larger number of training points, the source density doesn't vanish in the domain.

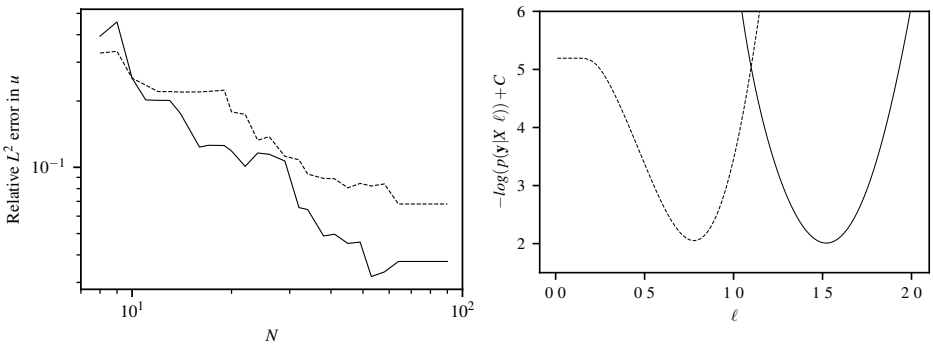


Figure 5. (**Left**) Comparison of relative L^2 error in u for specialized kernel (solid line) and squared exponential kernel (dashed line) for Laplace's equation for N quasi-random training points. (**Right**) Negative log likelihood from 8 training data of Figure 2 with optimum at $\ell = 1.52$ for specialized kernel (solid line) and at $\ell = 0.78$ for the squared exponential kernel (dashed line).

3.2. Heat Equation: Physical Parameter Estimation

Let us now consider the 1D homogeneous heat/diffusion equation over position x and time t,

$$\frac{\partial u(x,t)}{\partial t} - D\Delta u(x,t) = 0 \tag{34}$$

for $(x,t) \in \mathbb{R} \times \mathbb{R}^+$. Here, the diffusivity D is a physical parameter determining how fast solutions spread in space. Integrating the fundamental solution

$$G(x,t,\xi,\tau) = \frac{1}{\sqrt{4\pi D(t-\tau)}} \exp\left(-\frac{(x-\xi)^2}{4D(t-\tau)}\right) \tag{35}$$

from $\xi = -\infty$ to ∞ at $\tau = 0$, i.e., placing sources everywhere in space at a single initial time, and adding a scale hyperparameter σ_u leads to the convolution kernel

$$k_{\mathrm{n}}(x,t,x',t';D,\sigma_u) = \frac{\sigma_u^2}{\sqrt{4\pi D(t+t')}} \exp\left(-\frac{(x-x')^2}{4D(t+t')}\right). \tag{36}$$

In terms of x, this is a stationary squared exponential kernel and the natural kernel over the domain $x \in \mathbb{R}$. The kernel broadens with increasing t and t'. Nonstationarity in time can also be considered natural to the heat equation, as its solutions show a preferred time direction on each side of the singularity $t = 0$. The only difference of (36) to the fundamental solution (35) is the positive sign between t and t'. As both t and t' are positive, k is guaranteed to take finite values and, in contrast to (35), does not become singular at $(x,t) = (x',t')$.

As for the Laplace equation it is also convenient to define a non-stationary kernel by cutting out a domain that is known to be free of sources. In case heat sources are known to exist only left of the origin we evaluate the integral over the fundamental solution over $(-\infty, 0)$ to

$$k(x,t,x',t';D,\sigma_u) = k_{\mathrm{n}}(x,t,x',t';D,\sigma_u)\left[1 + \frac{g(x,t,x',t';D)}{2}\right], \tag{37}$$

where

$$g(x,t,x',t';D) \equiv \mathrm{erf}\left(-\frac{x/t + x'/t'}{2\sqrt{D}\sqrt{1/t + 1/t'}}\right) \tag{38}$$

is defined via the error function erf. Choosing instead a source-free region domain interval (a, b) we integrate over $\mathbb{R}\backslash(a, b)$ and obtain

$$k(x,t,x',t';D,\sigma_u) = k_{\mathrm{n}}(x,t,x',t';D,\sigma_u)\left[1 - \frac{g(x-b,t,x'-b,t';D) - g(x-a,t,x'-a,t';D)}{2}\right]. \quad (39)$$

Incorporating the prior knowledge that there are no domain sources is expected to improve the reconstruction.

As a physical example, we consider a rod with temperatures held fixed at two ends and a given initial temperature distribution. We model this as an initial-boundary value problem for (34) on the interval $x \in (0,1)$ with Dirichlet boundary data $u(0) = 1$ and $u(1) = 0$. As initial conditions, we set $u(x,0) = 0$ everywhere except at the left end where $u(0,0) = 1$. The actual diffusivity is chosen as $D = 0.1$, and we let $u(x,t)$ evolve from $t_0 = 0$ until $t_1 = 1$. With increasing t the initial conditions are smoothed out as u approaches the stationary solution $u(x, t \to \infty) = 1 - x$. Measurements of u are performed at three positions $x = 0, 0.1, 1$ at four times $t = 10^{-5}, 0.25, 0.5, 0.75$, yielding 12 training points in total. In Figure 6 the resulting reconstruction of $u(x, t = 0.125)$ is plotted for each of the three kernels defined above. Kernel (39) allowing initial sources on both sides of the interval yields the best reconstruction. Furthermore, it is the only one that reproduces meaningful uncertainty bands based on the 95% confidence interval $\bar{u} \pm 1.96\sigma$, whereas the ones for (36) and (36) span the whole plot domain. Estimation of diffusivity D is also most reliable with kernel (39). The according negative log likelihood can be seen on the right plot in Figure 6. Although all three kernels produce well posed optimization problems, only (39) has the minimum at the correct position $D = 0.1$.

The reason for the requirement of kernel (39) is clear from the statement of the problem: keeping u fixed on both sides of the interval can only be achieved by restricting the heat flux in a predefined way that requires sources on both sides at $t = 0$. However, the domain itself should not contain any heat sources at any time. If we had placed an open boundary condition on the right side, kernel (37) would have been the more natural choice instead.

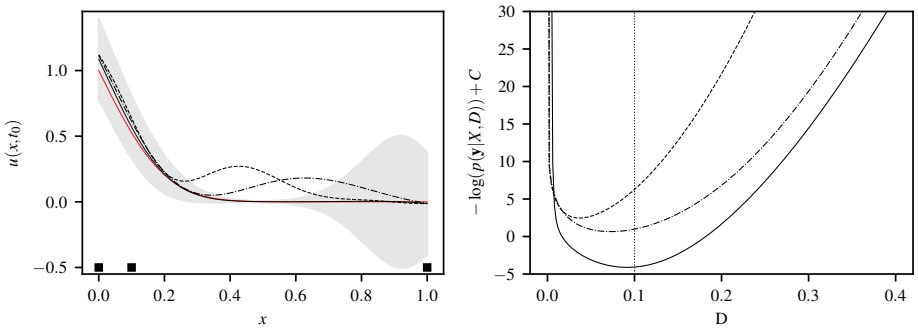


Figure 6. (**Left**) GP reconstruction of $u(x, t = 0.125)$ for 1D heat equation Dirichlet problem based on measurement points (■) at $x = 0, 0.1, 1$, reference in red. Kernels (36), (37) and (39) marked by dashed, dash-dotted and solid lines, respectively. 95% confidence interval bands shown only for (39), producing the best fit. (**Right**) negative log likelihood over diffusivity D.

3.3. Helmholtz Equation: Source and Wavenumber Reconstruction

Finally, to demonstrate the full method, we consider the Helmholtz equation with sources:

$$\Delta u(\mathbf{x}) + k_0^2 u(\mathbf{x}) = q(\mathbf{x}). \tag{40}$$

In 1D, solutions for the homogeneous equation with $\mathbf{x} = x$ are given by linear combinations of $\cos(k_0 x), \sin(k_0 x)$. Choosing a diagonal prior in (18) leads to a stationary kernel

$$k(x, x'; k_0, \sigma_u) = \cos(k_0 x)\sigma_u \cos(k_0 x') + \sin(k_0 x)\sigma_u \sin(k_0 x') = \sigma_u \cos(k_0(x - x')), \tag{41}$$

as presented in [11]. For the two-dimensional case in polar coordinates, a family of solutions based on the separation of variables is

$$\cos(m\theta), \sin(m\theta), J_m(k_0 r), Y_m(k_0 r), \tag{42}$$

where J_m and Y_m are Bessel functions of first and second kind, respectively. Similar to the simpler 1D case, by applying Neumann's addition theorem, we obtain a specialized kernel

$$k(\mathbf{x}, \mathbf{x}'; k_0, \sigma_u) = \sigma_u^2 J_0(k_0|\mathbf{x} - \mathbf{x}'|). \tag{43}$$

In the 3D case, one would proceed in a similar fashion with spherical Bessel functions, which yields the kernel that was already postulated in [11]. In contrast to the case of Laplace's equation in the previous section, these source-free Helmholtz kernels do not possess singularities at any finite distance from the origin, i.e., no virtual exterior sources in the Mercer kernel (20). As a consequence they provide smoothing regularization on the order of the wavelength $\lambda_0 = 2\pi/k_0$ to reconstructed fields and boundary conditions that may or may not be desired. Internal sources at positions $\mathbf{x}_k^q$ are linearly modeled according to (10) with basis

$$h_i(\mathbf{x}) = G(\mathbf{x}, \mathbf{x}_i^q) = H_0^{(2)}(k_0|\mathbf{x} - \mathbf{x}_i^q|), \tag{44}$$

where $H_0^{(2)}$ is the Hankel function of the second kind. The method of source strength reconstruction is improved compared to [11], as it constitutes a linear problem according to (15). Nonlinear optimization is instead applied to σ_u and wavenumber k_0 as free hyperparameters to be estimated during the GP regression. The set-up is the same as in [11]: a 2D cavity with various boundary conditions and two sound sources of strengths 0.5 and 1.0, respectively. Results for sound pressure fulfilling (40) are normalized to have a maximum $p/p_0 = 1$. We compare three variants of GP regression for these data:

(1) Superposition of specialized kernel GP for homogeneous part u_h and linear source model for u_p.
(2) Superposition of generic squared-exponential kernel GP for u_h and linear source model for u_p.
(3) Generic squared-exponential kernel GP model for the full field u.

Naturally, after the presented analysis, only (1) can be the "correct" way of regression for this kind of data from a PDE with point sources. Variant (2) is a "hybrid" that should be able to identify point sources, while polluting the source-free part with volumetric contributions. Considering that (2) helps to separate the effect from this pollution from the effect of adding a linear source model. Variant (3) is expected to show worse performance compared to (1) and (2), as neither source-free part nor singularities of u at point source positions can be modeled correctly.

Figure 7 shows the local absolute field reconstruction error based on 12 training data points with artificial noise of $\sigma_\mathrm{n} = 0.01$. Hyperparameters k_0 and σ_u are set to optimized ML values, and σ_n is fixed to its actual value. The upper left plot shows results for variant (1) with the specialized kernel (43). Variant (3) with a generic squared exponential kernel (33) of length scale $\ell = \pi/(\sqrt{2}k_0)$ to model u yields a much higher field reconstruction error as depicted in the lower left of Figure 7. The field reconstruction using the generic kernel is improved when a linear model for the inhomogeneous term

is included (variant (2)), as shown in the upper right of Figure 7. However, the original differential Equation (40) is only fulfilled exactly when using a specialized kernel with $\hat{L}k(\mathbf{x}, \mathbf{x}')\hat{L}' = 0$. As expected, variant (1) produces the best reconstruction at a given number of training points (Figure 8). There the first 12 points are chosen as marked in Figure 7, and more points are generated from a quasi-random Halton sequence. The obtained negative log-likelihood (Figure 7, lower right) depending on k_0 and σ_u at its ML value demonstrates the well-posedness of estimating k_0 having the physical meaning of a wavenumber. Variants (2) and (3) lead to a slightly less peaked estimate for a spatial length scale hyperparameter without a direct physical interpretation.

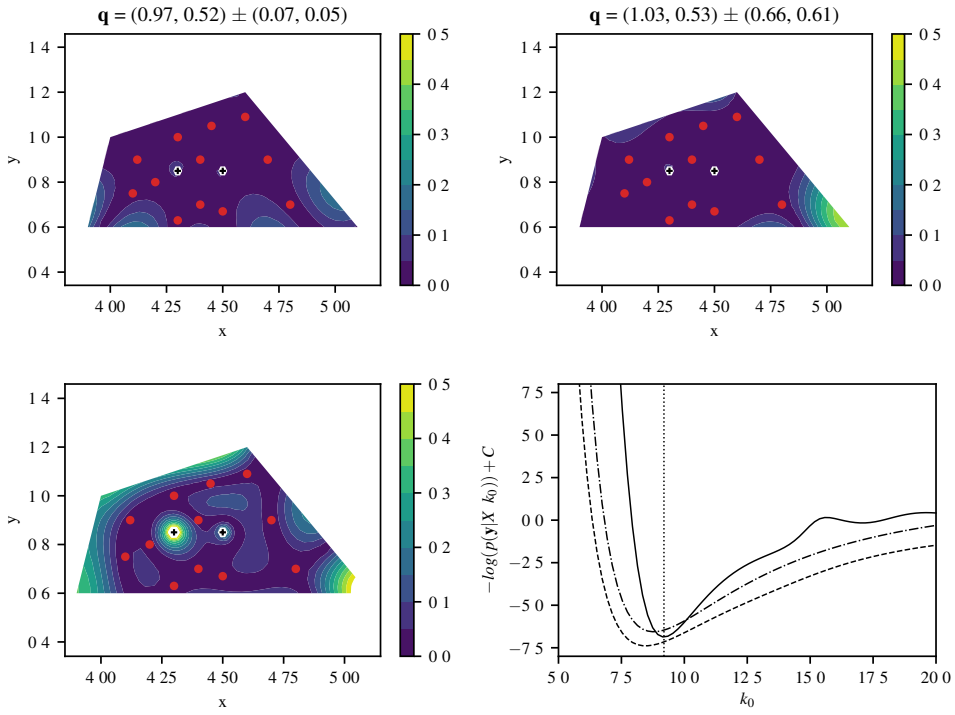


Figure 7. Reconstruction error for the Helmholtz equation from 12 training points for specialized kernel (**top left**), squared exponential kernel with linear source model (**top right**) and squared exponential kernel (**bottom right**); reconstructed source strengths $\mathbf{q}$ with 95% confidence interval via posterior (15) and (16). Negative log likelihood (**bottom right**) with optimum $k_0^{\mathrm{ML}} = 9.19$ for specialized kernel (solid line), sq.exp. kernel with linear source model (dashed), and sq.exp. kernel alone (dash-dotted).

For estimation of source positions, nonlinear optimization is applied to source positions as free hyperparameters within the given boundaries, employing an evolutionary algorithm CMA-ES [17]. The results of source strength and position estimation using (15) and (16) in the configuration with 12 training points is given in Table 1. Both estimates match the exact values reasonably well. At an increasing number of training data the reconstruction becomes more accurate, stagnating at an error between 0.1% and 1% and showing the advantage of the specialized kernel more clearly (Figures 8 and 9). The relative L^2 error in source positions for specialized and generic squared exponential kernel with linear source model is depicted in the left plot of Figure 9. Again, results from the specialized kernel are usually more accurate and stable compared to using a squared-exponential kernel for the source-free part of the field at a given number of training points.

Table 1. Comparison and results for estimation of source strength $\mathbf{q}$ and source position $\mathbf{x}_i^q$ for 12 training data points for specialized and squared exponential kernel with linear source model.

Exact Values	Specialized Kernel	sq. exp. Kernel
$\mathbf{q} = (1.0, 0.5)$	$\mathbf{q} = (0.97, 0.52)$	$\mathbf{q} = (1.03, 0.53)$
$\mathbf{x}_1^q = (4.3, 0.85)$	$\mathbf{x}_1^q = (4.31, 0.85)$	$\mathbf{x}_1^q = (4.30, 0.82)$
$\mathbf{x}_2^q = (4.5, 0.85)$	$\mathbf{x}_2^q = (4.65, 0.90)$	$\mathbf{x}_2^q = (4.61, 0.84)$

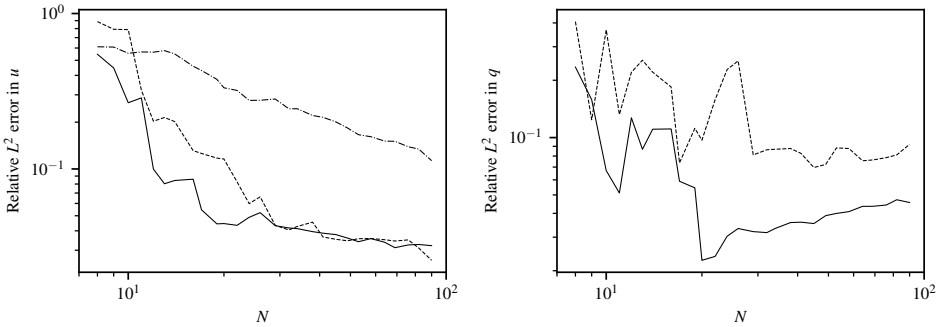


Figure 8. Comparison of relative L^2 error in u (**left**) and q (**right**) for specialized kernel (solid line), squared exponential kernel (dash-dotted) and squared exponential kernel with linear source model (dashed) for Helmholtz equation with N quasi-random training points. As the squared exponential kernel alone (without linear source model) cannot reproduce point sources, no result is shown for the point source strength estimation in the right plot for this case.

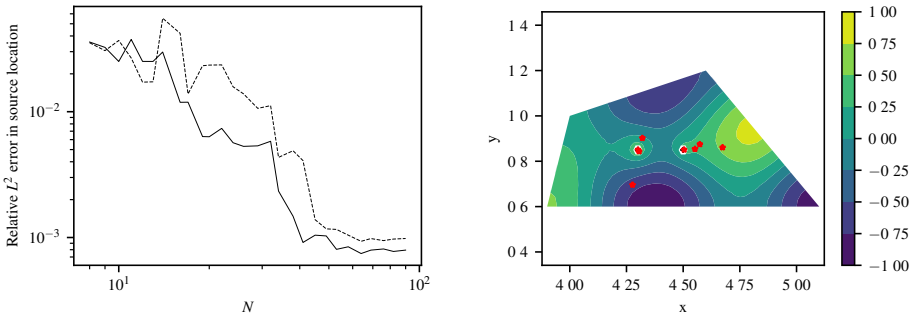


Figure 9. (**Left**) Comparison of relative L^2 error in source position for specialized kernel (solid line) and squared exponential kernel with linear source model (dashed) for Helmholtz equation with N quasi-random training points. (**Right**) reconstructed field using specialized kernel (43) and showing convergence of estimated source location for $N = (12, 15, 20, 30)$ quasi-random training points.

4. Summary and Outlook

A framework for application of Gaussian process regression to data from underlying linear partial differential equations with localized sources has been presented. The method is based on superposition of a Gaussian process that generates exact solutions of the homogeneous equation, complemented by a linear model for sources. For the homogeneous part, specialized kernels are constructed from fundamental solutions via Mercer's theorem. For source contributions, fundamental solutions are used as basis functions in the linear model. Examples for suitable kernels have been given for Laplace's equation, heat equation and Helmholtz equation. Regression has been shown to yield better results compared to using a squared exponential kernel at the same number of training

points in the considered application cases. Advantages of the specialized kernel approach are the possibility to represent exact absence of sources as well as physical interpretability of hyperparameters. This comes at the cost of requiring non-standard, possibly nonstationary kernels. The presented method has been demonstrated to be able to accurately estimate system parameters such as diffusivity and wavenumber, as well as position and strength of point sources using only around 10 training data points in two-dimensional domains.

In a next step, reconstruction of vector fields via GPs could be formulated, taking laws such as Maxwell's equations or Hamilton's equations of motion into account. A starting point could be squared exponential kernels for divergence- and curl-free vector fields [18]. Such kernels have been used in [19] to perform statistical reconstruction, and [20] apply them to GPs for source identification in the Laplace/Poisson equation. To model Hamiltonian dynamics in phase-space, vector-valued GPs could possibly be extended to represent not only volume-preserving (divergence-free) maps but retain full symplectic properties, conserving all integrals of motion such as energy or momentum.

Author Contributions: Conceptualization, C.G.A. and K.R.; investigation, C.G.A. and K.R.; methodology, C.G.A. and K.R.; software, C.G.A. and K.R.; supervision, C.G.A.; visualization, C.G.A. and K.R.; writing—original draft, C.G.A. and K.R.; writing—review and editing, C.G.A. and K.R. All authors have read and agreed to the published version of the manuscript.

Funding: This study is a contribution to the Reduced Complexity Models grant number ZT-I-0010 funded by the Helmholtz Association of German Research Centers and received funding from the Munich School of Data Science (MUDS).

Acknowledgments: The authors would like to thank Dirk Nille, Roland Preuss, and Udo von Toussaint for insightful discussions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Albert, C.G. Gaussian Processes for Data Fulfilling Linear Differential Equations. In Proceedings of the 39th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Munich, Germany, 30 June–5 July 2019. [CrossRef]
2. Dong, A. Kriging Variables that Satisfy the Partial Differential Equation ΔZ = Y. *Geostatistics* **1989**, *4*, 237–248. [CrossRef]
3. van den Boogaart, K.G. Kriging for Processes Solving Partial Differential Equations. In Proceedings of the IAMG2001, Cancun, Mexiko, 10–12 September 2001; pp. 1–21.
4. Graepel, T. Solving Noisy Linear Operator Equations by Gaussian Processes: Application to Ordinary and Partial Differential Equations. In Proceedings of the Twentieth International Conference on Machine Learning, Washington, DC, USA, 21–24 August 2003; pp. 234–241.
5. Särkkä, S. Linear Operators and Stochastic Partial Differential Equations in Gaussian Process Regression. In Proceedings of the 21st International Conference on Artificial Neural Networks, Espoo, Finland, 14–17 June 2011; pp. 151–158. [CrossRef]
6. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Inferring Solutions of Differential Equations Using Noisy Multi-Fidelity Data. *J. Comput. Phys.* **2017**, *335*, 736–746. [CrossRef]
7. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Machine Learning of Linear Differential Equations Using Gaussian Processes. *J. Comput. Phys.* **2017**, *348*, 683–693. [CrossRef]
8. Yang, X.; Tartakovsky, G.; Tartakovsky, A. Physics-Informed Kriging: A Physics-Informed Gaussian Process Regression Method for Data-Model Convergence. *arXiv* **2018**, arXiv:1809.03461.
9. Mendes, F.M.; da Costa Junior, E.A. Bayesian Inference in the Numerical Solution of Laplace's Equation. *AIP Conf. Proc.* **2012**, *1443*, 72–79. [CrossRef]
10. Cockayne, J.; Oates, C.; Sullivan, T.; Girolami, M. Probabilistic Numerical Methods for Partial Differential Equations and Bayesian Inverse Problems. *arXiv* **2016**, arXiv:1605.07811.
11. Albert, C. Physics-Informed Transfer Path Analysis With Parameter Estimation Using Gaussian Processes. In Proceedings of the 23rd International Congress on Acoustics, Aachen, Germany, 9–13 September 2019; pp. 459–466. [CrossRef]

12. Lackner, K. Computation of Ideal MHD Equilibria. *Comput. Phys. Commun.* **1976**, *12*, 33–44. [CrossRef]
13. Golberg, M.A. The Method of Fundamental Solutions for Poisson's Equation. *Eng. Anal. Bound. Elem.* **1995**, *16*, 205–213. [CrossRef]
14. Schaback, R.; Wendland, H. Kernel Techniques: From Machine Learning to Meshless Methods. *Acta Numer.* **2006**, *15*, 543–639. [CrossRef]
15. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
16. O'Hagan, A. Curve Fitting and Optimal Design for Prediction. *J. R. Stat. Soc. Ser. B* **1978**, *40*, 1–24. [CrossRef]
17. Hansen, N.; Akimoto, Y.; Baudis, P. CMA-ES/Pycma on Github. Available online: https://doi.org/10.5281/zenodo.2559634 (accessed on 13 December 2019).
18. Narcowich, F.J.; Ward, J.D. Generalized Hermite Interpolation via Matrix-Valued Conditionally Positive Definite Functions. *Math. Comput.* **1994**, *63*, 661. [CrossRef]
19. Macêdo, I.; Castro, R. Learning Divergence-Free and Curl-Free Vector Fields with Matrix-Valued Kernels. Available online: http://preprint.impa.br/FullText/Macedo__Thu_Oct_21_16_38_10_BRDT_2010/macedo-MVRBFs.pdf (accessed on 13 December 2019).
20. Cobb, A.D.; Everett, R.; Markham, A.; Roberts, S.J. Identifying Sources and Sinks in the Presence of Multiple Agents with Gaussian Process Vector Calculus. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18), London, UK, 19–23 August 2018; pp. 1254–1262.

3.2 Symplectic Gaussian Process Regression of maps in Hamiltonian systems

Main novelty:
The paper introduces a structure-preserving symplectic surrogate model for Hamiltonian flow maps and Poincaré maps based on Gaussian process regression deploying a specialized, matrix-valued kernel.

Contributing article:
Rath, K., Albert, C. G., Bischl, B., and von Toussaint, U. (2021b). Symplectic Gaussian process regression of maps in Hamiltonian systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(5):053121

Author contributions:
Christopher Albert devised the conceptual idea of the project. Katharina Rath implemented the code, performed numerical experiments and designed the validation and evaluation routines. Christopher Albert transferred part of the software implementation to Fortran. Katharina Rath wrote the main parts of the paper with some contributions by Christopher Albert. Bernd Bischl and Udo von Toussaint gave valuable input throughout the project and suggested several notable modifications.

Chaos
An Interdisciplinary Journal of Nonlinear Science

RESEARCH ARTICLE | MAY 18 2021

Symplectic Gaussian process regression of maps in Hamiltonian systems

Katharina Rath ; Christopher G. Albert ; Bernd Bischl ; Udo von Toussaint

Chaos 31, 053121 (2021)
https://doi.org/10.1063/5.0048129

Chaos ARTICLE scitation.org/journal/cha

Symplectic Gaussian process regression of maps in Hamiltonian systems

Cite as: Chaos **31**, 053121 (2021); doi: 10.1063/5.0048129
Submitted: 19 February 2021 · Accepted: 26 April 2021 ·
Published Online: 18 May 2021

View Online Export Citation CrossMark

Katharina Rath,[1,2,a] Christopher G. Albert,[2] Bernd Bischl,[1] and Udo von Toussaint[2]

AFFILIATIONS

[1] Department of Statistics, Ludwig-Maximilians-Universität München, Ludwigstr. 33, 80539 Munich, Germany
[2] Max Planck Institute for Plasma Physics, Boltzmannstr. 2, 85748 Garching, Germany

[a] **Author to whom correspondence should be addressed:** katharina.rath@ipp.mpg.de

ABSTRACT

We present an approach to construct structure-preserving emulators for Hamiltonian flow maps and Poincaré maps based directly on orbit data. Intended applications are in moderate-dimensional systems, in particular, long-term tracing of fast charged particles in accelerators and magnetic plasma confinement configurations. The method is based on multi-output Gaussian process (GP) regression on scattered training data. To obtain long-term stability, the symplectic property is enforced via the choice of the matrix-valued covariance function. Based on earlier work on spline interpolation, we observe derivatives of the generating function of a canonical transformation. A product kernel produces an accurate implicit method, whereas a sum kernel results in a fast explicit method from this approach. Both are related to symplectic Euler methods in terms of numerical integration but fulfill a complementary purpose. The developed methods are first tested on the pendulum and the Hénon–Heiles system and results compared to spectral regression of the flow map with orthogonal polynomials. Chaotic behavior is studied on the standard map. Finally, the application to magnetic field line tracing in a perturbed tokamak configuration is demonstrated. As an additional feature, in the limit of small mapping times, the Hamiltonian function can be identified with a part of the generating function and thereby learned from observed time-series data of the system's evolution. For implicit GP methods, we demonstrate regression performance comparable to spectral bases and artificial neural networks for symplectic flow maps, applicability to Poincaré maps, and correct representation of chaotic diffusion as well as a substantial increase in performance for learning the Hamiltonian function compared to existing approaches.

Discrete representations of Hamiltonian systems require structure-preserving properties in order to preserve invariants of motion and the orbit topology in the phase space. Here, we investigate techniques based on Gaussian process regression to learn such a representation of flow maps and Poincaré maps from orbit data without explicit knowledge of the Hamiltonian function. The approach supports unstructured data on irregular domains as well as non-canonical coordinates if an implicit transformation to canonical ones is available. Similarly to existing work on interpolated flow maps, the method relies on canonical transformations and their generating functions and is related to first order symplectic integrators. After the construction of the map, it can be used to compute evolving system states over long periods of time. A single mapping time step can cover the time that the system spends between two Poincaré sections. Besides the use to characterize Hamiltonian systems from observational data, the method can thus be used to construct fast emulators for numerical orbit tracers.

I. INTRODUCTION

Hamlitonian mechanics form the basis for a large number of models for dynamical systems in physics and engineering. This includes systems with negligible dissipation found in classical mechanics, electrodynamics, continuum mechanics, and plasma physics[1–3] as well as artificial systems created for numerical purposes such as hybrid-Monte-Carlo algorithms[4] for sampling from probability distributions. A specific feature of Hamiltonian systems is their long-term behavior with conservation of invariants of motion and lack of attractors to which different initial conditions converge. Alternatively, a diverse spectrum of resonant and stochastic

Chaos ARTICLE scitation.org/journal/cha

features emerges that has been extensively studied in the field of chaos theory.[5] These particular properties are a consequence of the symplectic structure of the phase space together with equations of motion based on derivatives of a scalar field—the Hamiltonian H.

Numerical methods that partially or fully preserve this structure in a discretized system are known as geometric or symplectic integrators.[6–10] Most importantly, such integrators do not accumulate energy or momentum and remain long-term stable at relatively large time steps compared to non-geometric methods.[11] Symplectic integrators are generally (semi-)implicit and formulated as (partitioned) Runge–Kutta schemes that evaluate derivatives of H at different points in time.

Here, we investigate mapping techniques[12–16] that serve a purpose complementary to numerical integration. Numerical integrators yield approximate orbits from the knowledge of (derivatives of) H. Mapping techniques instead rely on given orbit data over a period of time and find an approximation of the flow over this mapping time step in a functional basis. Training data can come from numerical integration or from experiment. Conversely to numerical integration, this allows one to learn the dynamics, i.e., the Hamiltonian of a system under investigation.[17] In contrast to time steps of numerical integrators, the mapping time step is not necessarily required to be small compared to periods of motion of a system. On the contrary, a map can be constructed between Poincaré sections of interest.[18] Once the map is learned, it can be applied to traverse time in such "giant" steps as long as orbits remain in the training region. Hence, the constructed map allows one to trace a system over many periods based on data from numerical integration or experiment of only a single period. This is especially useful to accelerate long-term computations and to study chaos in systems with a broken symmetry.

Naively interpolating a map in both, position and momentum variables, destroys the symplectic property of the Hamiltonian flow. In turn, all favorable properties of symplectic integrators are lost, and subsequent applications of the map become unstable very quickly. This problem is illustrated in Fig. 1, where the flow map of a pendulum is interpolated in a symplectic and a non-symplectic manner, respectively. If one enforces symplecticity of the interpolated map by some means, structure-preservation and long-term stability are again natural features of the approximate map. Here, this will be realized via generating functions introduced by Warnock *et al.*[13,16] in this context. This existing work relies on a tensor-product basis of Fourier series and/or piecewise spline polynomials. This choice of basis has two major drawbacks: rapid decrease of efficiency in higher dimensions and limitation to box-shaped domains. One possibility to overcome these limitations would be the application of artificial neural networks with symplectic properties.[18–26] Here, we rather use a kernel-based method as a new way to construct approximate symplectic maps via Gaussian process (GP) regression. In contrast to other existing works[27,28] on learning dynamical systems using GPs, the present method is specialized to symplectic Hamiltonian flow and Poincaré maps. An important limitation of the use of mixed-variable generating functions is the possibility that these functions or their derivatives may become non-unique-valued. This issue will be pointed out in the text and can in certain cases be overcome by composition of multiple maps with shorter step lengths ("deep" GP).

GP regression,[29] also known as Kriging, is a flexible method to represent smooth functions based on covariance functions (kernels) with tunable parameters. These kernel hyperparameters can be directly optimized in the training process by maximizing the marginal likelihood. Predictions, in particular, posterior mean and (co-)variance for function values are then made via the inverse kernel covariance matrix. Observation of derivatives required to fit Hamiltonian flow maps is possible via correlated multi-output Gaussian processes.[30–33]

We apply the developed method to toy models as well as an application case on magnetic field line tracing that represents a simplified variant of tracing plasma particles. Details on this problem are found in Appendix B. Tracing field lines of magnetic confinement devices over many periods is an important task on its own, in particular, near the device wall and in the presence of non-axisymmetric perturbations.[12,34] Charged particle orbits in strongly

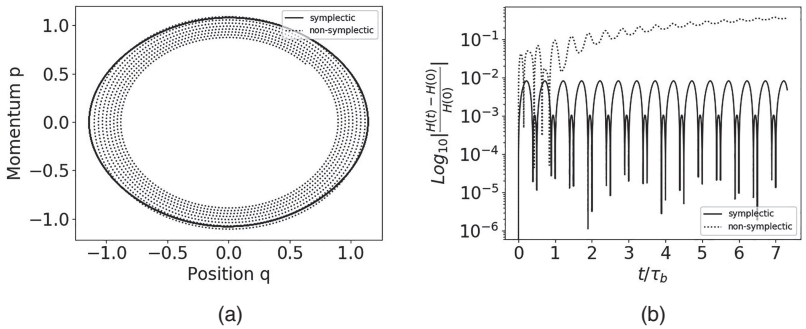


FIG. 1. Illustration of a pendulum orbit in the phase space (a) and relative energy error (b) using symplectic Euler (solid line) and non-symplectic explicit Euler (dotted line) schemes with step size $h = 0.01$ for initial conditions $(q, p) = (1, 0.5)$. The horizontal axis is given by t/τ_b, where τ_b is the bounce time.

Chaos ARTICLE scitation.org/journal/cha

magnetized plasmas are commonly represented by their center of gyration around magnetic field lines: the guiding-center.[34–37] In the limiting case of zero energy and magnetic moment, guiding-centers coincide with magnetic field lines and can be treated in a similar formalism.[38] This feature is used here to demonstrate the mapping technique on magnetic field perturbations while already keeping the more general case of guiding-center motion in mind for future application. The main difference here lies in the variety of orbit classes rather than fundamental features of the system. In all cases, we treat systems without explicit time dependencies. Within the guiding-center formalism, the rapidly changing gyrophase becomes an ignorable variable, thereby reducing the effective phase space dimension from 6 to 4. This reduction comes at the cost of a switch to non-canonical variables. Nevertheless, it is possible[34,39–41] to identify canonical variables $\boldsymbol{q}, \boldsymbol{p}$ as functions of non-canonical variables $\boldsymbol{z}$. The inverse coordinate transformation is then given implicitly. The availability of such a transformation makes canonical symplectic methods, in particular, the ones developed in this work, applicable despite a non-canonical formulation.

The paper is structured as follows: First, Hamiltonian systems and canonical transformations that preserve the symplectic structure of the phase space are briefly reviewed (Sec. II). In Sec. III, general derivations of multi-output Gaussian processes with derivative observations are given. The analogous derivation for linear regression in a spectral basis is found in Appendix A. Then, two algorithms to construct and apply symplectic mappings using Gaussian processes are introduced. The presented methods are tested on a simple pendulum and compared to linear regression using an expansion in Hermite polynomials combined with a periodic Fourier basis (Sec. IV). Poincaré maps are studied starting with the perturbed pendulum and the more complex Hénon–Heiles system. Then, the correct reproduction of chaotic diffusion is tested based on the standard map. Finally, the method is applied to the magnetic field in a tokamak with non-axisymmetric perturbations, again showing a transition from regular to chaotic behavior with increasing perturbation strength.

II. DYNAMICAL HAMILTONIAN SYSTEMS AND SYMPLECTIC FLOW MAPS

A. Hamiltonian systems

It is well known[1,69] that an f-dimensional classical mechanical system is fully characterized by its Hamiltonian function $H(\boldsymbol{q}, \boldsymbol{p}, t)$, which depends on f generalized coordinates $\boldsymbol{q}$, f generalized momenta $\boldsymbol{p}$, and time t. Here, we restrict ourselves to autonomous systems with $H(\boldsymbol{q}, \boldsymbol{p})$ having no explicit time-dependence and conceptually treat non-autonomous systems in an extended phase space with t as an additional position coordinate. Derivatives of H define a Hamiltonian vector field being

$$\boldsymbol{X}_H(\boldsymbol{q}, \boldsymbol{p}) = \begin{pmatrix} \nabla_p H(\boldsymbol{q}, \boldsymbol{p}) \\ -\nabla_q H(\boldsymbol{q}, \boldsymbol{p}) \end{pmatrix} \tag{1}$$

in a canonical representation. The time evolution of orbit's canonical coordinates $(\boldsymbol{q}(t), \boldsymbol{p}(t))$ is given as integral curves of $\mathbf{X}_H(\boldsymbol{q}, \boldsymbol{p})$, i.e., a solution to Hamilton's canonical equations of motion,

$$\dot{\boldsymbol{q}}(t) = \frac{d\boldsymbol{q}(t)}{dt} = \nabla_p H(\boldsymbol{q}(t), \boldsymbol{p}(t)), \tag{2}$$

$$\dot{\boldsymbol{p}}(t) = \frac{d\boldsymbol{p}(t)}{dt} = -\nabla_q H(\boldsymbol{q}(t), \boldsymbol{p}(t)). \tag{3}$$

The evolution of a system along $\mathbf{X}_H(\boldsymbol{q}, \boldsymbol{p})$ over finite time intervals is described by the Hamiltonian flow map φ_H. This map preserves the symplectic structure of the phase space,[2] resulting in important properties such as conservation of phase volume (Liouville's theorem) and invariants such as energy along orbits.

B. Canonical transformations

One is usually interested in the temporal evolution according to Eq. (3), that is, position $\boldsymbol{Q}$ and momentum $\boldsymbol{P}$ of a system at time $t = h$ that has been initialized with position $\boldsymbol{q}$ and momentum $\boldsymbol{p}$ at time $t = 0$. Motion (or a shift in time) in a Hamiltonian system is conveniently represented by a canonical transformation.[1,70] Due to the canonical structure of equations of motion, the mapping relations linking $\boldsymbol{q}$, $\boldsymbol{p}$, $\boldsymbol{Q}$, and $\boldsymbol{P}$ are not independent from each other but linked via the symplectic property

$$\frac{\partial \boldsymbol{Q}(\boldsymbol{q}, \boldsymbol{P})}{\partial \boldsymbol{q}} - \frac{\partial \boldsymbol{p}(\boldsymbol{q}, \boldsymbol{P})}{\partial \boldsymbol{P}} = 0. \tag{4}$$

Equation (4) is analogous to divergence- or curl-freeness of vector fields, which is seen in a formulation using differential forms.[2] Similar to using a scalar or vector potential to guarantee such properties, symplecticity [Eq. (4)] can be automatically fulfilled by introducing a mixed-variable generating function $F(\boldsymbol{q}, \boldsymbol{P})$ that links old coordinates $(\boldsymbol{q}, \boldsymbol{p})$ to new coordinates $(\boldsymbol{Q}, \boldsymbol{P})$. For a type 2 generating function, the associated canonical transformation is given by

$$\boldsymbol{Q}(\boldsymbol{q}, \boldsymbol{P}) = \frac{\partial F(\boldsymbol{q}, \boldsymbol{P})}{\partial \boldsymbol{P}}, \tag{5}$$

$$\boldsymbol{p}(\boldsymbol{q}, \boldsymbol{P}) = \frac{\partial F(\boldsymbol{q}, \boldsymbol{P})}{\partial \boldsymbol{q}}. \tag{6}$$

For the intended application of kernel regression, representing a linear term is not favorable. This is why we split the generating function into a sum,

$$F(\boldsymbol{q}, \boldsymbol{P}) = \boldsymbol{q} \cdot \boldsymbol{P} + \tilde{F}(\boldsymbol{q}, \boldsymbol{P}). \tag{7}$$

The first part $\boldsymbol{q} \cdot \boldsymbol{P}$ in Eq. (7) describes the identity transformation $\boldsymbol{q} \to \boldsymbol{Q}$, $\boldsymbol{p} \to \boldsymbol{P}$. The relation between $(\boldsymbol{q}, \boldsymbol{p})$ and $(\boldsymbol{Q}, \boldsymbol{P})$ can then be written as

$$\begin{pmatrix} \nabla_q \tilde{F}(\boldsymbol{q}, \boldsymbol{P}) \\ \nabla_P \tilde{F}(\boldsymbol{q}, \boldsymbol{P}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}(\boldsymbol{q}, \boldsymbol{P}) - \boldsymbol{P} \\ \boldsymbol{Q}(\boldsymbol{q}, \boldsymbol{P}) - \boldsymbol{q} \end{pmatrix} = \begin{pmatrix} -\Delta \boldsymbol{p}(\boldsymbol{q}, \boldsymbol{P}) \\ \Delta \boldsymbol{q}(\boldsymbol{q}, \boldsymbol{P}) \end{pmatrix}. \tag{8}$$

In the limit of small mapping times, the Hamiltonian H can be identified (up to a constant) with $\tilde{F}$, as the time evolution of canonical coordinates can be represented by an infinitesimal canonical transformation. Specifically, from Eq. (8), we obtain the following

Chaos ARTICLE scitation.org/journal/cha

expressions for $(\boldsymbol{Q}, \boldsymbol{P})$:

$$\boldsymbol{Q} = \boldsymbol{q} + \frac{\partial \tilde{F}}{\partial \boldsymbol{P}}, \tag{9}$$

$$\boldsymbol{P} = \boldsymbol{p} - \frac{\partial \tilde{F}}{\partial \boldsymbol{q}}. \tag{10}$$

This can be compared by the equations of motion in Eq. (3), where the first order approximation yields a symplectic Euler integration step,

$$\boldsymbol{Q} \approx \boldsymbol{q} + h\frac{\partial H}{\partial \boldsymbol{P}}, \tag{11}$$

$$\boldsymbol{P} \approx \boldsymbol{p} - h\frac{\partial H}{\partial \boldsymbol{q}}. \tag{12}$$

Comparing those sets of equations yields the relation $\tilde{F} = Hh$ up to an irrelevant constant shift, where h is the mapping time step.

III. REGRESSION OF HAMILTONIAN FLOW MAPS

A. Multi-output GPs and derivative observations

A Gaussian process (GP)[29] is a stochastic process with the convenient property that any finite marginal distribution of the GP is Gaussian. For $\boldsymbol{x} \in \mathbb{R}^d$, a GP with mean $\boldsymbol{m}(\boldsymbol{x})$ and kernel or covariance function $K(\boldsymbol{x}, \boldsymbol{x}')$ is denoted as

$$\boldsymbol{f}(\boldsymbol{x}) \sim \mathcal{GP}(\boldsymbol{m}(\boldsymbol{x}), K(\boldsymbol{x}, \boldsymbol{x}')), \tag{13}$$

where we allow vector-valued functions.[33] In contrast to the single output case, where the random variables are associated with a single process for $f(\boldsymbol{x}) \in \mathbb{R}$, a multi-output GP for $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^D$ consists of random variables associated with different and generally correlated processes. The covariance function is a positive semidefinite matrix-valued function whose entries $(K(\boldsymbol{x}, \boldsymbol{x}'))_{ij}$ express the covariance between the output dimensions i and j of $\boldsymbol{f}(\boldsymbol{x})$. In case a linear model for the mean $\boldsymbol{m}$ with some functional basis φ_i and unknown coefficients is used, a modified Gaussian process follows according to Chap. 2.7 of Rasmussen and Williams.[29]

For regression via a GP, we assume that the observed function values $Y \in \mathbb{R}^{D\times N}$ may contain local Gaussian noise ϵ; i.e., the noise is independent at different position $\boldsymbol{x}$ but may be correlated between components of $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) + \epsilon$. The input variables are aggregated in the $d \times N$ design matrix X, where N is the number of training data points. After observing Y, the posterior mean $F_* \equiv \mathbb{E}(F(X_*))$ and covariance evaluated for test data X_* are given analytically by

$$F_* = K(X_*, X)(K(X, X) + \Sigma_n)^{-1} Y, \tag{14}$$

$$\operatorname{cov}(F_*) = K(X_*, X_*) - K(X_*, X)(K(X, X) + \Sigma_n)^{-1} K(X, X_*), \tag{15}$$

where $\Sigma_n \in \mathbb{R}^{ND\times ND}$ is the covariance matrix of the multivariate output noise for each training data point. In the simplest case, it is diagonal with entries σ_n^2. Estimation of kernel parameters and σ_n^2 given the input data are usually performed via optimization or sampling according to the marginal log-likelihood.[29]

When a linear operator $\mathcal{L}$, e.g., differentiation, is applied to the Gaussian process, this yields a new Gaussian process,[28,30,31]

$$\mathcal{L}\boldsymbol{f}(\boldsymbol{x}) \sim \mathcal{GP}(\boldsymbol{l}(\boldsymbol{x}), L(\boldsymbol{x}, \boldsymbol{x}')). \tag{16}$$

Here, the mean $\boldsymbol{l}(\boldsymbol{x})$ is given by $\boldsymbol{l}(\boldsymbol{x}) = \mathcal{L}\boldsymbol{m}(\boldsymbol{x})$ and a matrix-valued gradient kernel

$$L(\boldsymbol{x}, \boldsymbol{x}') = (\mathcal{L}_{\boldsymbol{x}} \otimes \mathcal{L}_{\boldsymbol{x}'})K(\boldsymbol{x}, \boldsymbol{x}') = \mathcal{L}_{\boldsymbol{x}} K(\boldsymbol{x}, \boldsymbol{x}')\mathcal{L}_{\boldsymbol{x}'}^T \tag{17}$$

follows, where $\mathcal{L}_{\boldsymbol{x}'}^T$ is applied from the right to yield an exterior product.[42]

As differentiation is a linear operation, in particular the gradient of a Gaussian process over scalar functions $g(\boldsymbol{x})$ with kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ remains a Gaussian process. The result is a multi-output GP where the covariance matrix is the Hessian of $K(\boldsymbol{x}, \boldsymbol{x}')$ containing all second derivatives in $(\boldsymbol{x}, \boldsymbol{x}')$. A joint GP describing both values and gradients is given by

$$\begin{pmatrix} g(\boldsymbol{x}) \\ \nabla g(\boldsymbol{x}) \end{pmatrix} \sim \mathcal{GP}(\boldsymbol{n}(\boldsymbol{x}), K(\boldsymbol{x}, \boldsymbol{x}')), \tag{18}$$

with $\boldsymbol{n}(\boldsymbol{x}) = (m(\boldsymbol{x}), \boldsymbol{l}(\boldsymbol{x}))^T$ and where

$$K(\boldsymbol{x}, \boldsymbol{x}') = \begin{pmatrix} k(\boldsymbol{x}, \boldsymbol{x}') & k(\boldsymbol{x}, \boldsymbol{x}')\nabla_{\boldsymbol{x}'}^T \\ \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{x}') & \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{x}')\nabla_{\boldsymbol{x}'}^T \end{pmatrix} \tag{19}$$

contains $L(\boldsymbol{x}, \boldsymbol{x}')$ as the lower-right block. In the more general case of a linear operator $\mathcal{L}$ in place of ∇, one may use the joint GP in Eq. (19) as a *symmetric meshless formulation*[43,44] to find approximate solutions of the according linear (partial differential) equation.

B. Symplectic GP regression

To apply GP regression on symplectic maps, we use Eq. (19) for the joint distribution of the generating function and its gradients in Eq. (8),

$$\begin{pmatrix} \tilde{F}(\boldsymbol{q}, \boldsymbol{P}) \\ \partial_{\boldsymbol{q}} \tilde{F}(\boldsymbol{q}, \boldsymbol{P}) \\ \partial_{\boldsymbol{P}} \tilde{F}(\boldsymbol{q}, \boldsymbol{P}) \end{pmatrix} \sim \mathcal{GP}(\boldsymbol{n}(\boldsymbol{q}, \boldsymbol{P}), K(\boldsymbol{q}, \boldsymbol{P}, \boldsymbol{q}', \boldsymbol{P}')), \tag{20}$$

with

$$K(\boldsymbol{q}, \boldsymbol{P}, \boldsymbol{q}', \boldsymbol{P}') = \begin{pmatrix} k & \partial_{\boldsymbol{q}'} k & \partial_{\boldsymbol{P}'} k \\ \partial_{\boldsymbol{q}} k & \partial_{\boldsymbol{q}\boldsymbol{q}'} k & \partial_{\boldsymbol{q}\boldsymbol{P}'} k \\ \partial_{\boldsymbol{P}} k & \partial_{\boldsymbol{P}\boldsymbol{q}'} k & \partial_{\boldsymbol{P}\boldsymbol{P}'} k \end{pmatrix}. \tag{21}$$

We cannot observe the generating function $\tilde{F}(\boldsymbol{q}, \boldsymbol{P})$, but it is determined up to an additive constant via the predictor

$$\tilde{F}_* = \begin{pmatrix} \partial_{\boldsymbol{q}} k(X_*, X) \\ \partial_{\boldsymbol{P}} k(X_*, X) \end{pmatrix} (L(X, X) + \Sigma_n)^{-1} Y, \tag{35}$$

where column i of X for the ith training orbit is composed of rows

$$x_{1\ldots f,\, i} = \boldsymbol{q}_i \quad \text{and} \quad x_{(f+1)\ldots 2f,\, i} = \boldsymbol{P}_i \tag{36}$$

and similarly for X_* and test points. Columns of Y contain

$$y_{1\ldots f,\, i} = -\Delta \boldsymbol{p}_i = \partial_{\boldsymbol{q}} \tilde{F}(\boldsymbol{q}_i, \boldsymbol{P}_i), \tag{37}$$

$$y_{(f+1)\ldots 2f,\, i} = \Delta \boldsymbol{q}_i = \partial_{\boldsymbol{P}} \tilde{F}(\boldsymbol{q}_i, \boldsymbol{P}_i). \tag{38}$$

Chaos ARTICLE scitation.org/journal/cha

The matrix L denotes the lower block

$$L(\boldsymbol{q},\boldsymbol{P},\boldsymbol{q}',\boldsymbol{P}') = \begin{pmatrix} \partial_{\boldsymbol{q}\boldsymbol{q}'}k & \partial_{\boldsymbol{q}\boldsymbol{P}'}k \\ \partial_{\boldsymbol{P}\boldsymbol{q}'}k & \partial_{\boldsymbol{P}\boldsymbol{P}'}k \end{pmatrix}. \tag{39}$$

This also allows one to learn the Hamiltonian H from Eq. (22) as for sufficiently small mapping times H can be approximated by $\tilde{F}$ (up to a constant).

For further investigations on temporal evolution of the Hamiltonian system and the construction of symplectic maps, we are interested in the gradients of $\tilde{F}$ via the block L. The predictive mean for this GP's output is given by

$$\begin{pmatrix} -\Delta\boldsymbol{p}_* \\ \Delta\boldsymbol{q}_* \end{pmatrix} = L(X_*,X)(L(X,X)+\Sigma_n)^{-1}\begin{pmatrix} -\Delta\boldsymbol{p} \\ \Delta\boldsymbol{q} \end{pmatrix}. \tag{40}$$

The symplecticity condition for predictors $\boldsymbol{p}_* = \boldsymbol{P}_* - \Delta\boldsymbol{p}_*(\boldsymbol{q}_*,\boldsymbol{P}_*)$ and $\boldsymbol{Q}_* = \boldsymbol{q}_* + \Delta\boldsymbol{q}_*(\boldsymbol{q}_*,\boldsymbol{P}_*)$ holds according to Eq. (4): The derivatives of linear terms $\boldsymbol{P}_*$ and $\boldsymbol{q}_*$ vanish, and by using Eq. (27), the remaining derivatives enter upper and lower rows of $L(X_*,X)$, respectively,

$$\begin{aligned} \frac{\partial\Delta\boldsymbol{q}_*}{\partial\boldsymbol{q}_*} - \frac{\partial\Delta\boldsymbol{p}_*}{\partial\boldsymbol{P}_*} &\propto \frac{\partial}{\partial\boldsymbol{q}_*}\left(\frac{\partial^2 k}{\partial\boldsymbol{P}_*\partial\boldsymbol{q}}, \frac{\partial^2 k}{\partial\boldsymbol{P}_*\partial\boldsymbol{P}}\right) \\ &\quad - \frac{\partial}{\partial\boldsymbol{P}_*}\left(\frac{\partial^2 k}{\partial\boldsymbol{q}_*\partial\boldsymbol{q}}, \frac{\partial^2 k}{\partial\boldsymbol{q}_*\partial\boldsymbol{P}}\right). \end{aligned} \tag{41}$$

Due to symmetry of partial derivatives, the expected value of the symplecticity condition in Eq. (28) is identically zero; therefore, the predictive mean of Eq. (27) produces a symplectic map. Due to the mixing of initial and final coordinates by the generating function of the canonical transformation, we can generally not predict $\boldsymbol{Q}_*$ and $\boldsymbol{P}_*$ for a given $\boldsymbol{q}_*,\boldsymbol{p}_*$ right away. In the symplectic GP regression of this map, depending on the choice of the kernel, two cases have to be considered:

a. *(Semi-)implicit method.* In the general case with a generating function $\tilde{F}(\boldsymbol{q},\boldsymbol{P})$, equations for $\boldsymbol{P}_*$ in Eq. (27) are implicit and have to be solved iteratively as indicated in Algorithm 1. This corresponds to the implicit steps of a symplectic Euler integrator in a non-separable system.[6]
b. *Explicit method.* When considering a generating function in a separable form $\tilde{F}(\boldsymbol{q},\boldsymbol{P}) = V(\boldsymbol{q}) + T(\boldsymbol{P})$, the resulting transformation equations reduce to

$$\boldsymbol{p}(\boldsymbol{q},\boldsymbol{P}) = \frac{\partial\tilde{F}(\boldsymbol{q},\boldsymbol{P})}{\partial\boldsymbol{q}} = \frac{\partial V(\boldsymbol{q})}{\partial\boldsymbol{q}}, \tag{42}$$

$$\boldsymbol{Q}(\boldsymbol{q},\boldsymbol{P}) = \frac{\partial\tilde{F}(\boldsymbol{q},\boldsymbol{P})}{\partial\boldsymbol{P}} = \frac{\partial T(\boldsymbol{P})}{\partial\boldsymbol{P}}, \tag{43}$$

resulting in a simplified lower block $L(\boldsymbol{q},\boldsymbol{P},\boldsymbol{q}',\boldsymbol{P}')$ with off-diagonal elements equal to 0,

$$L(\boldsymbol{q},\boldsymbol{P},\boldsymbol{q}',\boldsymbol{P}') = \begin{pmatrix} \partial_{\boldsymbol{q}\boldsymbol{q}'}k & 0 \\ 0 & \partial_{\boldsymbol{P}\boldsymbol{P}'}k \end{pmatrix}. \tag{44}$$

This choice of generating function results in a simplified construction and application of the symplectic map as explained in Algorithm 2. This corresponds to the case of a separable Hamiltonian where the symplectic Euler method becomes fully explicit.

In Fig. 2, we illustrate training data for an application example of the standard map. Initial conditions at $t = 0$ [Fig. 2(a)] and final conditions at $t = h$ [Fig. 2(b)] are displayed on a regular grid (q,p). Those serve as input data for the regular GP used in Algorithm 1 whose prediction is needed for the initial guess for the Newton iterations. In Figs. 2(c) and 2(d), the training data are displayed on a mixed grid (q,P) and (Q,p). In both algorithms, (q,P) data are assembled in the design matrix, and (Q,p) data serve as observations

Algorithm 1: (Semi-)implicit symplectic GP map

Construction:
Step 1: Usual GP regression of $\boldsymbol{P}$ over initial $(\boldsymbol{q},\boldsymbol{p})$
Step 2: Symplectic GP regression of $-\Delta\boldsymbol{p}$ and $\Delta\boldsymbol{q}$ over mixed variables $(\boldsymbol{q},\boldsymbol{P})$ according to

$$\begin{pmatrix} -\Delta\boldsymbol{p} \\ \Delta\boldsymbol{q} \end{pmatrix} \sim \mathcal{GP}(\boldsymbol{I}(\boldsymbol{q},\boldsymbol{P}), L(\boldsymbol{q},\boldsymbol{P},\boldsymbol{q}',\boldsymbol{P}')) \tag{32}$$

Application:
Step 3: Initial guess $\boldsymbol{P}_*(\boldsymbol{q}_*,\boldsymbol{p}_*)$ from GP of Step 1
Step 4: Solve implicit equation in $\boldsymbol{P}_*$ via

$$\Delta\boldsymbol{p}_*(\boldsymbol{q}_*,\boldsymbol{P}_*) - (\boldsymbol{P}_* - \boldsymbol{p}_*) = 0, \tag{33}$$

predicting $\Delta\boldsymbol{p}_*$ via Eq. 27 from symplectic GP of Step 2.
Step 5: Explicitly evaluate

$$\boldsymbol{Q}_* = \boldsymbol{q}_* + \Delta\boldsymbol{q}_*(\boldsymbol{q}_*,\boldsymbol{P}_*), \tag{34}$$

predicting $\Delta\boldsymbol{q}_*$ via Eq. 27 from symplectic GP of Step 2.

Chaos ARTICLE scitation.org/journal/cha

Algorithm 2: Explicit symplectic GP map

Construction:
Step 1: GP regression of $-\Delta \boldsymbol{p}$ and $\Delta \boldsymbol{q}$ over mixed variables $(\boldsymbol{q}, \boldsymbol{P})$ according to

$$\begin{pmatrix} -\Delta \boldsymbol{p} \\ \Delta \boldsymbol{q} \end{pmatrix} \sim \mathcal{GP}(\boldsymbol{I}(\boldsymbol{q}, \boldsymbol{P}), L(\boldsymbol{q}, \boldsymbol{P}, \boldsymbol{q}', \boldsymbol{P}')) \tag{35}$$

Application:
Step 2: Solve explicit equation in $\boldsymbol{P}_*$ via

$$\boldsymbol{P}_* = \boldsymbol{p}_* + \Delta \boldsymbol{p}_*(\boldsymbol{q}_*), \tag{36}$$

with $\Delta \boldsymbol{p}_*(\boldsymbol{q}_*)$ predicted from GP of Step 1.
Step 3: Evaluate

$$\boldsymbol{Q}_* = \boldsymbol{q}_* + \Delta \boldsymbol{q}_*(\boldsymbol{P}_*), \tag{37}$$

with $\Delta \boldsymbol{q}_*(\boldsymbol{P}_*)$ predicted from GP of Step 1.

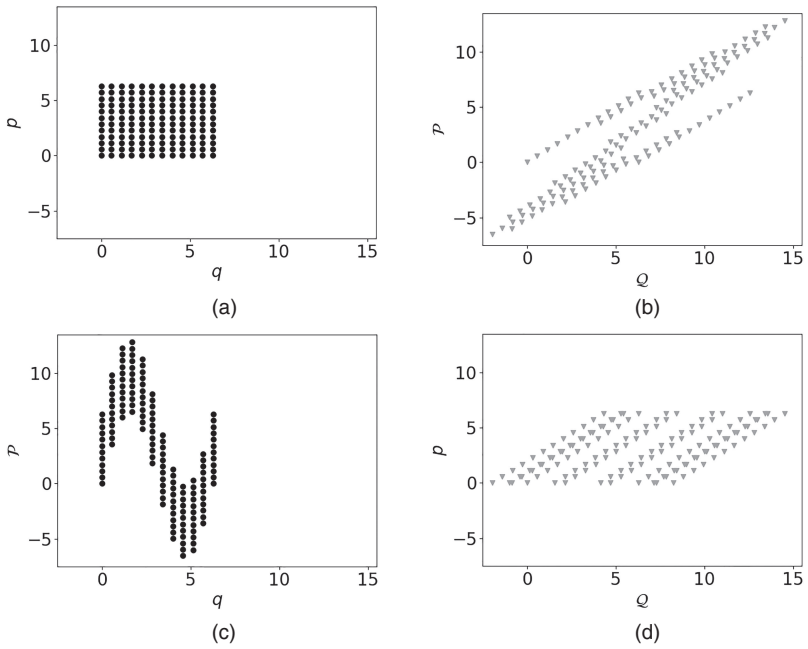


FIG. 2. Example of training data for a standard map: For illustration purposes, the training data points ($N = 144$) are sampled on a grid $[0, 2\pi] \times [0, 2\pi]$ for a stochasticity parameter $K = 6.6$—initial and final conditions shown in panels (a) and (b) serve as input and observations for the GP regression of P. Mixed variables in (c) serve as input and in (d) take the role of observations for the symplectic GP regression.

Chaos ARTICLE scitation.org/journal/cha

for the GP regression. Once the model is fitted and the covariance matrix is calculated, it is used to predict subsequent time steps.

Remarks on universality

Due to the universal approximation property of radial basis function expansions,[45] one may expect that in principle any flow that can be represented by a generating function can be approximated arbitrarily well by a symplectic GP. Indeed, for a number of kernel functions, especially the squared exponential kernel, GPs with scalar output are universal approximators.[46] However, the matrix-valued kernel $L(\boldsymbol{x},\boldsymbol{x}')$ of Eq. (19) cannot approximate arbitrary vector fields[47] in dimensions higher than one. Equation (26) shows this in an alternative manner for even-dimensional vector fields. The used GP can at most be a universal approximator for vector fields with a vanishing exterior derivative that are locally given as gradient fields. In particular, this includes curl-free vector fields in 3D and Hamiltonian vector fields in even dimensions with sign-flipped canonical components flipped by a symplectic matrix. Universality of a kernel approximation is equivalent[46] to universality of the underlying feature set $\{\phi_i(\boldsymbol{x})\}$ in the Mercer representation,

$$k(\boldsymbol{x},\boldsymbol{x}') = \sum_{i=1}^{\infty} \phi_i(\boldsymbol{x})\bar{\phi}_i(\boldsymbol{x}'). \tag{38}$$

It is easy to see that the matrix kernel $L(\boldsymbol{x},\boldsymbol{x}') = \nabla_x k(\boldsymbol{x},\boldsymbol{x}')\nabla_{x'}^T$ for gradient observations has the equivalent representation in features $\boldsymbol{\psi}_i(\boldsymbol{x}) = \nabla\phi_i(\boldsymbol{x})$ in case of sufficient convergence. In the extended framework of matrix-valued multi-task kernels,[47] $L(\boldsymbol{x},\boldsymbol{x}')$ should therefore be able to represent any vector field in the span of this gradient feature set required for Hamiltonian vector fields in the present application. Regarding convergence of regression results with an increasing number of training points, and theoretical errors are quickly masked by numerical accuracy due to bad conditioning of the kernel matrix.[43] This limits the practical use of theoretical estimates on this topic. As we will see below, non-unique-valued generating functions put additional limits on learning the flow maps even when gradient fields are approximated well.

IV. NUMERICAL EXPERIMENTS

In this section, we present the application of implicit and explicit symplectic Gaussian process regression (SympGPR) to unperturbed and perturbed Hamiltonian systems. We consider separable and non-separable autonomous Hamiltonian systems and use SympGPR to approximate Hamiltonian flows as well as Poincaré maps. We show the applicability of the proposed method to chaotic systems for the standard map and finally apply it to the magnetic field in a tokamak with a non-axisymmetric perturbation. In all application examples, the kernel hyperparameters are adjusted by maximizing the likelihood[29] of the training data using the L-BFGS-B routine implemented in Python.[48] Only for the more sophisticated application example of magnetic field line tracing in a non-axisymmetric perturbed tokamak, CMA-ES[49,50] is used to optimize the hyperparameters.

A. Hamiltonian flow maps: Pendulum

The pendulum is a nonlinear oscillator with position q and momentum p and exhibits two kinds of motion: libration and rotation, which are separated by the separatrix in the phase space. The system of a pendulum corresponds to a particle in a cosine potential.[5] The Hamiltonian is given by

$$H(q,p) = \frac{p^2}{2} + U_0(1-\cos(q)), \tag{39}$$

where we fix $U_0 = 1$. The underlying periodic topology suggests the choice of a periodic kernel function, which is universal for periodic functions, in q with periodicity 2π, $k_q(q,q_i) \propto \exp\left(-\frac{\sin^2((q-q_i)/2)}{2l_q^2}\right)$ and a squared exponential kernel function in p, $k_P(P,P_i) \propto \exp\left(-\frac{(P-P_i)^2}{2l_P^2}\right)$. For the product kernel $k(q,q_i,P,P_i) = \sigma_f^2 k_q(q,q_i)k_P(P,P_i)$ used in the implicit method, the noise in observations σ_n^2 [Eq. (15)] is set to 10^{-16}, whereas for the sum kernel $k(q,q_i,P,P_i) = \sigma_f^2(k_q(q,q_i) + k_P(P,P_i))$ for the explicit method, we set $\sigma_n^2 = 10^{-10}$ for numerical stability. The hyperparameters, l_q and l_P that correspond to the length scales in q and p, respectively, are set to optimized maximum likelihood values by minimizing the negative log-likelihood.[29] The scaling of the kernel σ_f^2 that quantifies the amplitude of the fit is set in accordance with the observations to $2\max(|Y|)^2$, where Y corresponds to the training output. We note here that in the following, we assume that the periodicity of the system is known. If this is not the case, the periodic kernel can easily be equipped with another hyperparameter to estimate the periodicity that is optimized during the training of the GP. Initial investigations showed that when assuming that $k_q(q,q_i) \propto \exp\left(-\frac{\sin^2((q-q_i)\tau)}{2l_q^2}\right)$, where τ is the periodicity hyperparameter, the implicit SympGPR is able to estimate integer multiples of the period accurately, leading to results of similar quality as shown in Fig. 3.

To evaluate the implicit and explicit SympGPR for the pendulum flow map, we use $N = 20$ initial data points sampled from a Halton sequence[51] within the range $q \in [0, 2\pi]$ and $p \in [-3, 3]$ and integrate them until $t = h$ using an RK45 integrator[52] with an adaptive step size and a relative tolerance of 10^{-13}, leading to results at machine accuracy. Each pair of initial and final conditions (as shown in Fig. 2) constitutes one sample of the training data set. The results for $n = 2000$ subsequent applications of the map are shown in Figs. 3 and 4 for a step size of $h = 0.2$ and $h = 0.07$, respectively, in direct comparison with reference data calculated using an RK45 integrator with an adaptive step size and a relative tolerance of 10^{-13}, leading to results at machine accuracy. The 15 test data points are randomly selected nodes of a regular grid within the range $q \in [\pi - 2.8, \pi + 1.5]$ and $p \in [-2.3, 1.8]$ with $\Delta q = 0.31$ and $\Delta p = 0.29$. Even for a small number of training data points, two kinds of motion can be perfectly distinguished by both methods; also, the motion near the separatrix is stable in the presented test cases. The obtained results for the implicit SympGPR (Fig. 3) generalize to similar step sizes as shown later in the numerical benchmark. As apparent in Fig. 4, the orbits produced by the explicit SympGPR are slightly deformed even for a very small step size, indicating a

Chaos ARTICLE scitation.org/journal/cha

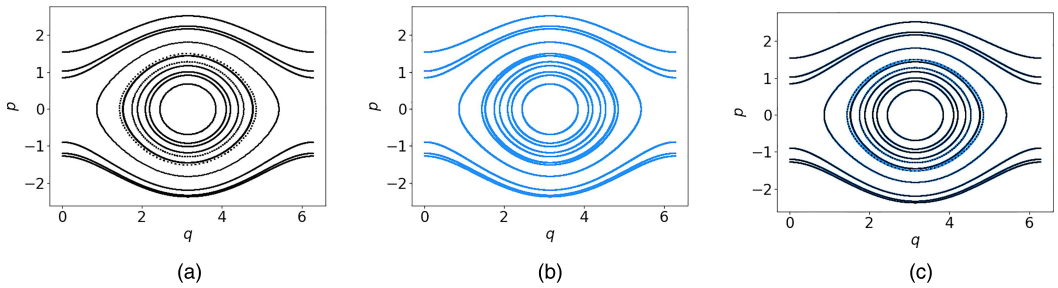


FIG. 3. Pendulum orbits in the phase space. *Implicit* SympGPR trained with $N = 20$ training data points for a step size $h = 0.2$ (a), reference orbits (b), and direct comparison of both flows (c) for 15 test data points and $n = 2000$ subsequent applications of the map.

lower accuracy. This effect becomes more severe for larger step sizes as shown later in the numerical benchmark (see Sec. IV B).

In Fig. 5, three different symplectic integration methods using a larger step size $h = 0.9$ are compared to reference orbits generated with an RK45 integrator at machine accuracy [Fig. 5(a)]. As the implicit SympGPR (as presented in Algorithm 1) mimics an implicit–explicit symplectic Euler, we use a symplectic Euler integrator and the Störmer–Verlet scheme with the same step size to calculate orbits for the same 15 test data points. When using the implicit SympGPR [Fig. 5(b)], large-scale features in the phase space can still be reproduced, and small stochastic layers and islands emerging due to the perturbation from numerical errors remain confined. There is no tilt in the phase space apparent contrary to forward integration by a usual symplectic Euler scheme [Fig. 5(c)]. Here, the distinction between trapped and passing orbits is not as clear as for the implicit SympGPR. In Fig. 5(d), the results obtained by using the Störmer–Verlet scheme are shown. As this is a symmetric scheme, there is no tilt in the phase space observable. Also, small islands are apparent. By comparing the obtained orbits, it is evident that the presented method, which is of order 1, can compete with symplectic integration methods of order 2. However, we have to point out once more that the explicit knowledge of the Hamiltonian or its derivatives is not needed when using (explicit or implicit) SympGPR in contrast to forward integration using a symplectic Euler scheme. Alternatively, observations of the initial conditions (q, p) at $t = 0$ and final conditions (Q, P) at time $t = h$ are used to construct the Hamiltonian flow map over the finite time step h.

Due to the enforced symplecticity of the SympGPR map, we benefit from structure-preservation, long-term stability, and conservation of invariants of motion within fixed bounds. This is shown by calculating the total energy given in Eq. (39) over $n = 10\,000$ subsequent mapping applications for one test orbit. It may be seen in Fig. 6 that the energy is conserved and varies within restricted bounds.

The Hamiltonian function H given in Eq. (39) can be learned from the initial (q, p) and final state (Q, P) using Eq. (22). In contrast to earlier proposed methods,[17] derivatives of H are not needed explicitly as the training data consist only of observable states of the dynamical system in time. In Fig. 7, the Hamiltonian function calculated exactly from Eq. (39) is compared to the approximation using

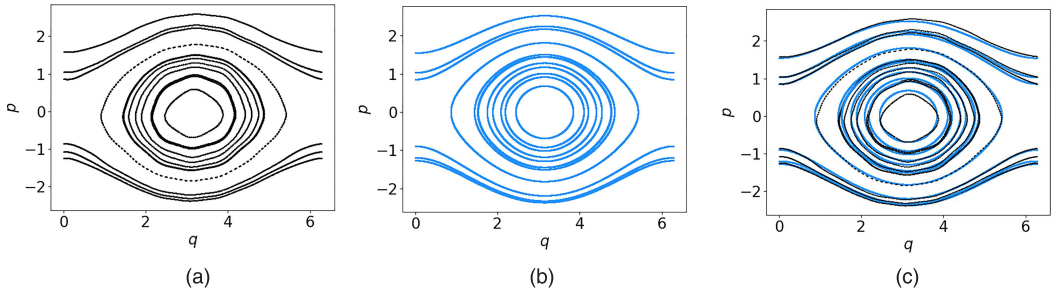


FIG. 4. Phase space plot of the pendulum comparing *explicit* SympGPR trained with $N = 20$ training data points for a step size $h = 0.07$ (a), reference orbits (b), and direct comparison of both flows (c) for 15 test data points and $n = 2000$ subsequent applications of the map.

Chaos ARTICLE scitation.org/journal/cha

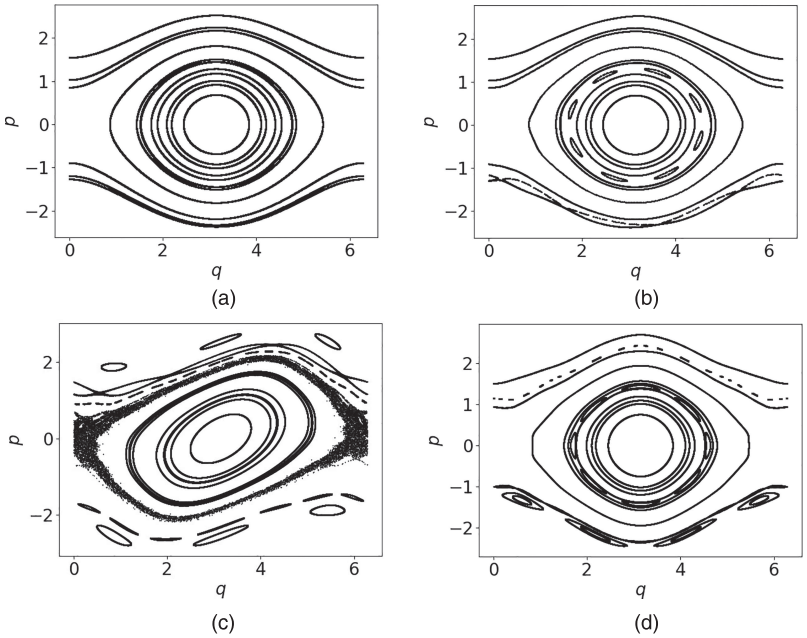


FIG. 5. Phase space plot of the pendulum for 15 test data points and $n = 2000$ subsequent applications of the map comparing reference orbits (a), implicit SympGPR (b) trained with $N = 50$ training data points with step size $h = 0.9$, symplectic Euler integrator (c), and Störmer–Verlet integrator (d) with step size $h = 0.9$.

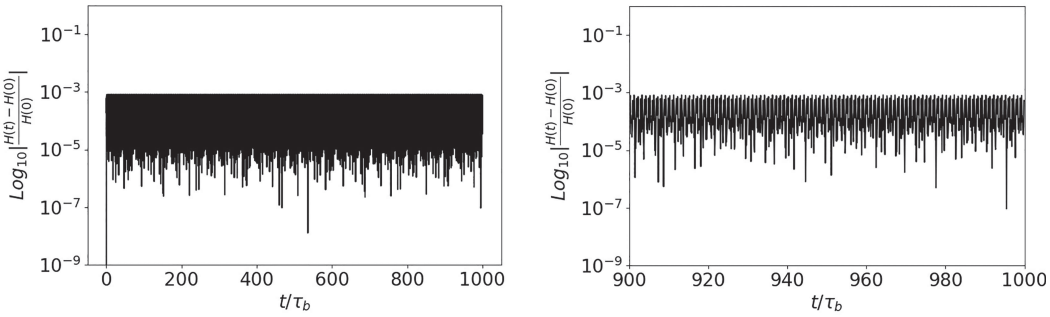


FIG. 6. Relative energy error $\log_{10}|\frac{H(t)-H(0)}{H(0)}|$ of the pendulum for one test data point $(q, p) = (1, 0)$ and 1000 bounce times calculated via the implicit SympGPR map [panel (b): detailed zoom]. The model was trained using $N = 20$ training data points with $h = 0.5$. The horizontal axis is given by t/τ_b, where τ_b is the bounce time. A similar quality of energy conservation within fixed bounds is generally realized.

Chaos ARTICLE scitation.org/journal/cha

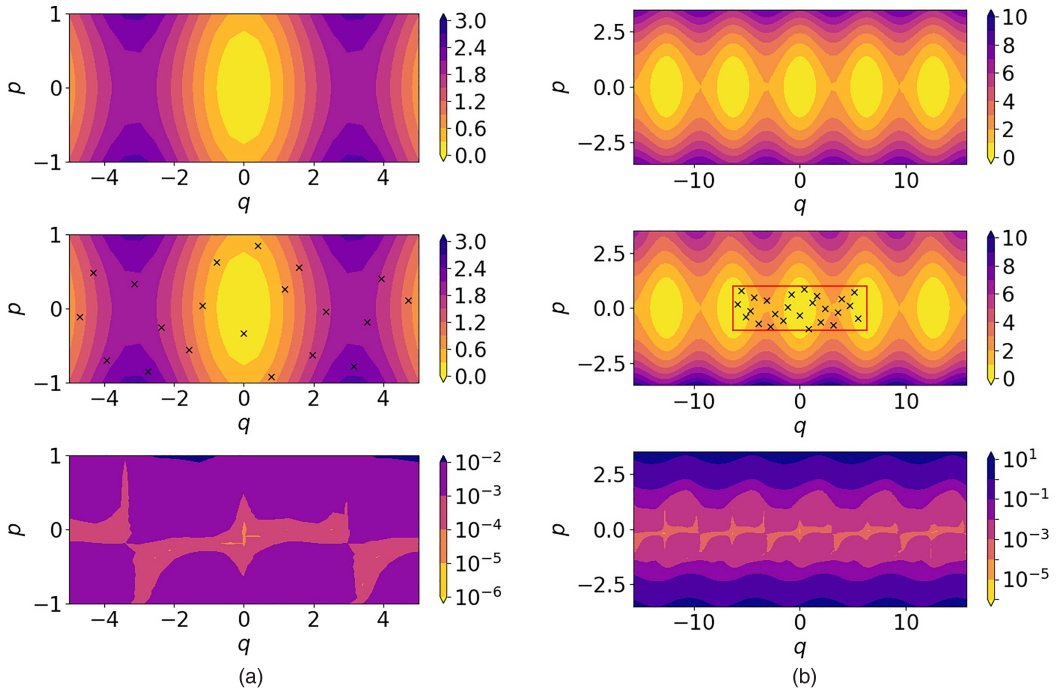


FIG. 7. Upper: Hamiltonian function of the pendulum [Eq. (39)]. Middle: Approximation using implicit SympGPR of the generating function via Eq. (22) for $h = 0.01$. The 25 training points are denoted by $\times$. Lower: Absolute difference between the analytical solution and approximation. Excellent interpolation performance (a) and surprisingly good extrapolation (b) are observed.

the implicit symplectic GP method for $h = 0.01$ trained with 25 initial and the corresponding final conditions sampled from a Halton sequence[51] within the range $q \in [-2\pi, 2\pi]$ and $p \in [-1.0, 1.0]$. The approximation is validated using 5625 random points within the same range. Using the mean squared error to evaluate the losses, we get for the training loss 1.3×10^{-5} and for the test loss 6.3×10^{-5}. As evident in Fig. 7, a certain degree of extrapolation is possible in areas close to the range of the training data in p, whereas evaluation at arbitrary q is possible without extrapolation due to periodicity of the system and the kernel function. For bigger step sizes $h = 0.2$, the training loss is 1.5×10^{-3} and the test loss 2.3×10^{-3} producing still valid results within the training region.

B. Numerical benchmark

We compare implicit SympGPR (Algorithm 1), explicit SympGPR (Algorithm 2), implicit symmetric regression with a spectral basis (see Appendix A), and semi-implicit symplectic Euler integration for the test case of a pendulum. Non-symplectic regression of flow maps is not compared, as it is inherently unstable over the considered time intervals (see Fig. 1).

To assess the quality and stability of the proposed mapping methods, two quality measures are used. The geometric distance is computed to compare the first application of the constructed map step to the respective time step of a reference orbit in the phase space. This phase space distance is given by

$$g_d = \sqrt{(\boldsymbol{q} - \boldsymbol{q}_{\mathrm{ref}})^2 + (\boldsymbol{p} - \boldsymbol{p}_{\mathrm{ref}})^2}, \tag{40}$$

where $\boldsymbol{q}_{\mathrm{ref}}$ and $\boldsymbol{p}_{\mathrm{ref}}$ denote the reference orbits and $\boldsymbol{q}$ and $\boldsymbol{p}$ the mapped orbits. The reference orbits are calculated using an adaptive step size RK45 scheme with relative tolerance 10^{-13} and absolute tolerance 10^{-16}.

Even though energy is preserved on average, we can measure normalized oscillations given by

$$E_{\mathrm{osc}} = \frac{\mathrm{Std}(\bar{H})}{\bar{H}}, \tag{41}$$

Chaos ARTICLE scitation.org/journal/cha

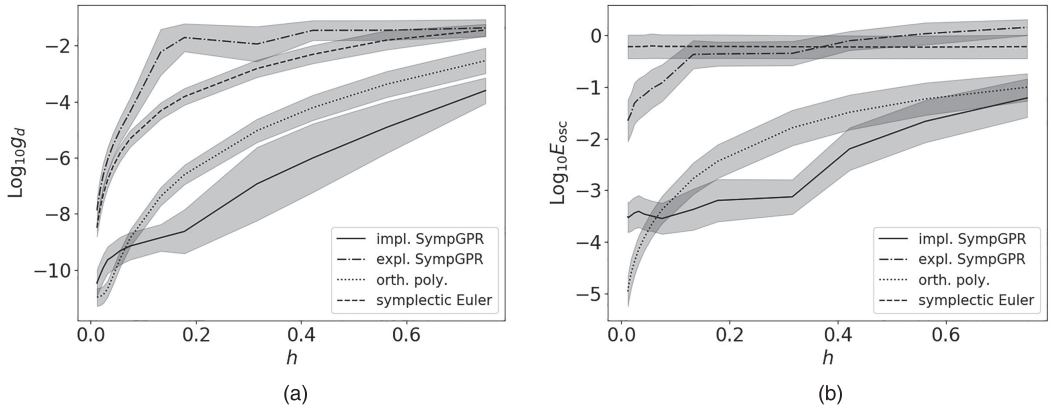


FIG. 8. Pendulum: Comparison of the geometrical distance [Eq. (40)] (a) and normalized energy oscillations [Eq. (41)] (b) of implicit and explicit SympGPR, standard symplectic Euler and Fourier–Hermite basis functions for a fixed number of training points $N = 15$ and a variable step size h. Gray areas correspond to the standard deviation for 100 test points.

where $\bar{H}$ is the mean, to serve as a criterion for mapping quality. Here, periodic energy oscillations are averaged over $n = 300$ subsequent applications of the map.

In Fig. 8, the four methods are compared for the one dimensional pendulum using the quality criteria given in Eqs. (40) and (41) for a fixed number of training points $N = 15$ but a variable step size h. As expected, the geometric distance g_d as well as the energy oscillation E_{osc} are increasing for a increasing step size. Since no Newton iterations are needed, the explicit SympGPR method is faster than the implicit method in its region of validity. As for the first guess for Newton's method, a separate GP is used as indicated in Algorithm 1; less than five Newton iterations are typically necessary in the implicit case.

As discussed above, the orbits in the phase space resulting from the explicit SympGPR are deformed, which explains the bad performance regarding the geometrical distance. For smaller mapping times, the deformation and, therefore, also the energy oscillation reduce. This is in accordance with similar behavior of explicit–implicit Euler integration schemes.

Spectral linear regression produces very accurate results for very small mapping times, as the interpolated data (the change in coordinates) are almost 0, and the generating function inherits the polynomial structure of the Hamiltonian H that can be fitted exactly. At larger mapping times, implicit SympGPR and spectral methods perform similarly.

To investigate the behavior with increasing number of training data points N, in Fig. 9, the quality measures are compared for fixed step size h but with a variable number of training points. The implicit methods improve considerably with N. The visible steps for the implicit method with a spectral basis arise from the used number of modes that depends on N. The explicit SympGPR does not improve with N due to the deformation of orbits in the phase space, which is an inherent structural feature of the forced splitting into a sum kernel that cannot be fixed by adding more training data. This unstable behavior is also observable in other application systems (see Appendix C) (separable and non-separable) for step sizes $h > 0.1$. We conclude that a separable approximation via a sum kernel is not competitive when compared to the implicit SympGPR that is able to deal with much larger step sizes.

Also, from the comparison with an usual symplectic Euler integrator, we conclude that the implicit SympGPR reaches a significantly higher accuracy, whereas the explicit SympGPR performs worse when trying to represent the flow map. This is why we use the implicit SympGPR when fitting Poincaré maps rather than flow maps in the subsequent investigations. Depending on the application, desired accuracy, and also the cost of generating training data, Figs. 8 and 9 in connection with Tables I and II serve as guidance on how to choose step size h and training data N. Additionally, for large N, also, the conditioning of the covariance matrix has to be considered that usually gets worse with increasing number of N. Hence, more regularization is necessary. For the application case of the unperturbed pendulum, the square root of machine precision is reached at $h = 0.5$ and $N \approx 20$. Certainly, also other combinations of those parameters result in highly accurate predictions; e.g., $N = 15$ and $h = 0.1$ or $N = 30$ and $h = 0.6$ achieve results of similar quality. For the estimation of computational complexity and runtime, we compared the performance of implicit SympGPR with a standard symplectic Euler scheme at a similar prediction quality. Hence, we chose $N = 20$ training data points with a step size of $h = 0.1$ for the prediction using the implicit SympGPR, whereas for the symplectic Euler, a step size of $h = 0.002$ is needed to achieve a similar geometrical distance of 10^{-12}. To predict the trajectory of one orbit with initial conditions $(q, p) = (0.35, 0.5)$ for 20 bounce times, the implicit SympGPR takes (after training) 1.3 s, whereas the

Chaos ARTICLE scitation.org/journal/cha

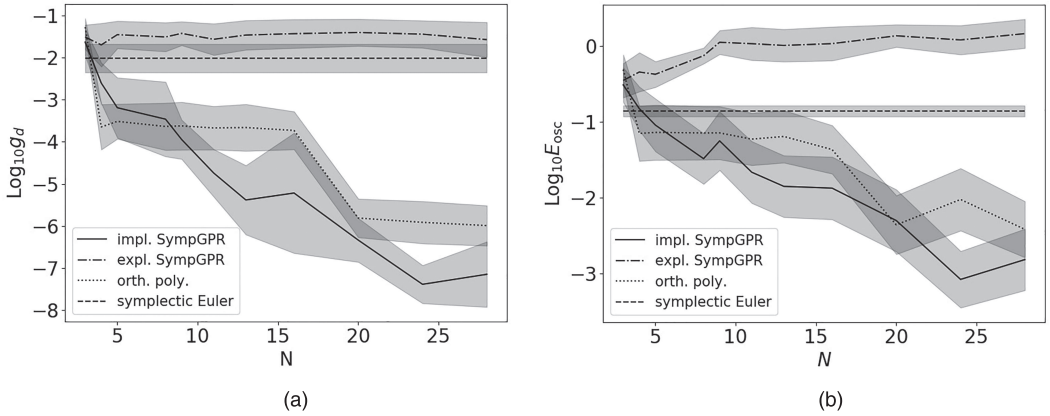


FIG. 9. Pendulum: Comparison of the geometrical distance [Eq. (40)] (a) and normalized energy oscillations [Eq. (41)] (b) of implicit and explicit SympGPR, standard symplectic Euler and Fourier–Hermite basis functions for a variable number of training points N and a fixed step size of $h = 0.5$. Gray areas correspond to the standard deviation for 100 test points.

symplectic Euler integrator needs 0.3 s. One should keep in mind that in this special application case, Hamilton's equations are available in a (simple) analytical form, which is not the case in usual practical applications (see Sec. IV C 4).

C. Poincaré maps

1. Perturbed pendulum

To show the applicability of the proposed method for a more complicated system and also larger step sizes, we consider approximating Poincaré maps between surface sections in the phase space. For this purpose, we consider the Hamiltonian of a perturbed pendulum

$$H(q,p,\phi) = \frac{1}{2}p^2 - \omega^2\cos(q) - \epsilon(0.3qp\sin(2\phi) + 0.7qp\sin(3\phi)), \tag{42}$$

where $\omega = 0.5$ and $\epsilon = 0.5$. This system has been used by Burby *et al.*[18] to demonstrate the interpolation of Poincaré maps by structure-preserving artificial neural networks. Here, we use a similar training setup with a much lower number of $N = 50$ initial conditions sampled from a Halton sequence within a disk of $r \leq 0.9$ meaning that we consider only orbits inside the separatrix. The Poincaré map to obtain the corresponding final conditions for the section at $\phi = 2\pi$ is generated by a RK4 approximation used by Burby *et al.*[18] with 1500 steps. Again, we use a periodic kernel function in q and a squared exponential kernel function in p. The hyperparameters l_q, l_P are set to optimized maximum likelihood values. The noise in observations σ_n^2 is set to 10^{-12} and σ_f^2 is set in accordance with the observations to $2\max(|Y|)^2$, where Y corresponds to the change in coordinates. 20 test points along the x axis and 10 test points with constant $q = \pi$ are chosen. Applying the implicit SympGPR map $n = 2000$ times results in the plot of Fig. 10 with reference orbits from RK4 with 100 steps.[18] Despite the small number of training data points, the dynamics in the phase space are extremely well captured. Also, the structure of the islands is preserved.

2. Hénon–Heiles potential

The Hénon–Heiles system is a classical example of a nonlinear Hamiltonian system with $f = 2$ degrees of freedom and a 4D phase space.[5,53] The corresponding Hamiltonian is given by

$$H(\boldsymbol{q},\boldsymbol{p}) = \frac{1}{2}\left(q_1^2 + q_2^2\right) + \frac{1}{2}\left(p_1^2 + p_2^2\right) + \lambda\left(q_1^2 q_2 - \frac{1}{3}q_2^3\right), \tag{43}$$

where usually $\lambda = 1$. The underlying potential continuously varies from a harmonic potential for small values of q_1 and q_2 to triangular equipotential lines on the edges. For energies lower than the limiting potential energy $H_{\text{esc}} = 1/6$, the orbit is trapped within the potential. However, for larger energies, three escape channels appear due to the special shape of the potential, through which the orbit may escape.[54] Therefore, the training and test data are set to a restricted area in the phase space in order to keep the motion bounded within the potential.

Here, a squared exponential kernel function is used in all dimensions, where the hyperparameter l is set to its optimized maximum likelihood value. The noise in observations σ_n^2 [Eq. (15)] is set to 10^{-8} for the implicit SympGPR. As in the pendulum case, σ_f^2 is set in accordance with the observations to $2\max(|Y|)^2$, where Y corresponds to the change in coordinates. The Hamiltonian function [Eq. (43)] is learned from 50 initial conditions sampled from a Halton sequence[51] in the range $q_1, q_2, p_1, p_2 \in [-0.5, 0.5]$ and the corresponding final states $(\boldsymbol{Q}, \boldsymbol{P})$ calculated using an RK45 integrator with

Chaos ARTICLE scitation.org/journal/cha

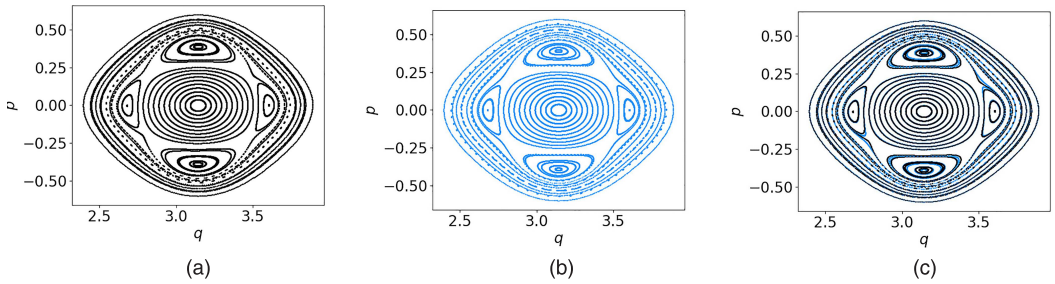


FIG. 10. Poincaré plot of the perturbed pendulum at $\phi = 2n\pi$ comparing implicit SympGPR trained with $N = 50$ training data points (a), reference orbits (b), and direct comparison of both Poincaré maps (c) for 30 test data points and $n = 2000$ subsequent applications of the map.

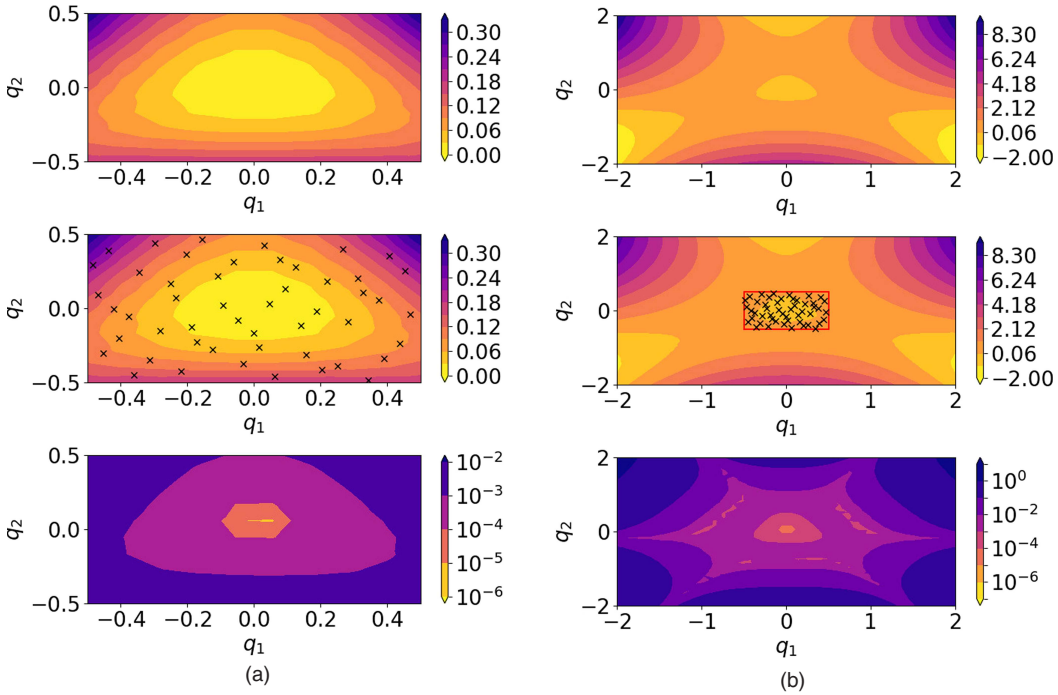


FIG. 11. Upper: Hamiltonian function of the Hénon–Heiles system [Eq. (43)]. Middle: Approximation using implicit SympGPR of the generating function via Eq. (22) for $h = 0.01$. The projection of the 50 training points onto the plane $\boldsymbol{p} = 0$ is denoted by $\times$. Lower: Absolute difference between the analytical solution and the approximation. As in the pendulum case, the Hamiltonian function H is interpolated well (a) and even extrapolated well (b) relatively far from the training region.

Chaos ARTICLE scitation.org/journal/cha

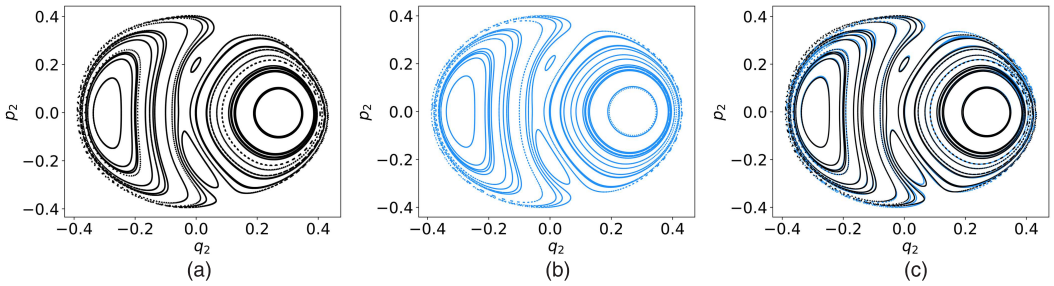


FIG. 12. Poincaré plots ($q_1 = 0, p_1 > 0$) of the Hénon–Heiles system for $E = 1/100$ comparing implicit SympGPR trained with $N = 55$ training data points (a), reference orbits calculated using a RK4 integrator (b), and a direct comparison of both Poincaré plots (c) for 37 test data points and $n = 2000$ subsequent applications of the map.

an adaptive step size and a relative tolerance of 10^{-13} using Eq. (22) for a fixed step size $h = 0.01$. In Fig. 11, the Hamiltonian function calculated from Eq. (43) is compared to the approximation using GPs. The approximation is validated using 46 656 points within the same range as the training points. Using the mean squared error to evaluate the losses, we get for the training loss 6.9×10^{-5} and for the test loss 8.1×10^{-5}. Similarly to the pendulum, the result within the training region generalizes also to bigger step sizes up to $h = 0.2$.

When observing particles for a fixed, sufficiently small energy in the Hénon–Heiles potential with various initial conditions crossing the $p_2 - q_2$ surface of section, invariant curves can be observed.[5] When fixing the energy, the dimensionality of the problem is reduced to 3, and only the intersections of the trajectories for $q_1 = 0$ are plotted (for $p_1 > 0$). In the following numerical experiments, $q_1 = 0$, and, as the energy is fixed, p_1 is given by $p_1 = \sqrt{2E - q_2^2 - p_2^2 + \frac{2}{3}q_2^3}$. Here, we consider Poincaré sections for $E = 1/100$ of $N = 55$ initial conditions sampled from a Halton sequence within $q_2 \in [-0.15, 0.15]$ and $p_2 \in [-0.15, 0.15]$ that are calculated using an adaptive ODE integrator at machine accuracy. The initial conditions are integrated until $q_1 = 0$ is satisfied, which results in a step size h (bounce time) in the range of $[6.18, 6.34]$. Therefore, we do not use the same step size h but the same surface of section. For better convergence in the optimization routine of the hyperparameters, where the Python implementation of L-BFGS-B[48] is used, the training data are rescaled by a factor 10^2 in order that the input to the optimizer is of order 1. For the approximation of the Poincaré plot using the implicit SympGPR, the noise in observations σ_n^2 is fixed to 10^{-12} and σ_f^2 is set in accordance with the observations to $2\max(|Y|)^2$. The hyperparameter l is set to its optimized maximum likelihood value. The 37 test points are randomly selected nodes on a grid within $q_2 \in [-0.1, 0.1]$ and $p_2 \in [-0.1, 0.1]$ with $\Delta q_2 = \Delta p_2 = 0.0055$. The map is applied $n = 2000$ times, producing the Poincaré plot shown in Fig. 12. Again, the comparison to the reference Poincaré plot shows that the dynamics of the system are well captured by the symplectic GP. As expected, four stable elliptic points are reproduced by the implicit SympGPR.

For energies $E > 1/50$, the generating function is multivalued as illustrated in Fig. 13, and therefore, it is not possible to uniquely predict points in the phase space without further measures. This problem occurs for highly non-linear systems for large time steps, e.g., when mapping between Poincaré sections that are too far apart. It is possible to split one mapping step into several sub-steps, which will be demonstrated for magnetic field lines in a tokamak in Sec. IV C 4. This, however, is not possible for Poincaré sections for the Hénon–Heiles system as no additional surfaces for the splitting can be identified. This results from the fact that turning points of the orbit can be arbitrarily close to the surface of section $q_1 = 0$. To tackle the non-uniqueness in this case, an unwinding transformation for the generating function onto a subspace has to be identified that allows a unique prediction by the GP. This possibility will be investigated in future work.

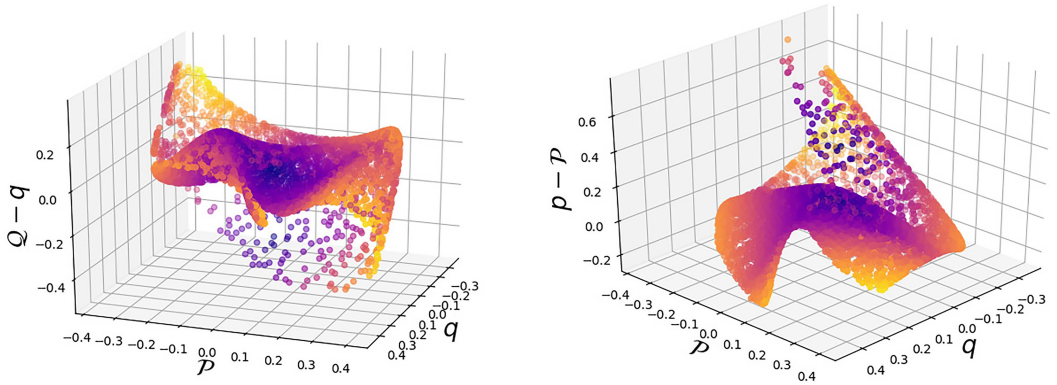
3. Standard map

The standard map,[55]

$$I_{n+1} = (I_n + K\sin\theta_n) \bmod 2\pi, \quad \theta_{n+1} = (\theta_n + I_{n+1}) \bmod 2\pi, \quad (44)$$

in action-angle variables (I, θ) is a well studied model to investigate chaos in Hamiltonian systems. The mapping steps correspond to the Poincaré sections of a periodically kicked rotator. The stochasticity parameter K represents the intensity of the perturbation. Here, the action appears as the momentum, and the angle corresponds to the position in the presented Algorithms 1 and 2. The behavior in the phase space suggests a periodic kernel function in θ but a squared exponential kernel function in I. Here, we consider the mapping of one iteration of the standard map only. This case is well-suited for validation purposes, as the availability of a closed-form expression permits analytical estimates of chaotic diffusion but does not influence the GP's performance. The initial conditions ($N = 20$) are sampled from a Halton sequence in the range $[0, 2\pi] \times [0, 2\pi]$. The noise in observations is set to $\sigma_n^2 = 10^{-8}$, σ_f^2 is set in accordance with the observations to $2\max(|Y|)^2$, where Y corresponds to the change in coordinates, and the hyperparameters l_I, l_θ are set to their maximum likelihood value. The 18 test data points are randomly

Chaos ARTICLE scitation.org/journal/cha



(a) (b)

FIG. 13. Multivalued change in coordinates (a) $\Delta q = \partial_P \tilde{F}(q, P)$ and (b) $-\Delta p = \partial_q \tilde{F}(q, P)$ for Hénon–Heiles Poincaré sections for $E = 1/12$. The twisted non-unique surfaces cannot be interpolated by the presented regression methods without modifications. The colors are scaled via the function $|q| + |P|$ in order to indicate the multivaluedness in a way where same coordinates (q, P) show the same color.

selected nodes on a grid within the range $[0, 2\pi] \times [0, 2\pi]$ with $\Delta I = \Delta\theta = 0.37$. Here, we give results for individual values of K, whereas a more detailed analysis of K-dependent chaotic diffusion is presented below.

In Fig. 14, the resulting phase space plots are shown for different values of K and $n = 1000$ subsequent applications of the implicit SympGPR map. When compared to the reference solution, the GP map reproduces the essential features of the standard map: the fixed points and contractible periodic orbits are clearly visible, and the regions of stochasticity are confined and only occur near separatrices [Fig. 14, panels (a)–(c)]. The onset of global stochasticity is reproduced accurately for increased perturbation strength for $K < K_{\text{crit}} \approx 0.971635\ldots$ [Fig. 14, panels (d)–(f)]. With increasing K shown in Fig. 14, panels (g)–(i), the chaotic region covers progressively more phase space.[5] Also, for higher values of $K = 6.6$ and a very limited number of training data points ($N = 30$) in the whole range $[0, 2\pi] \times [0, 2\pi]$, the SympGPR map is able to reproduce accelerator mode islands[55] around a period-4 periodic orbit at $(\theta, I) \approx (4.4, \pi)$. Accelerator modes occur within some intervals of $K \geq 2\pi$ and are stable regions in the phase space, where I is changing monotonically with time. Orbits trapped within those islets of stability[55] are ballistically transported and contribute to the periodic variation of the diffusion rate as examined below. As shown in Fig. 15 for 18 test points sampled from a Halton sequence within $[4.46, 4.52] \times [3.20, 3.39]$ and $n = 1000$ applications of the map, the island is surrounded by four accelerator mode islands that remain confined in the phase space and do not drift into the chaotic region surrounding the island.

As the standard map is given in an explicit form, the explicit SympGPR performs better than the implicit SympGPR. Especially for high values of K, the implicit SympGPR needs more training data than the explicit SympGPR to perform similarly well. However, the required number of training data points for the implicit SympGPR does not exceed $N = 30$ for the considered perturbations.

To evaluate the quality of the presented model for chaotic systems such as the standard map, it is crucial that the diffusion of chaotic orbits is reproduced correctly. As studied analytically and numerically,[56–63] the standard map exhibits anomalous (ballistic) diffusion (with a diffusion exponent of $1 < \beta < 2$) due to islands of stability surrounding accelerator mode fixed points in addition to normal (Brownian) diffusion with $\beta = 1$. Also, the diffusion coefficient D,

$$D = \lim_{n\to\infty} \frac{\langle (I_n - I_0)^2 \rangle}{2n}, \tag{45}$$

where the average is taken over a large ensemble of initial conditions, is oscillating for $K > K_{\text{crit}}$, where the last KAM surface is destroyed.[5] The theoretical diffusion rate[5,56] to order K^{-1} is

$$D = \frac{K^2}{2}\left(\frac{1}{2} - \mathcal{J}_2(K) - \mathcal{J}_1^2(K) + \mathcal{J}_2^2(K) + \mathcal{J}_3^2(K)\right), \tag{46}$$

where $\mathcal{J}_i$ is the Bessel function of the first kind. In Fig. 16, the observed values of the diffusion coefficient given in Eq. (45) resulting from $n = 300$ subsequent applications of the map are presented. The initial conditions for $N = 5000$ particles are sampled from a uniform distribution in the range $[0, 2\pi] \times [0, 2\pi]$. As expected, the presence of accelerator modes in some intervals of the stochasticity parameter K leads to anomalous diffusion resulting in sharp peaks.

To investigate the transport in time, the mean displacement of a particle ensemble near an accelerator mode island for $K = 6.8$

Chaos ARTICLE scitation.org/journal/cha

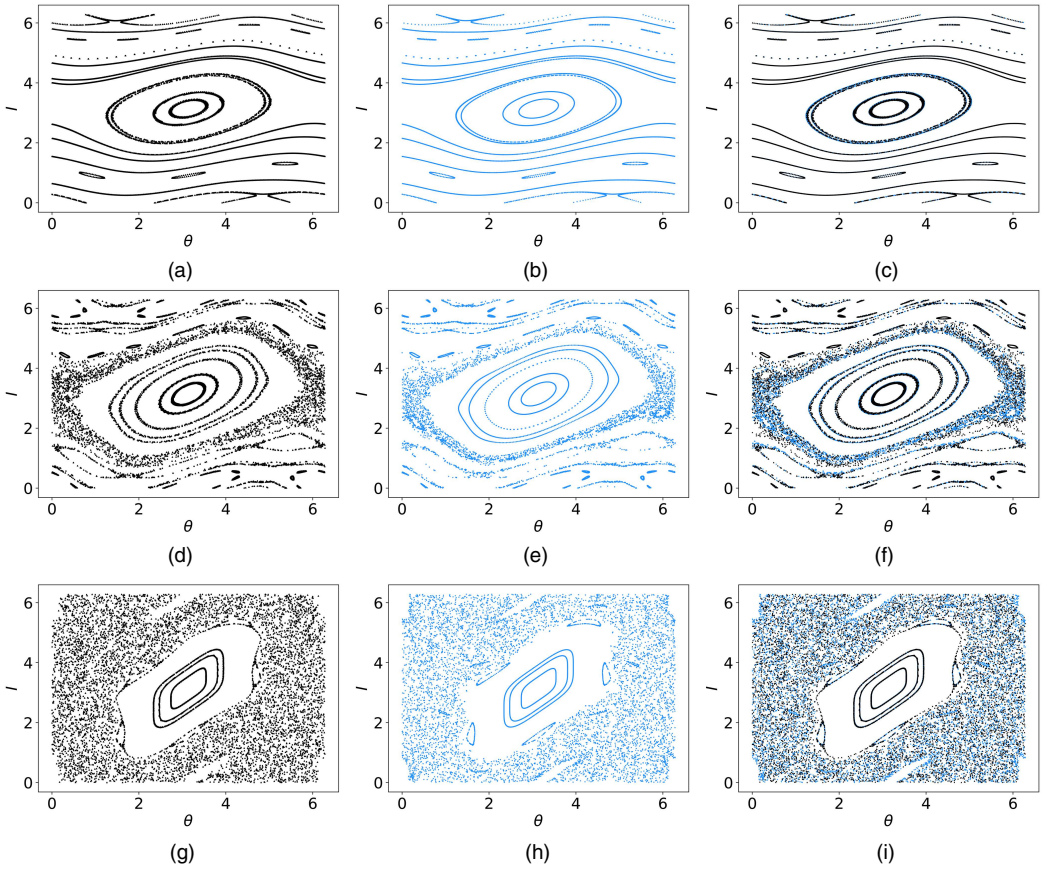


FIG. 14. Standard map for $K = 0.5$ [(a)–(c)], $K = 1.0$ [(d)–(f)], and $K = 2.0$ [(g)–(i)]. Implicit SympGPR trained with $N = 20$ [(a), (d), and (g)], reference solution [(b), (e), and (h)], and direct comparison of implicit SympGPR and reference solution [(c), (f), and (i)] for 18 test data points and $n = 1000$ subsequent applications of the map.

is simulated[61] to examine the type of diffusion. For $K = 6.8$, there exist two accelerator mode islands in the phase space surrounded by an extreme sticky region. For $N = 5000$ initial conditions, uniformly distributed in a box of size $(10^{-5} \times 10^{-5})$ at $(\theta, I) = (0.3 \cdot 2\pi, 0.1123)$, within the sticky region of one stable island, the mean displacement in the action coordinate is calculated. As illustrated in the right plot in Fig. 16, the diffusion exponent $\mu \approx 2$ corresponds to anomalous diffusion as long as the orbits are dragged by the accelerator mode in a ballistic motion. Then, the type of diffusion changes to normal diffusion ($\mu = 1$) for a large number of map applications. This behavior is reproduced accurately by the implicit and also explicit SympGPR map.

4. Magnetic field lines in a perturbed tokamak

As a final application example, we consider motion along magnetic field lines in a toroidal magnetic configuration with vector potential $\boldsymbol{A}$. By a special choice (see Appendix B) of spatial coordinates (r, ϑ, φ), it is possible to identify equations of motion according to Eq. (B7) as

$$\frac{\mathrm{d}\vartheta}{\mathrm{d}\varphi} = -\frac{\partial A_\varphi(\vartheta, p_\vartheta, \varphi)}{\partial p_\vartheta} = -\frac{\partial A_\varphi(r, \vartheta, \varphi)}{\partial r}\left(\frac{\partial A_\vartheta(r, \vartheta, \varphi)}{\partial r}\right)^{-1}, \tag{47}$$

Chaos ARTICLE scitation.org/journal/cha

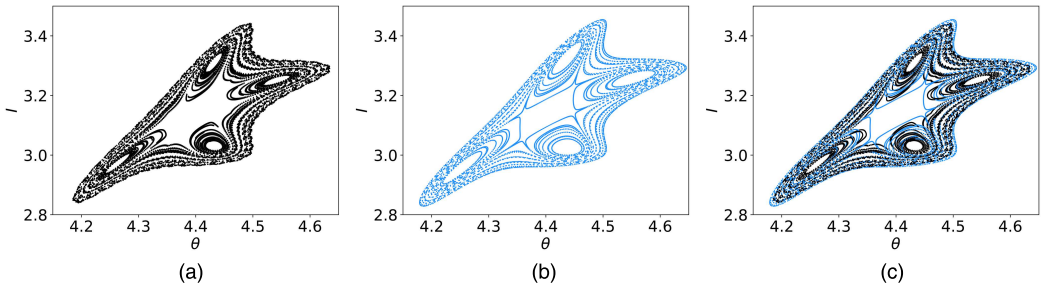


FIG. 15. Accelerator mode for $K = 6.6$ around $(\theta, I) \approx (4.4, \pi)$. Implicit SympGPR trained with $N = 30$ training data points within the range $[0, 2\pi] \times [0, 2\pi]$ (a), reference solution (b), and direct comparison of implicit SympGPR and reference solution (c) for 18 test data points and $n = 1000$ subsequent applications of the map.

$$\frac{\mathrm{d}p_\vartheta}{\mathrm{d}\varphi} = \frac{\partial A_\varphi(\vartheta, p_\vartheta, \varphi)}{\partial \vartheta} = \frac{\partial A_\varphi(r, \vartheta, \varphi)}{\partial \vartheta} - \frac{\partial A_\varphi(r, \vartheta, \varphi)}{\partial r} \left(\frac{\partial A_\vartheta(r, \vartheta, \varphi)}{\partial r} \right)^{-1} \frac{\partial A_\vartheta(r, \vartheta, \varphi)}{\partial \vartheta}. \tag{48}$$

The equations above are written in both canonical and non-canonical phase space variables. Interpolation and application of the map are performed in canonical coordinates (ϑ, p_ϑ), while evaluation points of $\boldsymbol{A}$ require non-canonical (r, ϑ). This means that in addition to usual computations, the relation

$$p_\vartheta = A_\vartheta(r, \vartheta, \varphi) \tag{49}$$

has to be solved implicitly in r.[41] For numerical tests, we consider an axisymmetric model field

$$A_\vartheta(r, \vartheta) = B_0 \left(\frac{r^2}{2} - \frac{r^3}{3R_0} \cos\vartheta \right), \quad A_\varphi(r) = -\iota_0 B_0 \left(\frac{r^2}{2} - \frac{r^4}{4a^2} \right) \tag{50}$$

to which we apply a non-axisymmetric perturbation with an additional dependency on φ. This is modeled as a Hamiltonian perturbation in a similar manner to Eder *et al.*,[64]

$$\delta A_\varphi(r, \vartheta, \varphi) = \varepsilon A_\varphi(r) \cos(m\vartheta + n\varphi). \tag{51}$$

For the numerical experiments, the perturbation mode $m = -3$, $n = 2$ was used as this corresponds to the main mode of a resonant

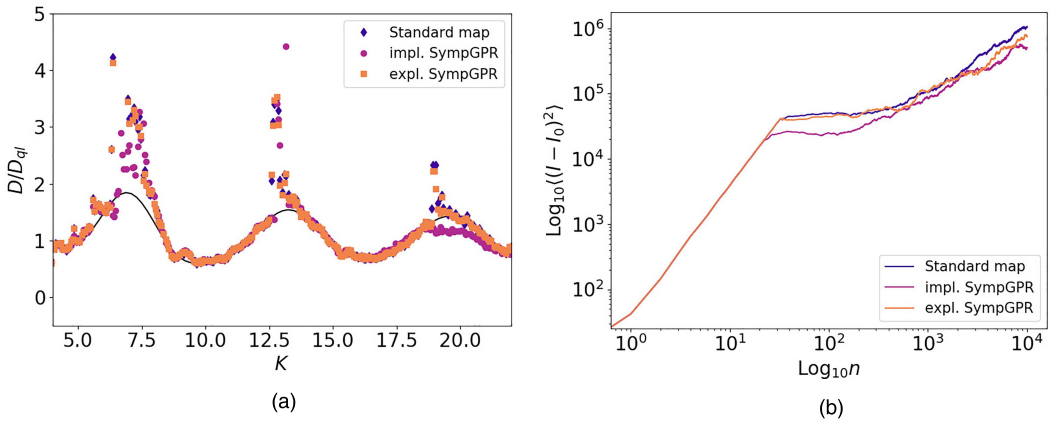


FIG. 16. (a) Theoretical diffusion rate [Eq. (46)] normalized with the quasi-linear approximation $D_{ql} = K^2/4$ (solid line) compared with simulations for implicit and explicit SympGPR (trained with $N = 20$) and the reference solution for 5000 test data points and $n = 300$ map applications. (b) Mean displacement in the action variable I as a function of the number of map applications n for a set of 50 initial conditions near an accelerator mode island for $K = 6.8$.

Chaos ARTICLE scitation.org/journal/cha

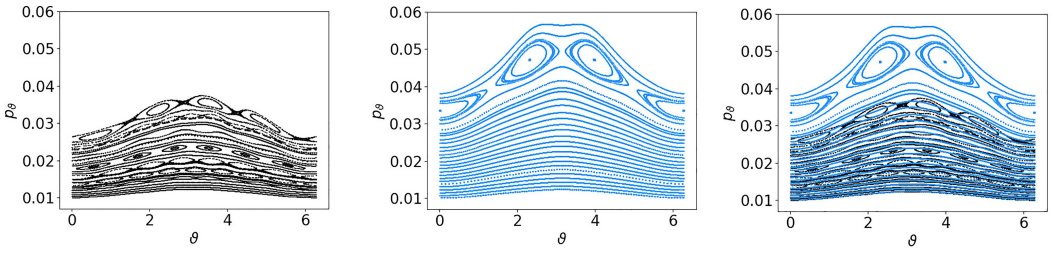


FIG. 17. Poincaré maps of magnetic field lines in a tokamak with non-axisymmetric perturbations in the phase space at $\varphi = 2n\pi$ for a perturbation strength of $\varepsilon = 0.001$ comparing implicit SympGPR trained with $N = 80$ training data points (a), reference orbits calculated using a symplectic Euler integrator with $h = 0.01$ (b), and direct comparison of both Poincaré maps (c) for 30 test data points and for $n = 1000$ map applications.

magnetic perturbation. Due to the dynamics of the system, a squared exponential kernel is used in p_ϑ, whereas ϑ is modeled by a periodic kernel function. We sampled $N = 80$ initial conditions in non-canonical coordinates from a Halton sequence from $r \in [0.1, 0.36]$ and $\vartheta \in [0, 2\pi]$. The Poincaré map for $\varphi = 2n\pi$ with integer n was numerically calculated using a symplectic Euler scheme with $h = 0.01$. Using Eq. (49), the training data points were transformed into canonical coordinates (p_ϑ, ϑ) to which the implicit SympGPR is applied. For better convergence in the optimization routine, p_ϑ is rescaled by 10^2. The noise in observations is set to $\sigma_n^2 = 10^{-12}$; the hyperparameters, l_q, l_p, σ_f^2, are set to their optimized maximum likelihood value. The implicit SympGPR map is evaluated for 30 test data points and are randomly selected on a regular grid within the range $r \in [0.15, 0.25]$ and $\vartheta \in [0, 2\pi]$ with $\Delta r = 0.003$ and $\Delta\vartheta = 0.22$ at $\varphi = 0$. For a perturbation strength $\varepsilon = 0.001$, the generating function is smooth and uniquely valued in the training region and can, therefore, be grasped by the implicit SympGPR. However, the derivatives of the generating function are infinitely steep for $r > 0.37$ and, therefore, cannot be represented by the SympGPR map as the Newton solver does not converge. Therefore, it is only possible to use a restricted region in the phase space to which the SympGPR map can be applied as apparent in Fig. 17.

For more stable results in a larger phase space region and also for higher perturbation strengths, e.g., $\varepsilon = 0.01$, it is favorable to split the Poincaré map into several sub-steps to obtain more stable results and also to circumvent multivalued generating functions. We introduced four independent implicit SympGPR maps corresponding to four sub-steps, each representing a leap of $\varphi = \pi/2$. Each GP was trained with $N = 70$ training data points whose initial conditions (r, ϑ) were sampled from a Halton sequence from $r \in [0.1, 0.48]$ and $\vartheta \in [0, 2\pi]$ and integrated for a leap of $\varphi = \pi/2$. The hyperparameters are set to their optimized maximum likelihood value for each GP separately; the noise in observations is set to $\sigma_n^2 = 10^{-8}$. We validate the map using 30 test data points randomly selected on a grid within the range $r \in [0.16, 0.31]$ and $\vartheta \in [0, 2\pi]$ with $\Delta r = 0.005$ and $\Delta\vartheta = 0.22$ at $\varphi = 0$. The result of each sub-step, corresponding to a leap of $\varphi = \pi/2$, was then consequently used as input for the next sub-step. While this approach leads to a greater number of needed training data and more (offline) training time as four independent GPs and also the corresponding GPs for the first guess of the Newton iteration have to be trained, the gained results are more stable and accurate. This approach is applicable to higher perturbation strengths where the generating function is not sufficiently smooth or non-unique for a full Poincaré section. As shown in the upper plots in Fig. 18 for $\varepsilon = 0.001$, the dynamics in the phase space are accurately captured by the split implicit SympGPR map. The three stable elliptic points are grasped correctly. The stochastic layer surrounding the island chain is represented accurately, and also, other periodic orbits close to the magnetic axis are not broken up into island chains.

For a higher perturbation strength, $\varepsilon = 0.01$ [Figs. 18(d)–18(f)], where the region of stochasticity covers a large region in the phase space, the dynamics of the system are correctly reproduced by the split implicit SympGPR map. Again, three islands are visible, each of which is surrounded by five smaller islands. Here, the implicit SympGPR map is trained with $N = 110$ training data points whose initial conditions (r, ϑ) were sampled from a Halton sequence from $r \in [0.1, 0.48]$ and $\vartheta \in [0, 2\pi]$. The hyperparameters are set to their optimized maximum likelihood value for each GP separately; the observation noise is set to $\sigma_n^2 = 10^{-8}$. We validate the map on 30 test data points selected randomly on a grid with $r \in [0.16, 0.31]$ and $\vartheta \in [0, 2\pi]$ with $\Delta r = 0.005$ and $\Delta\vartheta = 0.22$ at $\varphi = 0$.

To estimate the influence of the number of sub-steps and the number of training data points N on the performance, an analysis was carried out using the example of $\varepsilon = 0.001$ with training data sampled from a Halton sequence from $r \in [0.1, 0.48]$ and $\vartheta \in [0, 2\pi]$ and test data in the range $r \in [0.16, 0.31]$ and $\vartheta \in [0, 2\pi]$ and using up to six sub-steps. Again, perturbation modes $m = -3$ and $n = 2$ were used. As shown in Fig. 19(a), the geometrical distance decreases with increasing N up to a fivefold split. A sixfold split does not improve the obtained accuracy. The geometrical distance saturates at a relatively small number $N \approx 50$ for the three- and sixfold split. This is also apparent in the energy oscillations [Fig. 19(b)]. However, when splitting the SympGPR into four or five sub-steps, increasing N still improves the geometric distance. Naturally, the number of sub-steps influence the efficacy

Chaos ARTICLE scitation.org/journal/cha

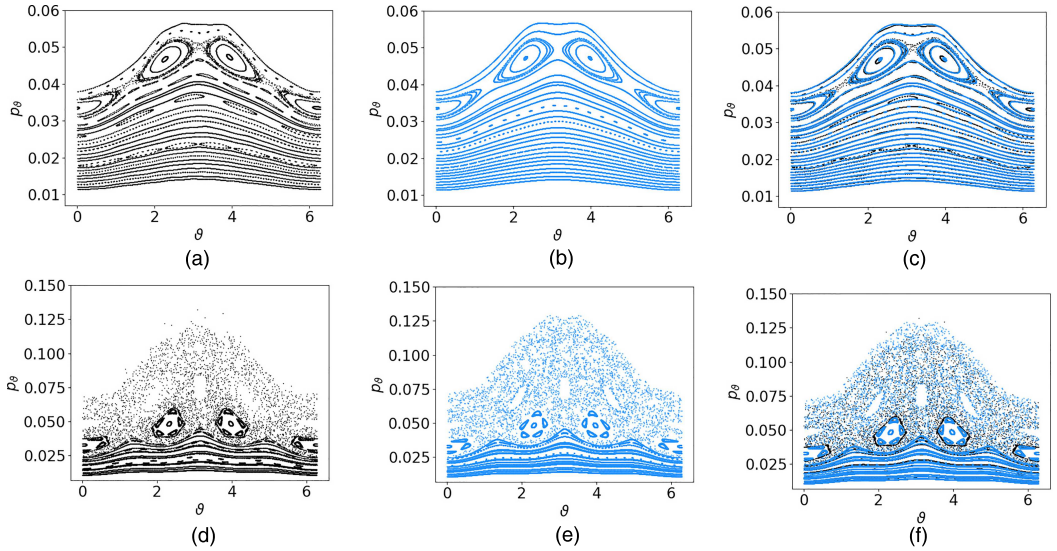


FIG. 18. Poincaré maps of magnetic field lines in a tokamak with non-axisymmetric perturbations in the phase space at $\phi = 2n\pi$ for a perturbation strength of $\varepsilon = 0.001$ (upper) and $\varepsilon = 0.01$ (lower) comparing fourfold split implicit SympGPR trained with $N = 70$ [(a)–(c)] and $N = 110$ [(d)–(f)] initial conditions [(a) and (d)], reference orbits calculated using a symplectic Euler scheme with $h = 0.01$ [(b) and (e)], and direct comparison of both Poincaré maps [(c) and (f)] for 30 test data points and for $n = 1000$ map applications.

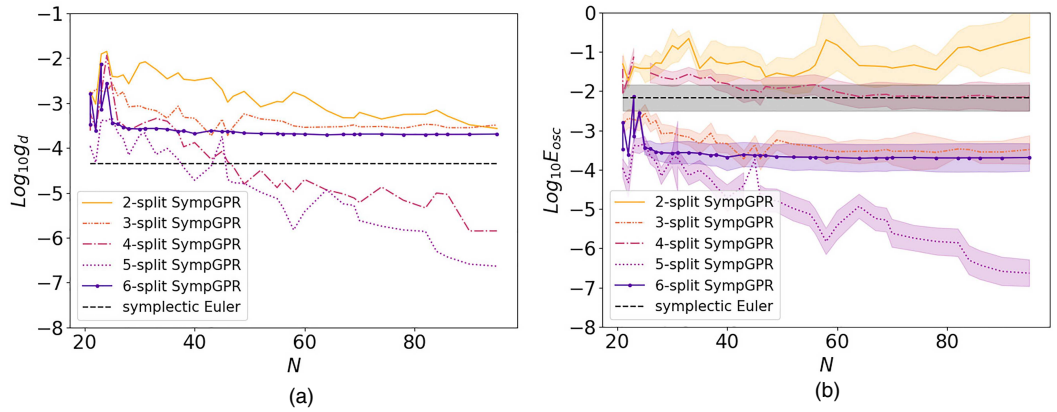


FIG. 19. Poincaré maps of magnetic field lines in a tokamak with non-axisymmetric perturbations in the phase space at $\varphi = 0$ for a perturbation strength of $\varepsilon = 0.001$: Comparison of the geometrical distance [Eq. (40)] (a) and normalized energy oscillations [Eq. (41)] (b) of l-fold split implicit SympGPR and usual symplectic Euler for a variable number of training points N. Colored areas in (b) correspond to the standard deviation for 100 test points. The standard deviation of $\log_{10} g_d$ in (a) is roughly 1 (one order of magnitude) and omitted in the plot for better visibility.

Chaos ARTICLE scitation.org/journal/cha

and run time as for a l-fold split SympGPR, where l denotes the number of sub-steps, l-times more operations have to be carried out for a full turn. Compared to a symplectic Euler integrator on a tensor-product basis with Fourier modes in angles ($n_\varphi = 3, n_\vartheta = 2$) and third order splines in the radius ($n_r = 32$), the fivefold split SympGPR still reduces the CPU time to evaluate $n = 2000$ Poincaré maps of a test orbit at similar accuracy from 22 s to 4 s.

V. DISCUSSION AND COMPARISON OF PERFORMANCE

Here, the performance of SympGPR on the given examples is summarized, and a comparison to alternative methods is drawn. All benchmarks have been run on an Intel Core i7-7500U CPU. In Table I, typical values for the required training time for the application examples are given.

To assess the efficacy of the method proposed in this paper, we provide a comparison with three existing methods: HNN,[21] SympNets,[24] and HénonNet.[18] Those methods rely on artificial neural networks (ANNs) instead of the present GP regression. The HNN approach covers only flow maps. In a similar manner to SympGPR, it also relies on (nonlinear) regression via a generating function F. There, F is, however, immediately identified with H without discussing the connection to canonical transformations. In contrast, SympNet and HénonNet, respectively, mimic time steps of a symplectic Euler scheme and a simplified dynamical system in a deep multilayer ANN. In contrast, the structure of SympGPR is shallow, with only few splittings realized and directly trained with data in each sub-step. HénonNet also puts emphasis on Poincaré maps also treated here. The mentioned methods are explicit in their application, while SympGPR is used in its implicit form. The implicit scheme still reaches competitive performance due to a smaller model size.

For a detailed comparison, the system of a non-perturbed pendulum is chosen, as this model was discussed in all publications. A performance overview is given in Table II, where the number of training data and some estimation of training and run time for the prediction of one orbit with initial conditions $(q, p) = (0.35, 0.5)$ for 20 bounce times are given. Similarly to linear regression, the computational complexity for training a Gaussian process is of order $\mathcal{O}(N^3)$ due to necessary matrix operations. Regression of the flow map in a spectral basis (Figs. 8 and 9) is not considered here due to the lack of an optimized implementation but is expected to reach similar performance. The training data points for training HénonNet are sampled randomly in the domain of $[-\sqrt{2}, \sqrt{2}] \times [-\pi/2, \pi/2]$ with a step size of $h = 0.1$; for SympNets, the 40 training data points are selected on a single trajectory starting from $(q, p) = (1, 0)$ with step size $h = 0.1$, whereas for HNN, a training set consisting of 25 trajectories with each 30 observations was used. For a similar test loss, (implicit) SympGPR needs only $N = 20$ training data points with step size $h = 0.1$ and is also competitive with respect to run time. Considering the interpolation of the pendulum flow map, 20 training data points are sufficient to accurately predict both kinds of motion (inside and outside the separatrix).

TABLE II. Performance comparison of flow map interpolation for an unperturbed pendulum.

	SympGPR	HénonNet[18]	SympNets[24]	HNN[21]
Training data	20	10 000	40	725
Training time	0.8 s	106 s	107 s	50 s
Run time	1.8 s	1.2 s	1.2 s	3 s

A substantial increase in performance for learning the Hamiltonian H is observable compared to existing work,[17] where H is estimated from either 625 (with GPs) or 20 000 (with a neural network) training points and requiring additional gradient information. Using the implicit SympGPR, a similar training and test loss is achieved with only 25 training samples and no gradient data (Fig. 7).

VI. CONCLUSION AND OUTLOOK

In this paper, we have presented a novel approach to represent flow and Poincaré maps of Hamiltonian dynamical systems using Gaussian process regression. A considerable advantage compared to existing methods in a spline or spectral basis is the possibility of using input data of arbitrary geometry with GPs. Conservation of invariants is ensured due to the construction of the SympGPR map. The concept was validated on several Hamiltonian systems for interpolation of flow maps and Poincaré maps. An implicit approach was shown to yield similar accuracy to linear regression in a spectral basis, whereas an explicit mapping requires no iterations in application of the map at the cost of accuracy and stability. Observation of training data within a short period of time allows for an accurate interpolation and even extrapolation of the Hamiltonian function H using substantially less training points compared to existing methods.

We conclude that interpolation of the flow map, while useful to characterize systems without known Hamiltonian, is not very promising to construct emulators that replace direct symplectic integration. While the accuracy is somewhat better than symplectic Euler for the same step size h, there is no substantial advantage in the step size that can be reached in these cases. The method becomes more attractive in cases where either the Hamiltonian is unknown or for the interpolation of Poincaré maps between sections of interest. In this case, the presented method uses considerably less training data than neural networks. Here, a natural limitation is the uniqueness of the generating function. This may be circumvented by

TABLE I. Overview of needed training data and training time.

	Pendulum	Pert. pendulum	Hénon–Heiles	Standard map	Tokamak (fourfold split, $\varepsilon = 0.001$)
Training data	20	50	55	30	70
Training time	0.8 s	4 s	6 s	1 s	40 s

Chaos ARTICLE scitation.org/journal/cha

splitting the GP into several sub-steps, which was successful in the tokamak case, but not for the Hénon–Heiles system, where no intermediate cuts can be identified. If data at intermediate time steps are available, this deep GP can be trained directly at these time steps and hence can keep the regression linear. This could also improve performance of composite neural networks.[18] Vice versa, a possible extension of a split SympGPR to systems where no intermediate cuts or data are available could consist of training a composite deep GP in a nonlinear manner using only the loss at the output as a fit criterion. Another option to handle the non-uniqueness of the generating function is to consider an unwinding transformation to reduce the amount of needed training data and will be investigated in future work.

To increase the accuracy of symplectic mappings as well as the prediction of H, higher order implicit methods analogous to symplectic schemes such as midpoint, Gauss–Legendre, or higher order RK schemes could be investigated in the future. Particularly, the explicit method in combination with a Verlet scheme seems promising to leverage fast computation and the possible higher accuracy. Another interesting direction could be the incorporation of the available variance of predictions from the SympGPR for uncertainty quantification.

ACKNOWLEDGMENTS

The authors would like to thank Johanna Ganglbauer and Bob Warnock for insightful discussions on spline methods for interpolating symplectic maps and David Rügamer for enlightening suggestions on optimization enhancements. The present contribution is supported by the Helmholtz Association of German Research Centers under the joint research school (No. HIDSS-0006) "Munich School for Data Science—MUDS" and the Reduced Complexity Grant No. ZT-I-0010.

APPENDIX A: DERIVATIVE OBSERVATION IN (SYMMETRIC) LINEAR REGRESSION

Collocation and regression via basis functions approximate observed function values $g(\boldsymbol{x}) \in \mathbb{R}$ for $\boldsymbol{x} \in \mathbb{R}^d$ by fitting a linear combination of the chosen basis $\varphi_i(\boldsymbol{x})$,

$$g(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i \varphi_i(\boldsymbol{x}), \tag{A1}$$

where α_i are the weights and n is the number of basis functions. Suitable bases are, e.g., orthogonal polynomials, splines, trigonometric functions (Fourier series), or radial basis functions with kernels $\varphi_i(\boldsymbol{x}) \equiv \varphi(\boldsymbol{x}, \boldsymbol{x}_i)$. In order to obtain a directly invertible positive definite system, one may use a *symmetric* least-squares regression method.[71] Here, we use a product basis, i.e., an univariate basis for each dimension of $\boldsymbol{x}$. Multiplying Eq. (A1) by $\varphi_j(\boldsymbol{x})$ and swapping indexes i and j yields

$$g(\boldsymbol{x})\varphi_i(\boldsymbol{x}) = \sum_{j=1}^{n} \varphi_i(\boldsymbol{x})\varphi_j(\boldsymbol{x})\alpha_j. \tag{A2}$$

Subsequently summing over N observations,

$$\sum_{k=1}^{N} g(\boldsymbol{x}_k)\varphi_i(\boldsymbol{x}_k) = \sum_{k=1}^{N}\sum_{j=1}^{n} \varphi_i(\boldsymbol{x}_k)\varphi_j(\boldsymbol{x}_k)\alpha_j, \tag{A3}$$

we arrive at a linear equation system corresponding to $\boldsymbol{A}\boldsymbol{\alpha} = \boldsymbol{b}$, with

$$A_{ij} = \sum_{k=1}^{N} \varphi_i(\boldsymbol{x}_k)\varphi_j(\boldsymbol{x}_k), \tag{A4}$$

$$b_i = \sum_{k=1}^{N} \varphi_i(\boldsymbol{x}_k) g(\boldsymbol{x}_k). \tag{A5}$$

When derivative observations are considered, the basis changes to $\psi_i = \mathcal{L}\varphi_i$, again resulting in a linear system of the form $\boldsymbol{A}'\boldsymbol{\alpha} = \boldsymbol{b}'$, with

$$A'_{ij} = \sum_{k=1}^{N} \psi_i(\boldsymbol{x}_k) \cdot \psi_j(\boldsymbol{x}_k), \tag{A6}$$

$$b'_i = \sum_{k=1}^{N} \psi_i(\boldsymbol{x}_k) \cdot \mathcal{L} g(\boldsymbol{x}_k). \tag{A7}$$

Here, $\boldsymbol{A}$ and $\boldsymbol{A}'$ are symmetric positive definite matrices that are directly invertible.

APPENDIX B: GUIDING-CENTER AND FIELD LINE LAGRANGIAN

The guiding-center Lagrangian[35,36]

$$L_{\mathrm{gc}}(\boldsymbol{z}) = \frac{e}{c}\sum_{k=1}^{3} A^{\star}_k(\boldsymbol{z})\dot{z}^k + J_{\perp}\dot{\phi} - H(\boldsymbol{z}) \tag{B1}$$

contains covariant components $A^{\star}_k(\boldsymbol{z})$ of the modified magnetic vector potential

$$\boldsymbol{A}^{\star} \equiv \boldsymbol{A} + m v_{\parallel} \frac{c}{e}\frac{\boldsymbol{B}}{B}, \tag{B2}$$

the ignorable pair of gyrophase ϕ and perpendicular invariant $J_{\perp} = mc\mu/e$, and the Hamiltonian

$$H = \frac{m v_{\parallel}^2}{2} + \mu B + e\Phi_e, \tag{B3}$$

where $\boldsymbol{A}$ is the original vector potential, $\boldsymbol{B}$ the magnetic field with modulus B, $v_{\parallel}$ the guiding-center velocity parallel to $\boldsymbol{B}$, Φ_e the electric scalar potential, μ the guiding-center magnetic moment, and e, m, c particle charge, mass, and speed of light, respectively.

We use three spatial variables $z^1 = r$, $z^2 = \vartheta$, $z^3 = \varphi$, and $z^4 = v_{\parallel}$ as phase space coordinates. Spatial coordinates describe nested toroidal surfaces $r = \text{const.}$ parameterized by a toroidal angle φ and a poloidal angle ϑ. By a specific choice of (r, ϑ, φ), one

Chaos ARTICLE scitation.org/journal/cha

component $A_r^\star$ of $\boldsymbol{A}^\star$ vanishes, and L_{gc} appears in a canonical form,

$$L_{gc}(\boldsymbol{z}) = \frac{e}{c}A_\vartheta^\star(\boldsymbol{z})\dot{\vartheta} + \frac{e}{c}A_\varphi^\star(\boldsymbol{z})\dot{\varphi} - H(\boldsymbol{z}). \tag{B4}$$

In Eq. (B4), we have omitted the term associated with the ignorable gyrophase. The two canonical momenta are directly identified as $\frac{e}{c}A_\vartheta^\star$ and $\frac{e}{c}A_\varphi^\star$.

For the subsequent limiting case of magnetic field lines and for the convenient definition of step size and Poincaré sections, we switch from time t to φ as the orbit parameter. This results in the new Lagrangian

$$L_{gc}^\varphi(\boldsymbol{z}) \equiv L_{gc}(\boldsymbol{z})\frac{\mathrm{d}t}{\mathrm{d}\varphi} = \frac{e}{c}A_\vartheta^\star(\boldsymbol{z})\vartheta' - H(\boldsymbol{z})t' + \frac{e}{c}A_\varphi^\star(\boldsymbol{z}), \tag{B5}$$

where $f' \equiv \mathrm{d}f/\mathrm{d}\varphi$. The original canonical toroidal momentum $p_\varphi = \frac{e}{c}A_\varphi^\star$ has now switched roles to become the new Hamiltonian, and

$$p_\vartheta = \frac{e}{c}A_\vartheta^\star(\boldsymbol{z}), \quad p_t = -H(\boldsymbol{z}) \tag{B6}$$

are the transformation equations from non-canonical coordinates $(r, \vartheta, t, v_\parallel)$ to canonical coordinates $(\vartheta, t, p_\vartheta, p_t)$. Physical time t appears now as a cyclic dynamical variable and p_t is an integral of motion, corresponding to conservation of total energy. For vanishing electric potential Φ_e and in the limiting case of strongly passing guiding-centers where μ vanishes, as well as zero kinetic energy with $v_\parallel \to 0$, the definition for p_ϑ reduces to the usual (non-modified) vector potential component A_ϑ. This means that orbits just follow magnetic field lines. We consider this case to study magnetic geometry without reference to physical time t; therefore, we can completely drop the ignorable pair (t, p_t) from L_{gc}^φ, as we did with $(\phi, J_\perp)$ before. Dropping the constant factor e/c, we obtain the canonical variant of the magnetic field line Lagrangian

$$L_{\boldsymbol{B}} = A_\vartheta(r, \vartheta, \varphi)\vartheta' + A_\varphi(r, \vartheta, \varphi) \tag{B7}$$

with position ϑ, momentum $p_\vartheta = A_\vartheta$, orbit parameter φ, and Hamiltonian A_φ. The Lagrangian (B7) is also directly obtained by treating the magnetic field arising from the vector potential $\boldsymbol{A}$ with vanishing A_r as a Hamiltonian system.[34,67,68]

APPENDIX C: NUMERICAL BENCHMARK: FLOW MAP OF THE HÉNON–HEILES SYSTEM

Similarly to the numerical benchmark of the unperturbed pendulum, we have performed an analysis for the interpolation of the flow map for the Hénon–Heiles system. As stated in Sec. IV C 2, a squared exponential kernel was used with l set to its optimized maximum likelihood value and $\sigma_n^2 = 10^{-16}$ (implicit SympGPR) and 10^{-10} (explicit SympGPR). As for the pendulum, σ_f^2 is set in accordance with the observations to $2\max(|Y|)^2$, where Y corresponds to the change in coordinates. In the four dimensional phase space, the training area is restricted to $q_1, q_2, p_1, p_2 \in [-0.5, 0.5]$, where initial conditions are sampled from a Halton sequence[51] and integrated until $t = h$ using a RK45 integrator with an adaptive step size and a relative tolerance of 10^{-13}. Consequently, we consider here the flow map of initial conditions resulting in energies in a range of $[0.02, 0.29]$. We used 100 randomly sampled test data points in the range $q_1, q_2, p_1, p_2 \in [-0.2, 0.2]$ and evaluated the performance of $n = 300$ subsequent mapping applications. The results shown in Figs. 21 and 20 are similar to the findings for the pendulum (Figs. 8 and 9): with increasing step size h, the geometrical distance also increases (Fig. 20). More severely, explicit SympGPR loses long-term stability at increasing step size h in the Hénon–Heiles system due to certain orbits that escape the trapped state after a few 10-100

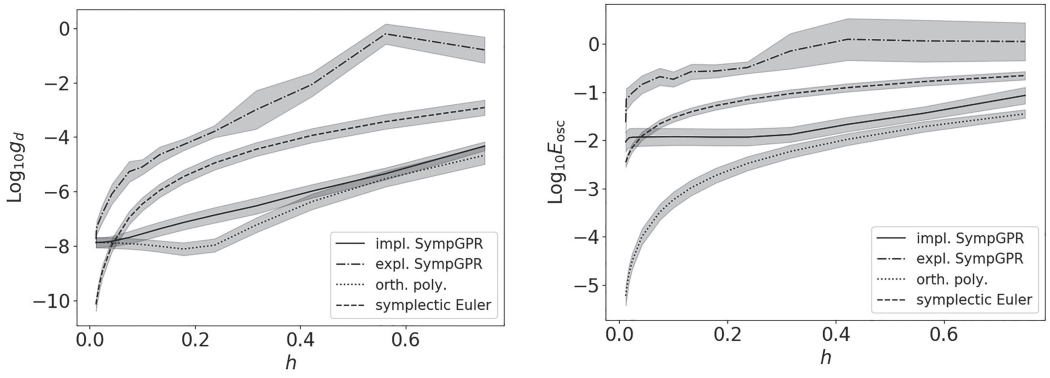


FIG. 20. Hénon–Heiles system: Comparison of geometrical distance [Eq. (40)] (a) and normalized energy oscillations [Eq. (41)] (b) of implicit and explicit SympGPR, standard symplectic Euler and Hermite basis functions for a fixed number of training points $N = 20$ at a variable step size h. The gray areas surrounding the mean correspond to the standard deviation for 100 test points.

Chaos ARTICLE scitation.org/journal/cha

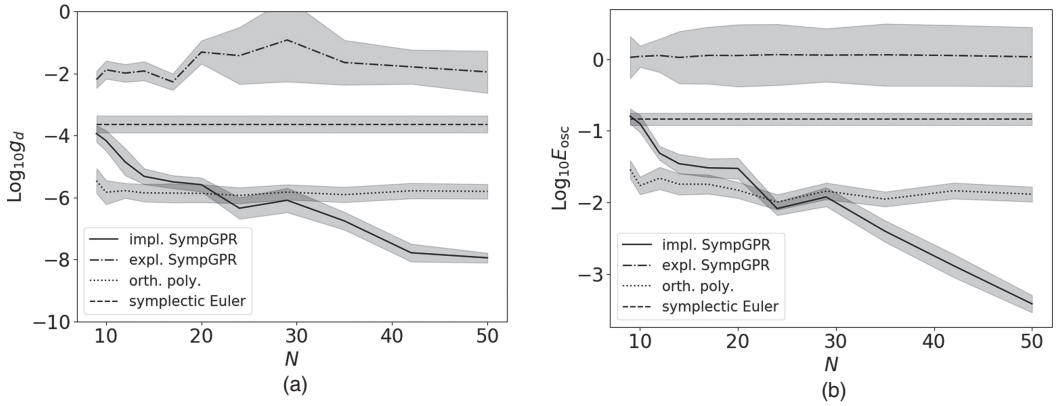


FIG. 21. Hénon–Heiles system: Comparison of the geometrical distance [Eq. (40)] (a) and normalized energy oscillations [Eq. (41)] (b) of implicit and explicit SympGPR, standard symplectic Euler and Hermite basis functions for a variable number of training points N, and a fixed step size $h = 0.5$. Gray areas correspond to the standard deviation for 100 test points.

applications of the map. This severely limits the applicability range of explicit SympGPR in its current state. The quality measures for fixed step size h and increasing N show better accuracy of implicit SympGPR compared to the symplectic Euler and for $N > 30$ also compared to spectral linear regression, reaching the square root of machine precision at $N \approx 40$ with $h = 0.5$.

DATA AVAILABILITY

The data and source code that support the findings of this study are openly available[65] and maintained on https://github.com/redmod-team/SympGPR. The numerical benchmark was performed using the proFit[66] toolkit, which is maintained on https://github.com/redmod-team/profit.

REFERENCES

[1] H. Goldstein, *Classical Mechanics*, 2nd ed. (Addison-Wesley, 1980).
[2] V. I. Arnold, *Mathematical Methods of Classical Mechanics*, Graduate Texts in Mathematics Vol. 60 (Springer, New York, 1989).
[3] J. E. Marsden and T. S. Ratiu, *Introduction to Mechanics and Symmetry* (Springer, New York, 1999).
[4] R. M. Neil, "MCMC using Hamiltonian dynamics," in *Handbook of Markov Chain Monte Carlo*, edited by S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng (Chapman & Hall/CRC, 2011), pp. 113–162.
[5] A. Lichtenberg and M. Lieberman, *Regular and Chaotic Dynamics*, Applied Mathematical Sciences (Springer, 1992).
[6] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations* (Springer, 2006).
[7] F. Casas and S. Blanes, *A Concise Introduction to Geometric Numerical Integration*, 1st ed. (Chapman and Hall/CRC, 2016).
[8] R. I. McLachlan and G. R. W. Quispel, "Geometric integrators for ODEs," J. Phys. A: Math. Gen. **39**, 5251–5285 (2006).
[9] R. I. McLachlan and G. R. W. Quispel, "Splitting methods," Acta Numer. **11**, 341–434 (2002).
[10] G. R. W. Quispel, and D. I. McLaren, "A new class of energy-preserving numerical integration methods," J. Phys. A: Math. Theor. **41**, 045206 (2008).
[11] C. Danieli, B. M. Manda, M. Thudiyangal, and C. Skokos, "Computational efficiency of numerical integration methods for the tangent dynamics of many-body Hamiltonian systems in one and two spatial dimensions," arXiv:1812.01870 (2019).
[12] S. S. Abdullaev, *Construction of Mappings for Hamiltonian Systems and Their Applications* (Springer, 2006).
[13] J. S. Berg, R. L. Warnock, R. D. Ruth, and É. Forest, "Construction of symplectic maps for nonlinear motion of particles in accelerators," Phys. Rev. E **49**, 722–739 (1994).
[14] S. V. Kasilov, V. E. Moiseenko, and M. F. Heyn, "Solution of the drift kinetic equation in the regime of weak collisions by stochastic mapping techniques," Phys. Plasmas **4**, 2422 (1997).
[15] S. V. Kasilov, W. Kernbichler, V. V. Nemov, and M. F. Heyn, "Mapping technique for stellarators," Phys. Plasmas **9**, 3508 (2002).
[16] R. Warnock, Y. Cai, and J. A. Ellison, "Construction of large period symplectic maps by interpolative methods," Report No. SLAC-PUB-13867, SLAC National Accelerator Laboratory, 2009.
[17] T. Bertalan, F. Dietrich, I. Mezić, and I. G. Kevrekidis, "On learning Hamiltonian systems from data," Chaos **29**, 121107 (2019).
[18] J. W. Burby, Q. Tang, and R. Maulik, "Fast neural Poincaré maps for toroidal magnetic fields," arXiv:2007.04496 (2020).
[19] G. Deco and W. Brauer, "Nonlinear higher-order statistical decorrelation by volume-conserving neural architectures," Neural Netw. **8**, 525–535 (1995).
[20] L. C. Parra, "Symplectic nonlinear component analysis," in *Proceedings of the 8th International Conference on Neural Information Processing Systems: NIPS'95* (MIT Press, Cambridge, MA, 1995), pp. 437–443.
[21] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," arXiv:1906.01563 (2019).
[22] Z. Chen, J. Zhang, M. Arjovsky, and L. Bottou, "Symplectic recurrent neural networks," arXiv:1909.13334 (2019).
[23] P. Toth, D. J. Rezende, A. Jaegle, S. Racanière, A. Botev, and I. Higgins, "Hamiltonian generative networks," arXiv:1909.13789 (2019).
[24] P. Jin, Z. Zhang, A. Zhu, Y. Tang, and G. E. Karniadakis, "Sympnets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems," arXiv:2001.03750 (2020).

[25]L. C. Parra, "Symplectic nonlinear component analysis," in *Proceedings of the 8th International Conference on Neural Information Processing Systems, NIPS'95* (MIT Press, Cambridge, MA, USA, 1995), pp. 437–443.
[26]S.-H. Li, C.-X. Dong, L. Zhang, and L. Wang, "Neural canonical transformation with symplectic flows," Phys. Rev. X **10**, 021020 (2020).
[27]B. Hamzi and H. Owhadi, "Learning dynamical systems from data: A simple cross-validation perspective," Technical Report arXiv:2007.05074v1 (2020).
[28]M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Inferring solutions of differential equations using noisy multi-fidelity data," J. Comput. Phys. **335**, 736–746 (2017).
[29]C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)* (The MIT Press, 2005).
[30]E. Solak, R. Murray-Smith, W. E. Leithead, D. J. Leith, and C. E. Rasmussen, "Derivative observations in Gaussian process models of dynamic systems," in *Advances in Neural Information Processing Systems 15*, edited by S. Becker, S. Thrun, and K. Obermayer (MIT Press, 2003), pp. 1057–1064.
[31]D. Eriksson, E. Lee, K. Dong, D. Bindel, and A. Wilson, "Scaling Gaussian process regression with derivatives," Adv. Neural Inf. Process. Syst. **2018**, 6867–6877.
[32]A. O'Hagan, "Some Bayesian numerical analysis," Bayesian Stat. **4**, 345–363 (1992).
[33]M. A. Álvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," Found. Trends Mach. Learn. **4**, 195–266 (2012).
[34]A. H. Boozer, "Physics of magnetically confined plasmas," Rev. Mod. Phys. **76**, 1071–1141 (2005).
[35]R. G. Littlejohn, "Variational principles of guiding centre motion," J. Plasma Phys. **29**, 111–125 (1983).
[36]J. R. Cary and A. J. Brizard, "Hamiltonian theory of guiding-center motion," Rev. Mod. Phys. **81**, 693–738 (2009).
[37]R. White, *The Theory of Toroidally Confined Plasmas* (Imperial College Press, 2006).
[38]J. R. Cary and R. G. Littlejohn, "Noncanonical Hamiltonian mechanics and its application to magnetic field line flow," Ann. Phys. **151**, 1–34 (1983).
[39]A. H. Boozer, "Time-dependent drift Hamiltonian," Phys. Fluids **27**, 2441–2445 (1984).
[40]M. Li, B. N. Breizman, and L. Zheng, "Canonical straight field line magnetic flux coordinates for tokamaks," J. Comput. Phys. **326**, 334–341 (2016).
[41]C. Albert, S. Kasilov, and W. Kernbichler, "Symplectic integration with non-canonical quadrature for guiding-center orbits in magnetic confinement devices," J. Comput. Phys. **403**, 109065 (2020).
[42]C. G. Albert and K. Rath, "Gaussian process regression for data fulfilling linear differential equations with localized sources," Entropy **22**, 152 (2020).
[43]G. E. Fasshauer, "Solving partial differential equations by collocation with radial basis functions," in *Surface Fitting and Multiresolution Methods*, edited by A. L. Méhauté, C. Rabut, and L. Schumaker (Vanderbilt University Press, 1997), Vol. 2, pp. 131–138.
[44]H. Owhadi, "Bayesian numerical homogenization," Multiscale Model. Simul. **13**, 812–828 (2015).
[45]J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," Neural Comput. **3**, 246–257 (1991).
[46]C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," J. Mach. Learn. Res. **7**, 2651–2667 (2006).
[47]A. Caponnetto, C. A. Micchelli, M. Pontil, and Y. Ying, "Universal multi-task kernels," J. Mach. Learn. Res. **9**, 1615–1646 (2008).
[48]C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," ACM Trans. Math. Softw. **23**, 550–560 (1997).
[49]N. Hansen, "The CMA evolution strategy: A tutorial," arXiv:1604.00772 (2016).
[50]N. Hansen, Y. Akimoto, and P. Baudis, CMA-ES/pycma on GitHub, Zenodo (2019).
[51]H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992).
[52]J. Dormand and P. Prince, "A family of embedded Runge-Kutta formulae," J. Comput. Appl. Math. **6**, 19–26 (1980).
[53]M. Hénon and C. Heiles, "The applicability of the third integral of motion: Some numerical experiments," Astron. J. **69**, 73–79 (1964).
[54]E. E. Zotos, "An overview of the escape dynamics in the Hénon-Heiles Hamiltonian system," Meccanica **52**, 2615–2630 (2017).
[55]B. V. Chirikov, "A universal instability of many-dimensional oscillator systems," Phys. Rep. **52**, 263–379 (1979).
[56]A. B. Rechester and R. B. White, "Calculation of turbulent diffusion for the Chirikov-Taylor model," Phys. Rev. Lett. **44**, 1586–1589 (1980).
[57]Y. H. Ichikawa, T. Kamimura, and T. Hatori, "Stochastic diffusion in the standard map," Physica D **29**, 247–255 (1987).
[58]I. Dana and S. Fishman, "Diffusion in the standard map," Physica D **17**, 63–74 (1985).
[59]R. Venegeroles, "Calculation of superdiffusion for the Chirikov-Taylor model," Phys. Rev. Lett. **101**, 054102 (2008).
[60]G. M. Zaslavsky, M. Edelman, and B. A. Niyazov, "Self-similarity, renormalization, and phase space nonuniformity of Hamiltonian chaotic dynamics," Chaos **7**, 159–181 (1997).
[61]M. Harsoula and G. Contopoulos, "Global and local diffusion in the standard map," Phys. Rev. E **97**, 022215 (2018).
[62]T. Manos and M. Robnik, "Dynamical localization in chaotic systems: Spectral statistics and localization measure in the kicked rotator as a paradigm for time-dependent and time-independent systems," Phys. Rev. E **87**, 062905 (2013).
[63]T. Manos and M. Robnik, "Survey on the role of accelerator modes for anomalous diffusion: The case of the standard map," Phys. Rev. E **89**, 022905 (2014).
[64]M. Eder, C. G. Albert, L. M. P. Bauer, S. V. Kasilov, and W. Kernbichler, "Quasi-geometric integration of guiding-center orbits in piecewise linear toroidal fields," Phys. Plasmas **27**, 122508 (2020).
[65]K. Rath, C. Albert, B. Bischl, and U. von Toussaint (2021). "SympGPR v1.0: Symplectic Gaussian process regression," Zenodo. 10.5281/zenodo.4549092
[66]C. Albert, R. Hofmeister, and K. Rath (2020). "proFit v0.3-alpha: Probabilistic response model fitting with interactive tools," Zenodo. 10.5281/zenodo.3580488
[67]J. W. Burby and C. L. Ellison, "Toroidal regularization of the guiding center Lagrangian," Phys. Plasmas **24**, 110703 (2017).
[68]P. J. Morrison, "Structure and structure-preserving algorithms for plasma physics," Phys. Plasmas **24**, 055502 (2017).
[69]R. Berndt, M. Klucznik, and A. M. Society, *An Introduction to Symplectic Geometry*, Contemporary Mathematics (American Mathematical Society, 2001).
[70]J. V. José and E. J. Saletan, *Classical Dynamics: A Contemporary Approach* (Cambridge University Press, 1998).
[71]G. Seber and A. Lee, *Linear Regression Analysis*, Wiley Series in Probability and Statistics (Wiley, 2012).

3.3 Orbit Classification in Dynamical Systems Using Surrogate Models

Main novelty:
Based on Symplectic Gaussian process regression (Rath et al., 2021b), we use the Jacobian available from the surrogate model for early classification of regular and chaotic trajectories.

Contributing article:
Rath, K., Albert, C. G., Bischl, B., and von Toussaint, U. (2021a). Orbit Classification and Sensitivity Analysis in Dynamical Systems Using Surrogate Models. *Physical Sciences Forum*, 3(1)

Author contributions:
Katharina Rath devised the conceptual idea of the project under supervision by and with support from Christopher Albert, Bernd Bischl and Udo von Toussaint. Katharina Rath implemented the software, performed the numerical experiments and wrote the paper. Christopher Albert, Bernd Bischl and Udo von Toussaint advised throughout the whole project.

physical sciences forum

MDPI

Proceeding Paper

Orbit Classification and Sensitivity Analysis in Dynamical Systems Using Surrogate Models †

Katharina Rath [1,2,*], Christopher G. Albert [2], Bernd Bischl [1] and Udo von Toussaint [2]

[1] Department of Statistics, Ludwig-Maximilians-Universität München, 80333 Munich, Germany; bernd.bischl@stat.uni-muenchen.de
[2] Max-Planck-Institut für Plasmaphysik, 85748 Garching, Germany; albert@alumni.tugraz.at (C.G.A.); udo.v.toussaint@ipp.mpg.de (U.v.T.)
* Correspondence: katharina.rath@ipp.mpg.de
† Presented at the 40th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, online, 4–9 July 2021.

Abstract: Dynamics of many classical physics systems are described in terms of Hamilton's equations. Commonly, initial conditions are only imperfectly known. The associated volume in phase space is preserved over time due to the symplecticity of the Hamiltonian flow. Here we study the propagation of uncertain initial conditions through dynamical systems using symplectic surrogate models of Hamiltonian flow maps. This allows fast sensitivity analysis with respect to the distribution of initial conditions and an estimation of local Lyapunov exponents (LLE) that give insight into local predictability of a dynamical system. In Hamiltonian systems, LLEs permit a distinction between regular and chaotic orbits. Combined with Bayesian methods we provide a statistical analysis of local stability and sensitivity in phase space for Hamiltonian systems. The intended application is the early classification of regular and chaotic orbits of fusion alpha particles in stellarator reactors. The degree of stochastization during a given time period is used as an estimate for the probability that orbits of a specific region in phase space are lost at the plasma boundary. Thus, the approach offers a promising way to accelerate the computation of fusion alpha particle losses.

Keywords: Gaussian process regression; surrogate model; Lyapunov exponent; sensitivity analysis; Hamiltonian systems

Citation: Rath, K.; Albert, C.G.; Bischl, B.; von Toussaint, U. Orbit Classification and Sensitivity Analysis in Dynamical Systems Using Surrogate Models. *Phys. Sci. Forum* **2021**, *3*, 5. https://doi.org/10.3390/psf2021003005

Academic Editors: Wolfgang von der Linden and Sascha Ranftl

Published: 5 November 2021

1. Introduction

Hamilton's equations describe the dynamics of many classical physics systems such as classical mechanics, plasma physics or electrodynamics. In most of these cases, chaos plays an important role [1]. One fundamental question in analyzing these chaotic Hamiltonian systems is the distinction between regular and chaotic regions in phase space. A commonly used tool are Poincaré maps, which connect subsequent intersections of orbits with a lower-dimensional subspace, called Poincaré section. For example, in a planetary system one could record a section each time the planet has made a turn around the Sun. The resulting pattern of intersection points on this subspace allow insight into the dynamics of the underlying system: regular orbits stay bound to a closed hyper-surface and do not leave the confinement volume, whereas chaotic orbits might spread over the whole phase space. This is related to the breaking of KAM (Kolmogorov-Arnold-Moser) surfaces that form barriers for motion in phase space [2]. The classification of regular versus chaotic orbits is performed, e.g., via box-counting [3] or by calculating the spectrum of Lyapunov exponents [4–6]. Lyapunov exponents measure the asymptotic average exponential rate of divergence of nearby orbits in phase space over infinite time and are therefore invariants of the dynamical system. When considering only finite time, the obtained *local* Lyapunov exponents (LLEs) for a specific starting position depend on the position in phase space and give insight into the local predictability of the dynamical system of interest [7–10].

Poincaré maps are in most cases inefficient to compute as their computation involves numerical integration of Hamilton's equations even though only intersections with the surface of interest are recorded. When using a surrogate model to interpolate the Poincaré map, the symplectic structure of phase space arising from the description in terms of the Hamiltonian description has to be preserved to obtain long-term stability and conservation of invariants of motion, e.g., volume preservation. Additional information on Hamiltonian systems and symplecticity can be found in [2,11]. Here, we use a structure-preserving Gaussian process surrogate model (SympGPR) that interpolates directly between Poincaré sections and thus avoids unnecessary computation while achieving similar accuracy as standard numerical integration schemes [12].

In the present work, we investigate how the symplectic surrogate model [12] can be used for early classification of chaotic versus regular trajectories based on the calculation of LLEs. The latter are calculated using the Jacobian that is directly available from the surrogate model [13]. As LLEs also depend on time, we study their distribution on various time scales to estimate the needed number of mapping iterations. We combine the orbit classification with a sensitivity analysis based on variance decomposition [14–16] to evaluate the influence of uncertain initial conditions in different regions of phase space. The analysis is carried out on the well-known standard map [17] that is well suited for validation purposes as a closed form expression for the Poincaré maps is available. This, however, does not influence the performance of the surrogate model that is applicable also in cases where such a closed form doesn't exist [12].

The intended application is the early classification of regular and chaotic orbits of fusion alpha particles in stellarator reactors [3]. While regular particles can be expected to remain confined indefinitely, only chaotic orbits have to be traced to the end. This offers a promising way to accelerate loss computations for stellarator optimization.

2. Methods

2.1. Hamiltonian Systems

A $f-$dimensional system (with $2f-$dimensional phase space) described by its Hamiltonian $H(\boldsymbol{q}, \boldsymbol{p}, t)$ depending on f generalized coordinates $\boldsymbol{q}$ and f generalized momenta $\boldsymbol{p}$ satisfies Hamilton's canonical equations of motion,

$$\dot{\boldsymbol{q}}(t) = \frac{d\boldsymbol{q}(t)}{dt} = \nabla_{\boldsymbol{p}} H(\boldsymbol{q}(t), \boldsymbol{p}(t)), \qquad \dot{\boldsymbol{p}}(t) = \frac{d\boldsymbol{p}(t)}{dt} = -\nabla_{\boldsymbol{q}} H(\boldsymbol{q}(t), \boldsymbol{p}(t)), \tag{1}$$

which represent the time evolution as integral curves of the Hamiltonian vector field.

Here, we consider the standard map [17] that is a well-studied model to investigate chaos in Hamiltonian systems. Each mapping step corresponds to one Poincaré map of a periodically kicked rotator:

$$p_{n+1} = (p_n + K\sin(q_n)) \bmod 2\pi, \qquad q_{n+1} = (q_n + p_{n+1}) \bmod 2\pi, \tag{2}$$

where K is the stochasticity parameter corresponding to the intensity of the perturbation. The standard map is an area-preserving map with $\det J = 1$, where J is its Jacobian:

$$J = \begin{pmatrix} \frac{\partial q_{n+1}}{\partial q_n} & \frac{\partial q_{n+1}}{\partial p_n} \\ \frac{\partial p_{n+1}}{\partial q_n} & \frac{\partial p_{n+1}}{\partial p_n} \end{pmatrix} = \begin{pmatrix} 1 + K\cos(q_n) & 1 \\ K\cos(q_n) & 1 \end{pmatrix} \tag{3}$$

2.2. Symplectic Gaussian Process Emulation

A Gaussian process (GP) [18] is a collection of random variables, any finite number of which have a joint Gaussian distribution. A GP is fully specified by its mean $\boldsymbol{m}(\boldsymbol{x})$ and kernel or covariance function $K(\boldsymbol{x}, \boldsymbol{x}')$ and is denoted as

$$\boldsymbol{f}(\boldsymbol{x}) \sim \mathcal{GP}(\boldsymbol{m}(\boldsymbol{x}), K(\boldsymbol{x}, \boldsymbol{x}')), \tag{4}$$

for input data points $\boldsymbol{x} \in \mathbb{R}^d$. Here, we allow vector-valued functions $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^D$ [19]. The covariance function is a positive semidefinite matrix-valued function, whose entries $(K(\boldsymbol{x}, \boldsymbol{x}'))_{ij}$ express the covariance between the output dimensions i and j of $\boldsymbol{f}(\boldsymbol{x})$.

For regression, we rely on observed function values $Y \in \mathbb{R}^{D\times N}$ with entries $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) + \epsilon$. These observations may contain local Gaussian noise ϵ, i.e., the noise is independent at different positions $\boldsymbol{x}$ but may be correlated between components $\boldsymbol{y}$. The input variables are aggregated in the $d \times N$ design matrix X, where N is the number of training data points. The posterior distribution, after taking training data points into account, is still a GP with updated mean $F_* \equiv \mathbb{E}(F(X_*))$ and covariance function allowing to make predictions for test data X_*:

$$F_* = K(X_*, X)(K(X, X) + \Sigma_n)^{-1} Y, \tag{5}$$

$$\mathrm{cov}(F_*) = K(X_*, X_*) - K(X_*, X)(K(X, X) + \Sigma_n)^{-1} K(X, X_*), \tag{6}$$

where $\Sigma_n \in \mathbb{R}^{ND\times ND}$ is the covariance matrix of the multivariate output noise for each training data point. Here we use the shorthand notation $K(X, X)$ for the block matrix assembled over the output dimension D in addition to the number of input points as in a single-output GP with a scalar covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ that expresses the covariance of different input data points $\boldsymbol{x}$ and $\boldsymbol{x}'$. The kernel parameters are estimated given the input data by minimizing the negative log-likelihood [18].

To construct a GP emulator that interpolates symplectic maps for Hamiltonian systems, symplectic Gaussian process regression (SympGPR) was presented in [12] where the generating function $F(q, P)$ and its gradients are interpolated using a multi-output GP with derivative observations [20,21]. The generating function links old coordinates $(q, p) = (q_n, p_n)$ to new coordinates $(Q, P) = (q_{n+1}, p_{n+1})$ (e.g., after one iteration of the standard map Equation (2)) via a canonical transformation such that the symplectic property of phase space is preserved. Thus, input data points consist of pairs (q, P). Then, the covariance matrix contains the Hessian of an original scalar covariance function $k(q, P, q', P')$ as the lower block matrix $L(q, P, q', P')$ (denoted with the red box):

$$K(q, P, q', P') = \begin{pmatrix} k & \partial_{q'}k & \partial_{P'}k \\ \partial_q k & \partial_{qq'}k & \partial_{qP'}k \\ \partial_P k & \partial_{Pq'}k & \partial_{PP'}k \end{pmatrix}. \tag{7}$$

Using the algorithm for the (semi-)implicit symplectic GP map as presented in [12], once the SympGPR model is trained and the covariance matrix calculated, the model is used to predict subsequent time steps or Poincaré maps for arbitrary initial conditions.

For the estimation of the Jacobian (Equation (3)) from the SympGPR, the Hessian of the generating function $F(q, P)$ has to be inferred from the training data. Thus, the covariance matrix is extended with a block matrix C containing third derivatives of $k(q, P, q', P')$:

$$C = \begin{pmatrix} \partial_{q,q',q}k & \partial_{q,P',q}k & \partial_{P,q',q}k & \partial_{P,P',q}k \\ \partial_{q,q',P}k & \partial_{q,P',P}k & \partial_{P,q',P}k & \partial_{P,P',P}k \end{pmatrix}. \tag{8}$$

The mean of the posterior distribution of the desired Hessian of the generating function $F(q, P)$ is inferred via

$$\nabla^2 F = (\partial^2_{qq}F, \partial^2_{qP}F, \partial^2_{Pq}F, \partial^2_{PP}F)^\top = CL^{-1}Y. \tag{9}$$

As we have a dependence on mixed coordinates $Q(\bar{q}(q, p), P(q, p))$ and $P(Q(q, p), \bar{p}(q, p))$, where we used $\bar{q}(q, p) = q$ and $\bar{p}(q, p) = p$ to correctly carry out the inner derivatives, the needed elements for the Jacobian can be calculated employing the chain rule. The Jacobian is then given as the solution of the well-determined linear set of equations:

$$\frac{\partial Q}{\partial q} = \frac{\partial Q}{\partial \bar{q}}\frac{\partial \bar{q}}{\partial q} + \frac{\partial Q}{\partial P}\frac{\partial P}{\partial q}, \qquad \frac{\partial Q}{\partial p} = \frac{\partial Q}{\partial \bar{q}}\frac{\partial \bar{q}}{\partial p} + \frac{\partial Q}{\partial P}\frac{\partial P}{\partial p}, \tag{10}$$

$$\frac{\partial P}{\partial q} = \frac{\partial P}{\partial Q}\frac{\partial Q}{\partial q} + \frac{\partial P}{\partial \bar{p}}\frac{\partial \bar{p}}{\partial q}, \qquad \frac{\partial P}{\partial p} = \frac{\partial P}{\partial Q}\frac{\partial Q}{\partial p} + \frac{\partial P}{\partial \bar{p}}\frac{\partial \bar{p}}{\partial p}, \tag{11}$$

where we use the following correspondence to determine all factors of the SOEs:

$$\begin{pmatrix} \frac{\partial Q}{\partial \bar{q}} & \frac{\partial Q}{\partial P} \\ \frac{\partial \bar{p}}{\partial \bar{q}} & \frac{\partial \bar{p}}{\partial P} \end{pmatrix} = \begin{pmatrix} \frac{\partial \bar{q}}{\partial Q} & \frac{\partial P}{\partial Q} \\ \frac{\partial \bar{q}}{\partial \bar{p}} & \frac{\partial P}{\partial \bar{p}} \end{pmatrix}^{\top} = \begin{pmatrix} 1 + \frac{\partial^2 F}{\partial q \partial P} & -\frac{\partial^2 F}{\partial P \partial P} \\ -\frac{\partial^2 F}{\partial q \partial q} & 1 + \frac{\partial^2 F}{\partial P \partial q} \end{pmatrix}. \tag{12}$$

2.3. Sensitivity Analysis

Variance-based sensitivity analysis decomposes the variance of the model output into portions associated with uncertainty in the model inputs or initial conditions [14,15]. Assuming independent input variables X_i, $i = 1, ..., d$, the functional analysis of variance (ANOVA) allows a decomposition of the scalar model output Y from which the decomposition of the variance can be deduced:

$$V[Y] = \sum_{i=1}^{d} V_i + \sum_{1 \le i < j \le d} V_{ij} + ... + V_{1,2,...,d} \tag{13}$$

The first term describes the variation in variance only due to changes in single variables X_i, whereas higher-order interactions are depicted in the contributions of the interaction terms. From this, first-order Sobol' indices S_i are defined as the corresponding fraction of the total variance, whereas *total* Sobol' indices S_{T_i} also take the influence of X_i interacting with other input variables into account [14,15]:

$$S_i = \frac{V_i}{\mathrm{Var}(Y)}, \qquad S_{T_i} = \frac{E_{\boldsymbol{X}_{\sim i}}(\mathrm{Var}_{X_i}(Y|\boldsymbol{X}_{\sim i})}{\mathrm{Var}(Y)} \tag{14}$$

Several methods for efficiently calculating Sobol' indices have been presented, e.g., MC sampling [14,16] or direct estimation from surrogate models [22,23]. Here, we use the MC sampling strategy presented in [16] using two sampling matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and a combination of both $\boldsymbol{A}_{\boldsymbol{B}}^{(i)}$, where all columns are from $\boldsymbol{A}$ except the i-th column which is from $\boldsymbol{B}$:

$$S_i \mathrm{Var}(Y) = \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{B})_j (f(\boldsymbol{A}_{\boldsymbol{B}}^{(i)})_j - f(\boldsymbol{A})_j), \qquad S_{T_i}\mathrm{Var}(Y) = \frac{1}{2N}\sum_{i=1}^{N} (f(\boldsymbol{A})_j - f(\boldsymbol{A}_{\boldsymbol{B}}^{(i)})_j)^2, \tag{15}$$

where f denotes the model to be evaluated.

2.4. Local Lyapunov Exponents

For a dynamical system in $\mathbb{R}^D$, D Lyapunov characteristic exponents λ_n give the exponential separation of trajectories with initial conditions $\boldsymbol{z}(0) = (q(0), p(0))$ of a dynamical system with perturbation $\delta \boldsymbol{z}$ over time:

$$|\delta \boldsymbol{z}(T)| = \mathcal{J}_{\boldsymbol{z}(T)}^{(T)} \delta \boldsymbol{z}(0) \approx e^{T\lambda} |\delta \boldsymbol{z}(0)|, \tag{16}$$

where $\mathcal{J}_{\boldsymbol{z}(T)}^{(T)}$ is a time-ordered product of Jacobians $\mathcal{J}_{\boldsymbol{z}(T-1)}\mathcal{J}_{\boldsymbol{z}(T-2)}...\mathcal{J}_{\boldsymbol{z}(1)}\mathcal{J}_{\boldsymbol{z}(0)}$ [4]. The Lyapunov exponents are then given as the logarithm of the eigenvalues of the positive and symmetric matrix.

$$\Lambda = \lim_{T\to\infty} [\mathcal{J}_{\boldsymbol{z}(T)}^{(T)\top} \mathcal{J}_{\boldsymbol{z}(T)}^{(T)}]^{1/(2T)}, \tag{17}$$

where $\top$ denotes the transpose of $\mathcal{J}_{\boldsymbol{z}(T)}^{(T)}$.

For a D-dimensional system, there exist D Lyapunov exponents λ_n giving the rate of growth of a D-volume element with $\lambda_1 + ... + \lambda_D$ corresponding to the rate of growth

of the determinant of the Jacobian $\det(\mathcal{J}^{(T)}_{\boldsymbol{z}(T)})$. From this follows that for a Hamiltonian system with a symplectic (e.g., volume-preserving) phase space structure, Lyapunov exponents exist in additive inverse pairs as the determinant of the Jacobian is constant, $\lambda_1 + ... + \lambda_D = 0$. In the dynamical system of the standard map with $D = 2$ considered here, the Lyapunov exponents allow a distinction between regular and chaotic motion. If the Lyapunov exponents $\lambda_1 = -\lambda_2 > 0$, neighboring orbits separate exponentially which corresponds to a chaotic region. In contrast, when $\lambda_1 = -\lambda_2 \approx 0$ the motion is regular [1].

As the product of Jacobians is ill-conditioned for large values of T, several algorithms have been proposed to calculate the spectrum of Lyapunov exponents [13]. Here, we determine *local* Lyapunov exponents (LLE) that determine the predictability of an orbit of the system at a specific phase point for finite time. In contrast to *global* Lyapunov exponents they depend on T and on the position in phase space $\boldsymbol{z}$. We use recurrent Gram-Schmidt orthonormalization procedure through QR decomposition [5,6,24], where we follow the evolution of D initially orthonormal deviation vectors $\boldsymbol{w}^n_0$. The Jacobian is decomposed into $\mathcal{J}_{\boldsymbol{z}(0)} = Q^{(1)}R^{(1)}$, where $Q^{(1)}$ is an orthogonal matrix and $R^{(1)}$ is an upper triangular matrix yielding a new set of orthonormal vectors $\boldsymbol{w}_i$. At the next mapping iteration, the matrix product $\mathcal{J}_{\boldsymbol{z}(1)}Q^{(1)}$ is again decomposed. This procedure is repeated T times to arrive at $\mathcal{J}^{(T)}_{z(t)} = Q^{(T)}R^{(T)}R^{(T-1)}...R^{(0)}$. The Lyapunov exponents are then estimated from the diagonal elements of $R^{(t)}$

$$\lambda_n = \frac{1}{T}\sum_{t=1}^{T} \ln R^{(t)}_{nn}. \tag{18}$$

3. Results and Discussion

In the following we apply an implicit SympGPR model with a product kernel [12]. Due to the periodic topology of the standard map we use a periodic kernel function to construct the covariance matrix in Equation (7) with periodicity 2π in q, whereas a squared exponential kernel is used in P:

$$k(q, q_i, P, P_i) = \sigma_f^2 \exp\left(-\frac{\sin^2((q-q_i)/2)}{2l_q^2}\right)\exp\left(-\frac{(P-P_i)^2}{2l_P^2}\right). \tag{19}$$

Here σ_f^2 specifies the amplitude of the fit and is set in accordance with the observations to 2 $\max(|Y|)^2$, where Y corresponds to the change in coordinates. The hyperparameters l_q, l_P are set to their maximum likelihood value by minimizing the negative log-likelihood given the input data using the L-BFGS-B routine implemented in Python [18]. The noise in observations is set to $\sigma_n^2 = 10^{-16}$. 30 initial data points are sampled from a Halton sequence to ensure good coverage of the training region in the range $[0, 2\pi] \times [0, 2\pi]$ and Equation (2) is evaluated once to obtain the corresponding final data points. Each pair of initial and final conditions constitutes one sample of the training data set. Once the model is trained, it is used to predict subsequent mapping steps for arbitrary initial conditions and to infer the corresponding Jacobians for the calculation of the local Lyapunov exponents. Here, we consider two test cases of the standard map with different values of the stochasticity parameter $K = 0.9$ and $K = 2.0$ (Equation (2)). For each of the test cases, a surrogate model is trained. While in the first case the last KAM surface is not yet broken and therefore the region of stochasticity is still confined in phase space, in the latter case the chaotic region covers a much larger portion of phase space. However, there still exist islands of stability with regular orbits [2]. For $K = 0.9$ the mean squared error (MSE) for the training data is 1.4×10^{-6}, whereas the test MSE after one mapping application is found to be 2.4×10^{-6}. A similar quality of the surrogate model is reached for $K = 2.0$, where the training MSE is 1.6×10^{-7} and the test MSE 2.4×10^{-7}.

3.1. Local Lyapunov Exponents and Orbit Classification

For the evaluation of the distribution of the local Lyapunov exponents with respect to the number of mapping iterations T and phase space position $\boldsymbol{z} = (q, p)$, 1000 points are sampled from each orbit under investigation. In the following, we only consider the maximum local Lyapunov exponent as it determines the predictability of the system. For each of the 1000 points, the LLEs are calculated using Equation (18), where the needed Jacobians are given by the surrogate model by evaluating Equation (9) and solving Equation (11).

Figure 1 shows the distributions for $K = 2.0$, $T = 50$, $T = 100$ and $T = 1000$ for two different initial conditions resulting in a regular and a chaotic orbit. In the regular case the distribution exhibits a sharp peak and with increasing T moves closer to 0. This bias due to the finite number of mapping iterations decreases with $\mathcal{O}(1/T)$ as shown in Figure 2 [25]. For the chaotic orbit, the distribution looks smooth and its median is clearly >0 as expected. For a smaller value of $K = 0.9$ the dynamics in phase space exhibit larger variety with regular, chaotic and also weakly chaotic orbits that remain confined in a small stochastic layer around hyperbolic points. Hence, the transition between regular, weakly chaotic and chaotic orbits is continuous due to the larger variety in phase space. For fewer mapping iterations, possible values of λ are overlapping, thus preventing a clear distinction between confined chaotic and chaotic orbits.

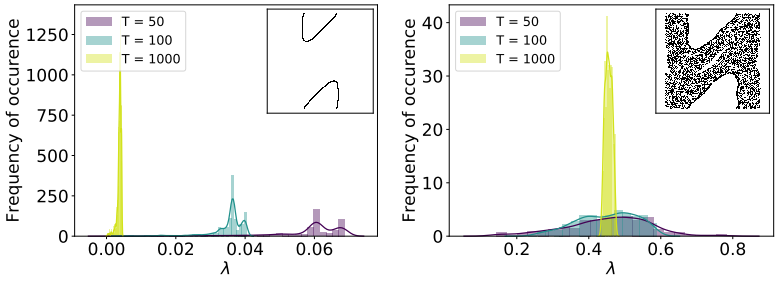


Figure 1. Distribution of local Lyapunov exponents for a (**a**) regular orbit $(q, p) = (1.96, 4.91)$ and (**b**) chaotic orbit $(q, p) = (0.39, 2.85)$ in the standard map with $K = 2.0$

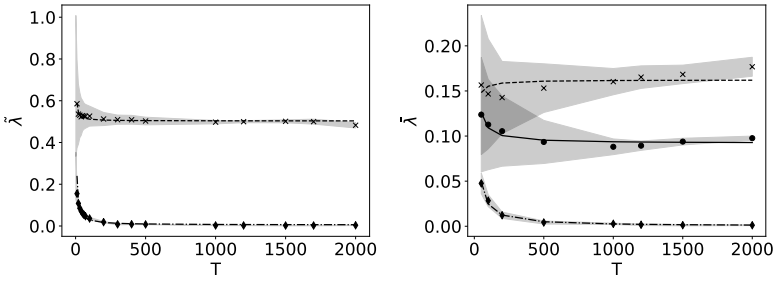
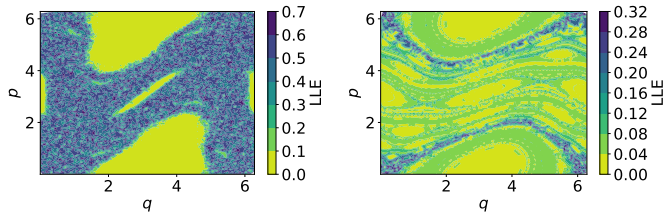


Figure 2. Rate of convergence of the block bias due to finite number of mapping iterations for (**a**) $K = 2.0$ with a regular orbit $(q, p) = (1.96, 4.91)$ (diamond) and a chaotic orbit $(q, p) = (0.39, 2.85)$ (x) and (**b**) $K = 0.9$ with a regular orbit $(q, p) = (1.76, 0.33)$ (diamond), a confined chaotic orbit $(q, p) = (0.02, 2.54)$ (circle) and a chaotic orbit $(q, p) = (0.2, 5.6)$ (x). The graphs show $\tilde{\lambda}_T$, the median of λ_T for each T, with $\tilde{\lambda}_T = \lambda + c/T$ fitted by linear regression of $T\tilde{\lambda}_T$ on T. The gray areas correspond to the standard deviation for 1000 test points.

When considering the whole phase space with 200 orbits with initial conditions sampled from a Halton sequence in the range $[0, \pi] \times [0, 2\pi]$, already $T = 50$ mapping

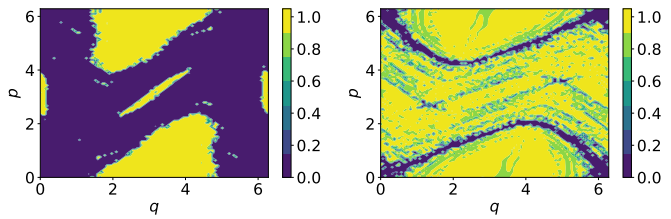
iterations provide insight in the predictability of the standard map (Figure 3). If for a region in phase space the obtained LLE is positive, the predictability in this region is restricted as the instability there is relatively large. If, however, the LLE is close to zero, we can conclude that this region in phase space is governed by regular motion and is therefore highly predictable. For $K = 2.0$ the orbits constituting the chaotic sea have large positive LLEs, whereas islands of stability built by regular orbits show LLEs close to 0. A similar behavior can be observed for $K = 0.9$, where again regions around stable elliptic points feature $\lambda \approx 0$ while stochastic regions exhibit a varying range of LLEs in accordance to Figure 2.

Based on the estimation of the LLEs, a Gaussian Bayesian classifier [26] is used to determine the probability of an orbit being regular, where we assume that LLEs are normally distributed in each class. First, the classifier is trained on LLEs resulting from 200 different initial conditions for T mapping iterations with the corresponding class labels resulting from the chosen reference being the generalized alignment index (GALI) [27]. Then, 10^4 test orbits are sampled from a regular grid in the range $[0, \pi] \times [0, 2\pi]$ with $\Delta q = \Delta p = 2\pi/10$, their LLE is calculated for T mapping iterations and the orbits are then classified. The results for $K = 0.9$ and $K = 2.0$ with $T = 50$ are shown in Figure 4, where the color map indicates the probability that the test orbit is regular. While for $K = 2.0$ the classifier provides a very clear distinction between regular and chaotic regions, the distinction between confined chaotic and regular orbits for $K = 0.9$ is less clear. With increasing number of mapping iterations, the number of misclassifications reduces as depicted in Figure 5. If the predicted probability that an orbit belongs to a certain class is lower than 70%, the prediction is not accepted and the orbit is marked as misclassified. With $K = 0.9$, the percentage of misclassified orbits does not drop below approximately 10%, because the transition between regular and chaotic motion is continuous.



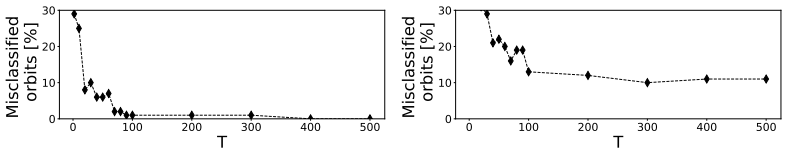
(**a**) (**b**)

Figure 3. Local Lyapunov exponents in phase space of the standard map calculated with $T = 50$ mapping iterations for (**a**) $K = 2.0$, (**b**) $K = 0.9$.



(**a**) (**b**)

Figure 4. Orbit classification in standard map, (**a**) $K = 2.0$, (**b**) $K = 0.9$ for $T = 50$. The color map indicates the probability that the orbit is regular.



(**a**) (**b**)

Figure 5. Percentage of misclassified orbits using a Bayesian classifier trained with 200 orbits for (**a**) $K = 2.0$ and (**b**) $K = 0.9$. 100 test orbits on an equally spaced grid in the range of $[0, \pi] \times [0, 2\pi]$ are classified as regular or chaotic depending on their LLE.

3.2. Sensitivity Analysis

The total Sobol' indices are calculated for the outputs from the symplectic surrogate model (Q, P) using Equation (15) with $N = 2000$ uniformly distributed random points within a box of size $[10^{-3} \times 10^{-3}]$ for each of the $T = 100$ mapping iterations as we are interested in the temporal evolution of the indices. For the standard map at $K = 0.9$ with $d = 2$ input and $D = 2$ output dimensions, 4 total Sobol' indices are obtained: S_q^Q and S_q^P denoting the influence of q and S_p^Q and S_p^P marking the influence of p on the output. We obtain good agreement with an MSE in the order of 10^{-6} between the indices obtained by the surrogate model and those using reference data.

As shown in Figure 6 for three different initial conditions for $K = 0.9$ depending on the orbit type, either chaotic or regular, the sensitivity indices behave differently. In case of a regular orbit close to a fixed point, S_j^i are oscillating, indicating that both input variables have similar influence on average. Getting further from the fixed point, closer to the border of stability, the influence of q gets bigger. This, however, is in contrast to the behavior in the chaotic case, where initially the variance in p has larger influence on the model output. However, when observing the indices over longer periods of time, both variables have similar influence. In Movie S01 in the supplemental material, the time evolution of all four total Sobol' indices obtained for the standard map are shown in phase space. Each frame is averaged over 10 subsequent mapping iterations. One snapshot is shown in Figure 7. The observation of the whole phase space sustains the findings in Figure 6.

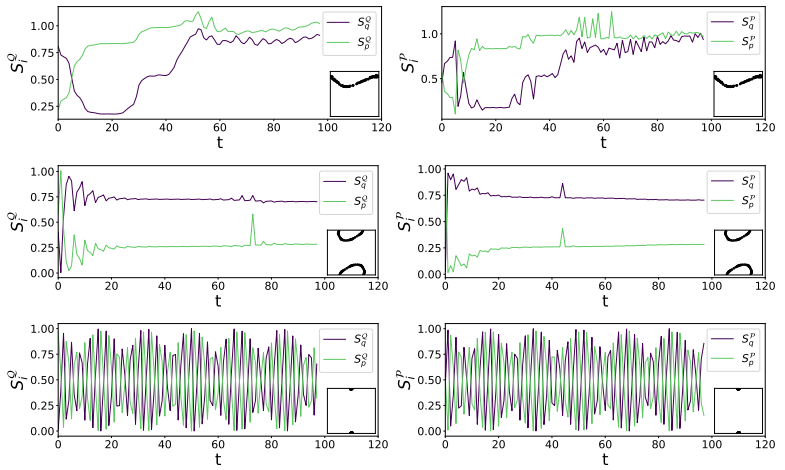


Figure 6. Total Sobol' indices as a function of time for three orbits of the standard map with $K = 0.9$—upper: chaotic orbit $(q, p) = (0.2, 5.6)$, middle: regular orbit $(q, p) = (1.76, 0.33)$, lower: regular orbit very close to fixed point $(q, p) = (\pi, 0.1)$.

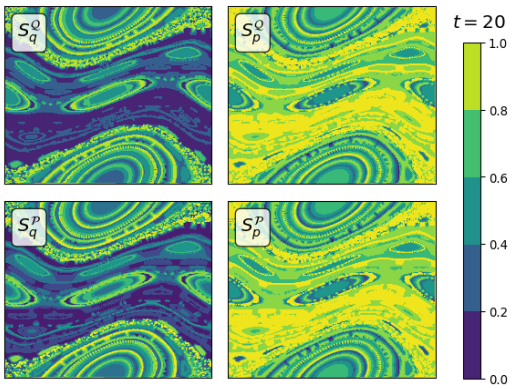


Figure 7. Total Sobol' indices (Equation (15)) for the standard map with $K = 0.9$ averaged from $t = 20$ to $t = 30$.

4. Conclusions

We presented an approach for orbit classification in Hamiltonian systems based on a structure preserving surrogate model combined with early classification based on local Lyapunov exponents directly available from the surrogate model. The approach was tested on two cases of the standard map. Depending on the perturbation strength, we either see a continuous transition from regular to chaotic orbits for $K = 0.9$ or a sharp separation between those two classes for higher perturbation strengths. This also impacts the classification results obtained from a Bayesian classifier. The presented method is applicable to chaotic Hamiltonian systems and is especially useful when a closed form expression for Poincaré maps is not available. Also, the accompanying sensitivity analysis provides valuable insight: in transition regions between regular and chaotic motion the Sobol' indices for time-series can be used to analyze the influence of input variables.

Author Contributions: Conceptualization, K.R., C.G.A., B.B. and U.v.T.; methodology, K.R., C.G.A., B.B. and U.v.T.; software, K.R.; validation, K.R., C.G.A., B.B. and U.v.T.; formal analysis, K.R., C.G.A., B.B. and U.v.T.; writing—original draft preparation, K.R.; visualization, K.R.; supervision, C.G.A., U.v.T. and B.B.; funding acquisition, C.G.A, B.B. and U.v.T. All authors have read and agreed to the published version of the manuscript.

Funding: The present contribution is supported by the Helmholtz Association of German Research Centers under the joint research school HIDSS-0006 "Munich School for Data Science-MUDS" and the Reduced Complexity grant No. ZT-I-0010.

Data Availability Statement: The data and source code that support the findings of this study are openly available [28] and maintained on https://github.com/redmod-team/SympGPR.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ott, E. Chaos in Hamiltonian systems. In *Chaos in Dynamical Systems*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2002; pp. 246–303. [CrossRef]
2. Lichtenberg, A.; Lieberman, M. *Regular and Chaotic Dynamics*; Springer: New York, NY, USA, 1992.
3. Albert, C.G.; Kasilov, S.V.; Kernbichler, W. Accelerated methods for direct computation of fusion alpha particle losses within, stellarator optimization. *J. Plasma Phys.* **2020**, *86*, 815860201. [CrossRef]
4. Eckmann, J.P.; Ruelle, D. Ergodic theory of chaos and strange attractors. *Rev. Mod. Phys.* **1985**, *57*, 617–656. [CrossRef]
5. Benettin, G.; Galgani, L.; Giorgilli, A.; Strelcyn, J. Lyapunov Characteristic Exponents for smooth dynamical systems and for Hamiltonian systems; A method for computing all of them. Part 1: Theory. *Meccanica* **1980**, *15*, 9–20. [CrossRef]
6. Benettin, G.; Galgani, L.; Giorgilli, A.; Strelcyn, J. Lyapunov Characteristic Exponents for smooth dynamical systems and for Hamiltonian systems; A method for computing all of them. Part 2: Numerical application. *Meccanica* **1980**, *15*, 21–30. [CrossRef]

7. Abarbanel, H.; Brown, R.; Kennel, M. Variation of Lyapunov exponents on a strange attractor. *J. Nonlinear Sci.* **1991**, *1*, 175–199. [CrossRef]
8. Abarbanel, H.D.I. Local Lyapunov Exponents Computed From Observed Data. *J. Nonlinear Sci.* **1992**, *2*, 343–365. [CrossRef]
9. Eckhardt, B.; Yao, D. Local Lyapunov exponents in chaotic systems. *Phys. D Nonlinear Phenom.* **1993**, *65*, 100–108. [CrossRef]
10. Amitrano, C.; Berry, R.S. Probability distributions of local Liapunov exponents for small clusters. *Phys. Rev. Lett.* **1992**, *68*, 729–732. [CrossRef] [PubMed]
11. Arnold, V. *Mathematical Methods of Classical Mechanics*; Springer: New York, NY, USA, 1989; Volume 60.
12. Rath, K.; Albert, C.G.; Bischl, B.; von Toussaint, U. Symplectic Gaussian process regression of maps in Hamiltonian systems. *Chaos* **2021**, *31*, 053121. [CrossRef]
13. Skokos, C. The Lyapunov Characteristic Exponents and Their Computation. In *Dynamics of Small Solar System Bodies and Exoplanets*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 63–135. [CrossRef]
14. Sobol, I. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Math. Comput. Simul.* **2001**, *55*, 271–280. [CrossRef]
15. Sobol, I. On sensitivity estimation for nonlinear math. models. *Matem. Mod.* **1990**, *2*, 112–118.
16. Saltelli, A.; Annoni, P.; Azzini, I.; Campolongo, F.; Ratto, M.; Tarantola, S. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Comput. Phys. Commun.* **2010**, *181*, 259–270. [CrossRef]
17. Chirikov, B.V. A universal instability of many-dimensional oscillator systems. *Phys. Rep.* **1979**, *52*, 263–379. [CrossRef]
18. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2005.
19. Álvarez, M.A.; Rosasco, L.; Lawrence, N.D. Kernels for Vector-Valued Functions: A Review. *Found. Trends Mach. Learn.* **2012**, *4*, 195–266. [CrossRef]
20. Solak, E.; Murray-smith, R.; Leithead, W.E.; Leith, D.J.; Rasmussen, C.E. Derivative Observations in Gaussian Process Models of Dynamic Systems. In *NIPS Proceedings 15*; Becker, S., Thrun, S., Obermayer, K., Eds.; MIT Press: Cambridge, MA, USA, 2003; pp. 1057–1064.
21. Eriksson, D.; Dong, K.; Lee, E.; Bindel, D.; Wilson, A. Scaling Gaussian process regression with derivatives. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018 (NeurIPS 2018), Montréal, QC, Canada*, 3–8 December 2018; Curran Associates, Inc.: Red Hook, NY, USA, 2018; pp. 6868–6878.
22. Sudret, B. Global sensitivity analysis using polynomial chaos expansions. *Reliab. Eng. Syst. Saf.* **2008**, *93*, 964–979. [CrossRef]
23. Marrel, A.; Iooss, B.; Laurent, B.; Roustant, O. Calculations of Sobol indices for the Gaussian process metamodel. *Reliab. Eng. Syst. Saf.* **2009**, *94*, 742–751. [CrossRef]
24. Geist, K.; Parlitz, U.; Lauterborn, W. Comparison of Different Methods for Computing Lyapunov Exponents. *Prog. Theor. Phys.* **1990**, *83*, 875–893. [CrossRef]
25. Ellner, S.; Gallant, A.; McCaffrey, D.; Nychka, D. Convergence rates and data requirements for Jacobian-based estimates of Lyapunov exponents from data. *Phys. Lett. A* **1991**, *153*, 357–363. [CrossRef]
26. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
27. Skokos, C.; Bountis, T.; Antonopoulos, C. Geometrical properties of local dynamics in Hamiltonian systems: The Generalized Alignment Index (GALI) method. *Phys. D Nonlinear Phenom.* **2007**, *231*, 30–54. [CrossRef]
28. Rath, K.; Albert, C.; Bischl, B.; von Toussaint, U. SympGPR v1.1: Symplectic Gaussian process regression. *Zenodo* **2021**. [CrossRef]

3.4 Data augmentation for disruption prediction via robust surrogate models

Main novelty:
We introduced a robust Student-t process regression model in state space formulation via Bayesian filtering to augment the database for training large ML-models for disruption prediction. The model neglects signal interdependecies but accounts for signal correlations and cross-correlations via coloring transformations in a post-processing step.

Contributing article:
Rath, K., Rügamer, D., Bischl, B., von Toussaint, U., Rea, C., Maris, A., Granetz, R., and Albert, C. G. (2022). Data augmentation for disruption prediction via robust surrogate models. *Journal of Plasma Physics*, 88(5):895880502

Author contributions:
Udo von Toussaint, Cristina Rea and Robert Granetz devised the conceptual idea of the project. Katharina Rath designed the model, implemented the algorithms, identified suitable training data, performed the numerical experiments, designed the validation and evaluation routines and wrote the paper. Andrew Maris helped with the retrieval of training data. David Rügamer, Christopher Albert, Udo von Toussaint, Bernd Bischl, Cristina Rea, Robert Granetz and Andrew Maris gave valuable input throughout the project and suggested several notable modifications.

J. Plasma Phys. (2022), *vol*. 88, 895880502
Published by Cambridge University Press

doi:10.1017/S0022377822000769

Data augmentation for disruption prediction via robust surrogate models

Katharina Rath[1,2,†], **David Rügamer**[1,3], **Bernd Bischl**[1], **Udo von Toussaint**[2], **Cristina Rea**[4], **Andrew Maris**[4], **Robert Granetz**[4] and **Christopher G. Albert**[5]

[1]Department of Statistics, Ludwig-Maximilians-Universität München, Germany

[2]Max-Planck-Institut für Plasmaphysik, Garching, Germany

[3]Institute of Statistics, RWTH Aachen University, Germany

[4]Plasma Science and Fusion Center, Massachusetts Institute of Technology, Cambridge, MA, USA

[5]Fusion@OEAW, Institute of Theoretical and Computational Physics, Graz University of Technology, Austria

(Received 18 May 2022; revised 7 August 2022; accepted 8 August 2022)

The goal of this work is to generate large statistically representative data sets to train machine learning models for disruption prediction provided by data from few existing discharges. Such a comprehensive training database is important to achieve satisfying and reliable prediction results in artificial neural network classifiers. Here, we aim for a robust augmentation of the training database for multivariate time series data using Student *t* process regression. We apply Student *t* process regression in a state space formulation via Bayesian filtering to tackle challenges imposed by outliers and noise in the training data set and to reduce the computational complexity. Thus, the method can also be used if the time resolution is high. We use an uncorrelated model for each dimension and impose correlations afterwards via colouring transformations. We demonstrate the efficacy of our approach on plasma diagnostics data of three different disruption classes from the DIII-D tokamak. To evaluate if the distribution of the generated data is similar to the training data, we additionally perform statistical analyses using methods from time series analysis, descriptive statistics and classic machine learning clustering algorithms.

Key words: fusion plasma, plasma instabilities

1. Introduction

Disruptions pose serious challenges to the operation and design of tokamaks. Due to rapidly growing instabilities, thermal and magnetic energy is rapidly lost during a disruption, the magnetic confinement of the plasma is destroyed and energy is deposited into the confining vessel, potentially causing serious damages. Hence, to maintain a reliable fusion operation, disruption mitigation mechanisms should be triggered with sufficient warning time prior to the disruption. Recent advances on real-time disruption

† Email address for correspondence: katharina.rath@ipp.mpg.de

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

prediction have been made using machine learning (Berkery *et al.* 2017; Rea & Granetz 2018; Kates-Harbeck, Svyatkovskiy & Tang 2019; Pau *et al.* 2019; Rea *et al.* 2019, 2020; Aymerich *et al.* 2022). Disruption prediction is a challenging task for various reasons. One of them is the imbalanced data situation; for some disruption classes, only a few measurements are available, making it difficult to obtain robust results. This is challenging, especially when working with neural networks, as they require a large training data set in order to give satisfying results and to avoid overfitting (see e.g. Aggarwal 2018). However, generating such an amount of training data from additional discharges is expensive and also potentially harmful for the reactor. Particularly with regard to future reactors such as ITER or SPARC, a sufficient data set will not be available at the time these reactors start operating.

Data augmentation is one possibility to balance the training data set by creating rare disruption events and thereby improving the prediction performance of machine learning models. The aim of data augmentation is to produce an arbitrarily large number of artificial samples that have the same statistical properties as the original small data set. Especially in the context of image classification, data augmentation is a widely used technique to improve the prediction accuracy and avoid overfitting (Shorten & Khoshgoftaar 2019). Commonly used methods are random transformation-based approaches, such as cropping or flipping. However, these methods are not expedient for the task at hand, as time dependencies and the causal structure of physical signals are destroyed by such transformations (Iwana & Uchida 2021; Wen *et al.* 2021). More elaborate methods for multivariate time series generation using neural networks (Yoon, Jarrett & van der Schaar 2019) require substantially more samples per class than usually available for disruption prediction. Other advanced data augmentation methods are based on decomposition into trend, seasonal/periodic signal and noise (Cleveland *et al.* 1990; Wen *et al.* 2019) or involve statistical modelling of the dynamics using, e.g. mixture autoregressive models (Kang, Hyndman & Li 2020).

Here, we tackle the above-mentioned challenges by relying on a non-parametric Bayesian approach to design the multivariate surrogate model based on Student t process regression (Shah, Wilson & Ghahramani 2014; Roth *et al.* 2017) to generate additional data. This model is closely related to the more commonly used Gaussian process regression (Williams & Rasmussen 1996). One drawback of standard Gaussian processes regression is the assumption of Gaussian noise, which is inaccurate due to outliers in the present application case. This results in unreliable uncertainty estimates. There have been attempts to make Gaussian process regression robust against outliers by using a Student t distributed noise model and relying on approximate inference (Neal 1997; Vanhatalo, Jylanki & Vehtari 2009). However, our approach rather builds on Student t processes with an analytic inference scheme (Shah *et al.* 2014) that also allows a heavy tailed noise distribution and gives robust results even for noisy data corrupted by outliers.

Another challenge imposed by high-resolution time series data is the computational complexity of multivariate Gaussian or Student t process regression of $O(N^3)$, where $N = DT$ is the number of training data points given by the product of dimensions D and time steps T of the multivariate time series. For typical values of $N > 1000$, traditional regression requires too much computing time. We instead use the state space formulation of a Student t process as a linear time invariant stochastic differential equation, which can be solved using a corresponding filter and smoother (Solin & Särkkä 2015). In the case of a Gaussian process, the analogous approach is the well-known Kalman filter and Rauch–Tung–Striebel (RTS) smoother (Särkkä 2013; Särkkä & Solin 2019). This ansatz reduces the computational complexity to $O(N)$, making it also suitable for high-resolution time series.

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

Here, we are working with a multi-output state space model to generate multivariate time series. We first assume that dimensions of the multivariate time series are not correlated. This is done to avoid the requirement of optimizing all hyperparameters at the same time, which is practically unfeasible due to the limited amount of available data. To still account for signal interdependencies, we then induce correlations and cross-correlations via colouring transformations in a post-processing step.

To balance the training data set, we use several local surrogate models to generate data coming from different disruption classes. From a small set – usually less than 10 discharges – of multivariate time series with D measurement signals coming from one disruption class with similar operating conditions, we estimate the posterior distribution. We then sample from the trained model in order to generate similar data that enlarge the training database. To evaluate if the generated samples are from the same distribution as the training data, we use several methods from time series analysis, descriptive statistics and clustering algorithms to show that generated and training samples are almost indistinguishable.

2. Methods

2.1. *Student t processes*

Student t processes (TPs) are a generalization of the widely used Gaussian processes (GPs) (Williams & Rasmussen 1996; Shah *et al.* 2014). TPs allow for a heavy tailed noise distribution (estimated by an additional hyperparameter $\nu > 2$) and therefore put less weight on outliers compared with GPs (Shah *et al.* 2014; Roth *et al.* 2017). This is illustrated in figure 1 for a test case of synthetic data corrupted by outliers. As in GP regression, we consider a set of N training observations $\mathcal{D} = \{(t_i, y_i)\}_{i=1}^{T}$ of scalar function values $y_i = f(t_i)$ plus measurement noise at training points t_i with $i = 0, 1, \ldots, T$ (in our case, time). We model these data points using a TP with zero mean and covariance function $k(t, t')$,

$$f(t) \sim \mathcal{TP}(0, k(t, t'), \nu). \tag{2.1}$$

Similar to the GP, a kernel function $k(t, t')$ quantifies the covariance between values of f at times (t, t') and yields an $N \times N$ covariance matrix $\boldsymbol{K}$ with components $K_{ij} = k(t_i, t_j)$ for the random vector of all observed y_i. Kernel hyperparameters determine further details, e.g. a length scale l quantifies how fast correlations vanish with increasing distance in t. The additional hyperparameter $\nu > 2$ corresponds to the degrees of freedom that specify the noise distribution. The predicted distribution of a scalar output $f(t_*)$ at test point t_* is given in closed form by

$$\mathbb{E}[f(t_*)] = \boldsymbol{k}_*^\top \boldsymbol{K}_y^{-1} \boldsymbol{y}, \tag{2.2}$$

$$\mathbb{V}[f(t_*)] = \frac{\nu - 2 + \boldsymbol{y}^\top \boldsymbol{K}_y^{-1} \boldsymbol{y}}{\nu - 2 + N} (k_{**} - \boldsymbol{k}_*^\top \boldsymbol{K}_y^{-1} \boldsymbol{k}_*), \tag{2.3}$$

where $\boldsymbol{K}_y = \boldsymbol{K} + \sigma_n^2 \boldsymbol{I}$ is the measurement noise parametrized by the noise variance σ_n^2. Here, $\boldsymbol{k}_*$ is an N-dimensional vector with the ith entry being $k(t_*, t_i)$; $k_{**} = k(t_*, t_*)$ describes the covariance between training and test data and the variance at the test point t_*. In contrast to GP regression, the posterior variance $\mathbb{V}[f(t_*)]$ of the prediction explicitly depends on training observations by taking data variability into account and results in more reliable uncertainty estimates. An analogous expression to (2.3) is obtained for the covariance matrix between predictions at multiple t_* (Shah *et al.* 2014).

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

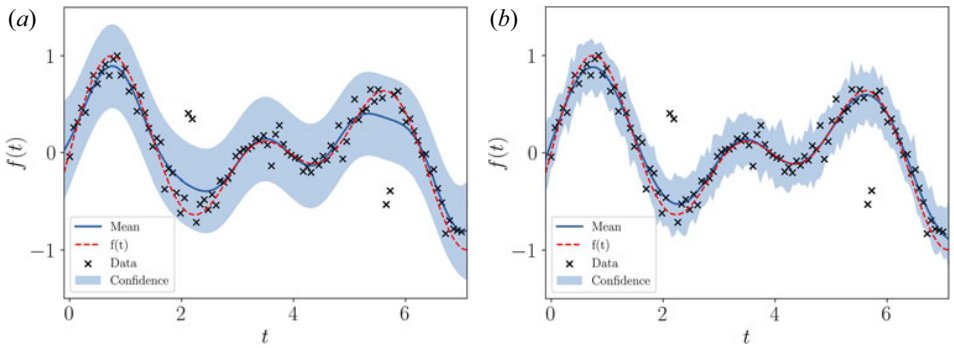


FIGURE 1. Predicted mean and 95 % confidence band with (*a*) GP and (*b*) TP trained on $N = 100$ training data points following $f(t) = \sin(2t)\cos(0.4t)$ corrupted by Gaussian noise $0.1\mathcal{N}(0, 1)$, with several outliers.

2.2. *State space formulation*

As in GP regression, the computational complexity increases with $O(N^3)$, as an inversion of the covariance matrix via Cholesky factorization is necessary to train TPs (Williams & Rasmussen 1996). This makes GP and also TP regression unfavourable for high-resolution time series data. However, as shown by Solin & Särkkä (2015), the TP regression problem can be reformulated as an mth-order linear time invariant stochastic differential equation (SDE)

$$\frac{\mathrm{d}\hat{\boldsymbol{f}}(t)}{\mathrm{d}t} = \boldsymbol{F}\hat{\boldsymbol{f}}(t) + \boldsymbol{L}\boldsymbol{w}(t), \tag{2.4}$$

$$f(t_i) = \boldsymbol{H}\hat{\boldsymbol{f}}(t_i), \tag{2.5}$$

where $\hat{\boldsymbol{f}}(t) = (f(t), \mathrm{d}f(t)/\mathrm{d}t, \ldots, \mathrm{d}^{m-1}f(t)/\mathrm{d}t^{m-1})^\top$, the feedback matrix $\boldsymbol{F}$ and noise effect matrix $\boldsymbol{L}$ are derived from the underlying TP, $\boldsymbol{H} = (1, 0, \ldots, 0)$ is the measurement or observation matrix and $\boldsymbol{w}(t)$ is a vector of white noise processes with spectral density $\gamma\boldsymbol{Q}$, where γ is a scaling factor (Solin & Särkkä 2015).

To solve this SDE for discrete points in time by estimating the posterior distribution $p(\hat{\boldsymbol{y}}_{0:T}|\boldsymbol{y}_{1:T})$ of the latent state $\hat{\boldsymbol{y}}_{0:T}$ given noisy observations $\boldsymbol{y}_{1:T}$, we use the corresponding Student t filter and smoother as outlined in Solin & Särkkä (2015). Here, the posterior is estimated by using marginal distributions: (i) filtering distribution $p(\hat{\boldsymbol{y}}_t|\boldsymbol{y}_{1:t})$ given by the update step in Algorithm 1, (ii) prediction distribution $p(\hat{\boldsymbol{y}}_{t+k}|\boldsymbol{y}_{1:t})$ given by the prediction step in Algorithm 1 for k steps after the current time step t and (iii) smoothing distributions $p(\hat{\boldsymbol{y}}_t|\boldsymbol{y}_{1:T})$ for $t < T$ given by Algorithm 2 (Särkkä 2013). The initial distribution is determined by the prior state mean given by the measurements at $t = 0$ and prior state covariance $\boldsymbol{P}_0$ given by the stationary covariance (Solin & Särkkä 2015). The augmented states $\mathrm{d}f/\mathrm{d}t$ that are not measured and noise are initialized with 0.

For example, the state space formulation of the Matérn 3/2 kernel is given by the following expressions for feedback, noise effect matrix and spectral density (Särkkä & Solin 2019):

$$\boldsymbol{F} = \begin{pmatrix} 0 & 1 & 0 \\ -\lambda^2 & -2\lambda & 0 \\ 0 & 0 & -\infty \end{pmatrix}, \quad \boldsymbol{P}_0 = \begin{pmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2\lambda^2 & 0 \\ 0 & 0 & \sigma_n^2 \end{pmatrix}, \quad \boldsymbol{H} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{L} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \tag{2.6a–d}$$

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

where $\lambda = \sqrt{3}/l$. Hyperparameters l, σ^2, σ_n^2 and ν needed in the Student t filter algorithm are estimated by minimizing the negative log likelihood (Solin & Särkkä 2015). The log likelihood is sequentially calculated using the Student t filter (Algorithm 1). When the hyperparameters are optimized, the predictive distribution is first calculated via Algorithm 1 and then smoothed using Algorithm 2. In order to include the noise model with σ_n^2 corresponding to $\boldsymbol{K}_y = \boldsymbol{K} + \sigma_n^2 \boldsymbol{I}$ in traditional TP regression, the SDE is directly augmented by the entangled noise model. As the model is not only augmented with the noise model, but also with the first derivative of the target function we want to predict, we can immediately infer $\mathrm{d}f(t)/\mathrm{d}t$ from the given observations y.

Here, the task at hand concerns multivariate time series $\boldsymbol{Y}$ with multiple measurements n with D dimensions where the ith row is $\boldsymbol{y}_i = \boldsymbol{f}(t_i)$ at every time step t_i. To facilitate the training of the model, we consider an uncorrelated model, such that the associated random processes are not correlated. In traditional GP/TP regression, this corresponds to a multi-output model with a block-diagonal covariance matrix. The multi-output state space model to estimate $p(\hat{\boldsymbol{Y}}_{0:T}|\boldsymbol{Y}_{1:T})$ is built by stacking the univariate SDE models resulting in a block-diagonal structure for feedback and covariance matrices. Then, the dynamics of $\boldsymbol{y}_i$ is independent. We sample uncorrelated multivariate time series from this model and apply colouring transformations in a following post-processing step to account for correlations (§ 2.4). Each dimension has its own set of hyperparameters in order to grasp the dynamics that happen on different time scales. The measurement covariance matrix $\boldsymbol{R}$ (Algorithm 1) is estimated using the covariance of n measurements for each dimension at every time step.

2.3. *Student t sampler*

To sample from the estimated posterior distribution, we employ a Student t sampler, which is a modified version of the sampling technique presented by Durbin & Koopman (2002). First, we draw a t distributed random sequence $\hat{\boldsymbol{X}}_{0:T} = \hat{\boldsymbol{x}}_{i,0:T}$ from the prior estimated by the trained Student t model. These sequences are initialized by $\mathcal{T}(0, \boldsymbol{P}_0)$ and then filtered using Algorithm 1 and smoothed via Algorithm 2, which yields $\mathbb{E}(\hat{\boldsymbol{X}}_{0:T}|\boldsymbol{Y}_{1:T}^+)$ where $\boldsymbol{Y}_{1:T}^+ = \boldsymbol{H}\hat{\boldsymbol{X}}_{0:T}$, with the stacked measurement matrix $\boldsymbol{H} = (\mathrm{l}, 0, 0)$ that extracts only the first component of $\hat{\boldsymbol{x}}_t$ in every time step t. Here, $\boldsymbol{Y}_{1:T}^+$ are data associated with the filtered and smoothed sequence $\hat{\boldsymbol{X}}_{0:T}$ given by (A2). Finally, to obtain a random sequence $\bar{\boldsymbol{Y}}_{0:T} = \bar{\boldsymbol{y}}_{i,0:T} \sim p(\hat{\boldsymbol{Y}}_{0:T}|\boldsymbol{Y}_{1:T})$, we combine

$$\bar{\boldsymbol{Y}}_{1:T} = \boldsymbol{H}(\mathbb{E}(\hat{\boldsymbol{Y}}_{0:T}|\boldsymbol{Y}_{1:T}) + \hat{\boldsymbol{X}}_{0:T} - \mathbb{E}(\hat{\boldsymbol{X}}_{0:T}|\boldsymbol{Y}_{1:T}^+)), \tag{2.7}$$

where H extracts the first component of $\hat{\boldsymbol{y}}_t$ in every time step t. This procedure gives a D-dimensional multivariate time series for T time steps.

2.4. *Post-processing*

Given the trained model, we sample data $\bar{\boldsymbol{Y}}_{1:T}$ from the estimated posterior, where rows are dimensions $\bar{y}_i$ and columns are time steps; $\bar{\boldsymbol{Y}}_{1:T}$ can be split into a mean given by the smoothing distribution and deviations due to the sampling. Correlations between dimensions D of the generated data are not reproduced correctly with the uncorrelated model. However, with three different post-processing methods of increasing complexity compared in the results, we aim to handle correlations.

We thus want to inscribe the average covariance $\boldsymbol{\Sigma}$ over all samples empirically observed in the training data $\boldsymbol{Y}_{1:T}$ into the generated data $\bar{\boldsymbol{Y}}_{1:T}$. However, the covariance matrix $\bar{\boldsymbol{\Sigma}}$ of $\bar{\boldsymbol{Y}}_{1:T}$ has small non-zero off-diagonal elements. Therefore, we first perform a

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

Zero Components Analysis (ZCA) whitening (also known as Mahalanobis) transformation (see e.g. Kessy, Lewin & Strimmer 2018):

$$\boldsymbol{Z} = \bar{\boldsymbol{\Sigma}}^{-1/2}\bar{\boldsymbol{Y}}. \tag{2.8}$$

The transformed data $\boldsymbol{Z}$ have a diagonal covariance matrix $\boldsymbol{\Lambda}_Z$, with unit variances on the diagonal. We then colour the generated data via a colouring transformation (Kessy *et al.* 2018)

$$\tilde{\boldsymbol{Y}} = \boldsymbol{\Sigma}^{1/2}\boldsymbol{Z} = \boldsymbol{\Sigma}^{1/2}\bar{\boldsymbol{\Sigma}}^{-1/2}\bar{\boldsymbol{Y}}, \tag{2.9}$$

obtaining data $\tilde{\boldsymbol{Y}}$, which now have the same (temporally local) covariance as the training data $\boldsymbol{Y}$.

Another possibility is to directly take the distribution of the training data covariance matrix $\boldsymbol{\Sigma}$ over samples into account by using samples from a corresponding multivariate Gaussian distribution as data covariance matrices. This generates variation in the covariance of the generated data, especially if there are local differences between the samples. However, on average for a large enough sample size, we recover the training data covariance matrix $\boldsymbol{\Sigma}$.

To also take time-lagged correlations into account, we must adjust not only covariances but also cross-covariances in our generated data. Therefore, we use the cross-covariance matrix given by

$$\bar{\boldsymbol{\Sigma}}_{c,rs}(t_1, t_2) = \mathbb{E}[(\bar{y}_{r,t_1} - \mu_{r,t_1})(\bar{y}_{s,t_2} - \mu_{s,t_2})], \tag{2.10}$$

where the expected value $\mathbb{E}[\cdot]$ is estimated by averaging over all combinations of lags $t_1 - t_2$ in addition to the sample mean. Here, $\mu_{i,t}$ is the expected value of $\bar{y}_{i,t}$. To decorrelate and colour the data in the way described above, we formally use a global covariance matrix $\boldsymbol{\Sigma}_g$ of size $DT \times DT$ involving correlations both over time and across dimensions of the multivariate time series. The global covariance matrix is a periodic block matrix given by

$$\boldsymbol{\Sigma}_{g,(t_1D+r)(t_2D+s)} = \boldsymbol{\Sigma}_{c,rs}(t_1, t_2) \tag{2.11}$$

for the cross-covariance $\boldsymbol{\Sigma}_c$ with lag. The generated data is coloured using the global covariance matrix:

$$\tilde{\boldsymbol{Y}} = \boldsymbol{\Sigma}_g^{1/2}\boldsymbol{Z} = \boldsymbol{\Sigma}_g^{1/2}\overline{\boldsymbol{\Sigma}_g}^{-1/2}\bar{\boldsymbol{Y}}. \tag{2.12}$$

This incorporates the empirical cross-covariance for all time lags and between all dimensions D of the generated data.

3. Evaluation of generated data

As the generated data serve as augmented training data for later analyses, statistical properties of the original training data should be reflected in the generated data. Therefore, we perform statistical tests to check if training and generated share key statistical properties.

3.1. *Distribution and Wasserstein distance*

To measure the distance between the distribution of the training and the generated data, we use the Wasserstein-1 metric (Villani 2008)

$$W_1(P, V) = \inf_{\gamma \in \Gamma(P,V)} \int_{\mathbb{R}\times\mathbb{R}} |x - y| \, \mathrm{d}\gamma(x, y), \tag{3.1}$$

where $\Gamma(P, V)$ denotes the set of all probability distributions on $\mathbb{R} \times \mathbb{R}$, with P, V being its marginals. The minimizer γ of (3.1) denotes the optimal transport plan to transport

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

P to V. We compare each signal separately and average the corresponding Wasserstein distances. Although the problem concerns time series data, we discard all time information and only consider the global distribution of the data due to the small amount of available training data samples.

3.2. *Maximum mean discrepancy two-sample test*

In addition to the Wasserstein distance, we perform the kernel two-sample test (Gretton *et al.* 2012) for each signal (again discarding time information). The null hypothesis we want to test is that both n training data $\boldsymbol{y}_{i,1:T}$ and m generated data samples $\tilde{\boldsymbol{y}}_{i,1:T}$ follow the same distribution P. We use the maximum mean discrepancy (MMD) test statistic via a kernel g

$$\mathrm{MMD}^2 = \frac{1}{n(n-1)}\sum_{i,j=1}^{n} g(\boldsymbol{y}_{i,1:T}, \boldsymbol{y}_{j,1:T}) + \frac{1}{m(m-1)}\sum_{i,j=1}^{m} g(\tilde{\boldsymbol{y}}_{i,1:T}, \tilde{\boldsymbol{y}}_{i,1:T}) - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} g(\boldsymbol{y}_{i,1:T}, \tilde{\boldsymbol{y}}_{i,1:T}), \tag{3.2}$$

where $g(x, y) = \exp(-||x-y||^2/(2\sigma^2))$ with $\sigma = \mathrm{Median}(|\Upsilon_i - \Upsilon_j|)/2$ and Υ is the combined sample of $\boldsymbol{y}_{i,1:T}$ and $\tilde{\boldsymbol{y}}_{i,1:T}$. To estimate a threshold for the acceptance of the null hypothesis for a given confidence level, bootstrapping is performed via mixing samples $\boldsymbol{y}_{i,1:T}$ and $\tilde{\boldsymbol{y}}_{i,1:T}$, which generates a distribution with 10 000 samples that satisfies the null hypothesis. Finally, we can estimate a p-value for the MMD of the generated data distributions.

3.3. *Auto- and cross-correlation*

To evaluate if the generated data reflect the temporal dependencies of the training data, we calculate auto- and cross-correlations ρ_{rs} for training and generated data by normalizing the cross-covariance $\boldsymbol{\Sigma}_c$ in (2.10) by $1/(\sigma_{r,t_1}\sigma_{s,t_2})$. Here, $\sigma_{s,t}$ is the standard deviation of $\tilde{y}_{s,t}$. If $r = s$, this diagnostic becomes the auto-correlation – see, e.g. Park (2017). For $t_1 = t_2$, the local correlation matrix follows. We evaluate the mean squared error (MSE) to the auto- and cross-correlation of the training data. Evidently, the global colouring transformation (2.10) produces a perfect match in this diagnostic.

3.4. *Power spectral density*

All frequencies that are present in the training data set should also appear in the generated data. This can be evaluated using the power spectral density (PSD), which provides an estimate of power distribution across the frequency of a signal. We evaluate the mean squared error between the PSD of the training data and generated data.

3.5. *Embedding via kernel principal component analysis*

We apply two-dimensional (2-D) kernel principal component analysis (PCA) on the training data with flattened temporal dimension and project the generated data onto the first two principal components of the training data to evaluate the embedding and visualize if both training and generated data lie on the same submanifold (Schölkopf, Smola & Müller 1998). In all test cases, a polynomial kernel of degree 3 with optimized kernel coefficient (minimization of the reconstruction error) is used.

The distance between the embedded distributions of training and generated data is measured by using the sliced Wasserstein distance that takes advantage of the very efficient

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

calculation of 1-D Wasserstein distances (Bonneel *et al.* 2015; Flamary *et al.* 2021). The multivariate distribution is sliced and randomly projected on a 1-D subspace, and the corresponding 1-D Wasserstein distances are averaged to obtain an estimation for the multivariate distribution. With an increasing number of projections, the sliced Wasserstein distance converges. Here, we use 10^3 projections to estimate the distance W_{emb} between the embedded distributions.

3.6. *Multivariate functional PCA*

For the evaluation of the correctly represented temporal evolution of the generated data, we apply multivariate functional principal component analysis (mfPCA) on the training data and project the generated data onto the eigenbasis of the training data (Happ & Greven 2018). Then, we reconstruct both training and generated data with the same eigenbasis and evaluate the variance of the residuals.

3.7. *Dynamic time warping*

For time series comparison, dynamic time warping (DTW) is widely used to measure the similarity between two temporal sequences $\boldsymbol{y}_{i,1:T}$ and $\tilde{\boldsymbol{y}}_{j,1:T}$ (Berndt & Clifford 1994). This metric is formulated as an optimization problem

$$\text{DTW}(\boldsymbol{y}_{i,1:T}, \tilde{\boldsymbol{y}}_{j,1:T}) = \min_{\gamma} \sqrt{\sum_{(i,j)\in\gamma} d(\boldsymbol{y}_i, \tilde{\boldsymbol{y}}_j)^2}, \tag{3.3}$$

where γ is the alignment path such that the Euclidean distance between $\boldsymbol{y}_{i,1:T}$ and $\tilde{\boldsymbol{y}}_{j,1:T}$ is minimal. Hence, DTW gives the distance between two time series with the best temporal alignment. We compare each training data sample with each generated data sample and use the mean to compare different post-processing methods.

3.8. *Self-organizing maps on time series*

Finally, we apply time series clustering based on DTW self-organizing maps (SOMs) on both the training and generated data (Vettigli 2018). If the generated data are a potentially useful extension of the training data, the clustering should show similar results. Therefore, we compute a clustering model on the training data and use the trained model to predict cluster labels of both the training and generated data. From the predicted labels, we evaluate the $F1$ score (harmonic mean of precision and recall) (Murphy 2022) with the ground truth.

4. Numerical experiments

We evaluate the performance of the proposed model using disruption data from several discharges from the DIII-D tokamak taken from the 2016 experimental campaign. These disruptions were already included in previously published papers on data-driven applications in fusion (Montes *et al.* 2021).

We cluster the available data sets depending on the similarity of the conditions and on the occurring instability. Here, we use the model to augment five signals of the training data set (referred to as β_n, the normalized β given by $\beta_n = \beta a B_T / I_p$, where β is the ratio of plasma pressure to magnetic pressure, B_T is the toroidal magnetic field, a the minor radius and I_p the plasma current; normalized internal inductance li; plasma elongation κ; safety factor q_{95}; Greenwald fraction n/n_G) for different disruptions: (i) disruptions due to locked modes (LMs) in high β, low torque plasmas with $n = 1$ resonant magnetic perturbations (RMPs) applied (shots 166463, 166464, 166465, 166466, 166468, 166469), (ii) disruptions

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

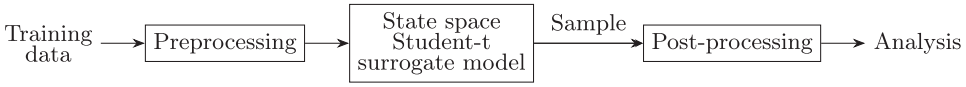


FIGURE 2. Data processing flow.

due to LMs during an RMP edge localized mode (ELM) suppression experiment applied to an ITER-like plasma shape (shots 166452, 166454, 166457, 166460) and (iii) density accumulation events during detachment studies of helium plasmas (shots 166933, 166934, 166937).

For each disruption class, the model is trained on these few available training samples. The choice of signals is influenced by the use case of augmenting the training database for a neural network for disruption prediction, but in general, the method is extendable to any number and any kind of signals.

Following the flow shown in figure 2, preprocessing is performed on the training data. As we are primarily interested in the behaviour close to a disruption, we align the samples according to their end time and only consider the stable flat-top phase. Additionally, all data are rescaled via min–max scaling to a range of $[-0.5, 0.5]$. This stabilizes the optimization of the hyperparameters in the Student t filter algorithm, as the input to the optimizer is of order 1. Missing data points are interpolated linearly. All discharges are sampled every 25 ms. Then, we set up the state space Student t surrogate model. In all experiments, a Matérn 3/2 kernel as in (2.6*a–d*) is used. We train the surrogate model by optimizing its hyperparameters by minimizing the negative log likelihood using the Scipy implementation of L-BFGS-B (Virtanen *et al.* 2020), and resulting values for all experiments can be found in Appendix B, table 4. Each signal has its own set of hyperparameters in order to be able to handle the dynamics that happen on different time scales. Subsequently, we apply the Student t filter and smoother (Algorithms 1 and 2) with optimized hyperparameters to our data. From the estimated distribution, we draw 1000 samples from the posterior using the Student t sampler and perform the colouring transformations in the post-processing. Finally, after rescaling the samples to the original data range, we evaluate the generated data sets by using the defined metrics. In general, the generation of the time series samples is of $O(N)$, but some of the metrics used to evaluate the generate data are not. Therefore, we limited the number of samples in the given analysis to 1000.

5. Results and analysis

For each disruption class, we draw 1000 samples from the posterior estimated by the trained model and compare four available post-processing methods: (I) uncorrelated model (here, no post-processing is performed), (II) colouring transformation with the empirical covariance matrix, (III) colouring transformation with the empirical cross-covariance matrix to account for lagged correlations and (IV) colouring transformation with the sampled covariance matrix. The results for test cases (i) and (ii) are presented in Appendix in C.1 and C.2.

In figure 3, a visual comparison is given between training data and generated data for the colouring transformation with empirical cross-covariance matrix, together with the estimated mean and 95 % confidence intervals for the disruption data from DIII-D for test case (i). The model is able to capture the general trend given by the training data and can also reproduce outliers. In general, the generated data fit the distribution of the training data.

We continue with a thorough statistical analysis, which allows a ranking of the different post-processing methods following the metrics outlined in § 3. The results are given in

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

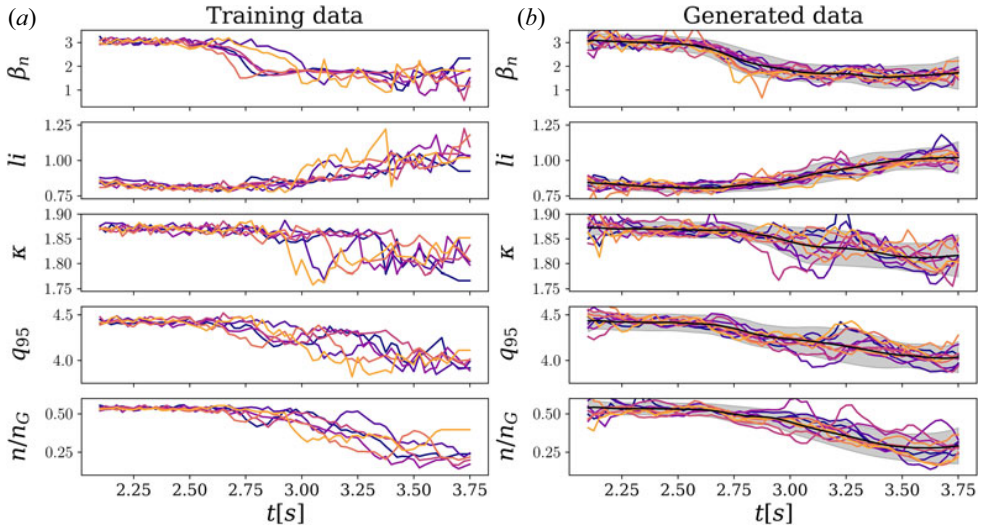


FIGURE 3. (*a*) Training data and (*b*) 10 generated data sets from the state space Student *t* surrogate model together with the estimated mean (black solid line) and 95 % confidence (grey shaded region) for test case (i). Different colours correspond to different shots of training data and different samples of the generated data, respectively.

Metric	Uncorrelated	Emp. cov	Emp. crosscov	Sample cov
W_1	0.035 ± 0.013	0.035 ± 0.011	0.038 ± 0.012	$\mathbf{0.036 \pm 0.01}$
MMD p-value	0.68 ± 0.35	0.82 ± 0.18	$\mathbf{0.92 \pm 0.06}$	0.85 ± 0.13
MSE ρ_{rs}	0.019 ± 0.009	0.018 ± 0.009	$\mathbf{0.0011 \pm 0.0004}$	0.017 ± 0.009
W_{emb}	$\mathbf{0.0592 \pm 0.0006}$	0.0682 ± 0.0006	0.0766 ± 0.0007	0.0682 ± 0.006
MSE PSD $[10^{-6}]$	5 ± 4	4 ± 3	$\mathbf{3 \pm 3}$	4 ± 3
DTW	0.8 ± 0.5	0.8 ± 0.4	$\mathbf{0.7 \pm 0.3}$	0.85 ± 0.4
MSE mfPCA	0.137	0.015	**0.011**	0.022

TABLE 1. Post-processing method comparison for test case (i). Mean and standard deviation over five dimensions and $N = 1000$ samples generated from the trained model for statistical metrics described in § 3. Best values are highlighted in bold.

table 1 for test case (i). Other experiments give similar results, as indicated in Appendix in C.1 (table 5), and C.2 (table 7) for test cases (ii) and (iii), respectively.

To put the calculated metrics into context, we identify nearby non-disruptive shots coming from the same specific campaign with similar operating conditions for test case (ii). Then, we evaluate the Wasserstein distance between nearby non-disruptive and disruptive discharges to compare the obtained Wasserstein distances for the generated data for this disruption class. For test case (ii), we identify five nearby non-disruptive shots 166433, 166434, 166442, 166444, 166455 and found $W_1 = 0.31 \pm 0.12$ between non-disruptive and disruptive discharges. Additionally, the 2-D kernel PCA embedding of nearby non-disruptive and disruptive discharges evaluated by the estimation of the 2-D sliced Wasserstein distance is estimated. We observe $W_{\text{emb}} = 0.74 \pm 0.01$ for test case (ii).

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

Test case	Uncorrelated	Emp. cov	Emp. crosscov	Sample cov
(i) stable	0.03 ± 0.01	0.03 ± 0.01	0.04 ± 0.01	0.03 ± 0.01
(i) unstable	0.04 ± 0.01	0.04 ± 0.01	0.04 ± 0.01	0.04 ± 0.01
(ii) stable	0.02 ± 0.02	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.01
(ii) unstable	0.07 ± 0.02	0.07 ± 0.02	0.08 ± 0.02	0.08 ± 0.02
(iii) stable	0.08 ± 0.08	0.03 ± 0.02	0.02 ± 0.01	0.03 ± 0.02
(iii) unstable	0.07 ± 0.08	0.03 ± 0.03	0.02 ± 0.02	0.04 ± 0.03

TABLE 2. Post-processing method comparison for disruption data from DIII-D. Mean and standard deviation of the Wasserstein metric between training and generated data for stable and unstable phases of the disruptive discharges. The Wasserstein metric is averaged over five dimensions and $N = 1000$ samples generated from the trained model.

The achieved Wasserstein distance between training and generated data for this disruption class is significantly smaller in all post-processing methods, as given in table 5. The same holds for the Wasserstein distance of the 2-D kernel PCA embedding. This is promising, as it implies that the augmented data are much more similar to disruptive discharges within their proper class than to non-disruptive discharges from the same campaign in these measures.

For test cases (i) and (iii), non-disruptive discharges from those specific campaigns are not available. Therefore, we investigate the distributions in stable and unstable phases of the training and generated disruptive discharges in more detail. Using the average time stamp of the manually labelled training data, this information about the stable and unstable phase was propagated to label the generated data. Then we calculate the Wasserstein distance averaged over all features between training and generated data for both phases separately. The obtained results for all test cases are given in table 2. For comparison, we also estimate the Wasserstein distances between stable and unstable phases and found $W_1 = 0.36 \pm 0.07$ for test case (i), $W_1 = 0.37 \pm 0.08$ for test case (ii) and $W_1 = 0.24 \pm 0.1$ for test case (iii). The obtained distances between training and generated data within the different phases lie sufficiently below the distances between stable and unstable parts of the discharges.

The superiority of the post-processing with the empirical cross-covariance is apparent in figure 4, where the auto- (on the diagonal) and cross-covariance for all estimated signals are shown. As we are inscribing the empirical cross-covariance into the uncorrelated generated data from the model, the cross-covariance fits exactly, and the cross-covariances lie on top of each other. When using either the empirical covariance or the sample covariance, only the cross-covariance at lag 0 matches the cross-covariance of the training data. Both post-processing methods give on average the same cross-covariance for 1000 generated samples. Additionally, the difference in covariance at lag 0 is shown in figure 5.

Figure 6 displays the kernel density of the 2-D kernel PCA embedding of the generated data in the eigenspace of the training data. All four methods generate data that lie on the same submanifold as the training data. However, when cross-covariances are included, the shape of the training data is better reproduced. In test case (i) shown in figure 6, one of the three extrema is not reproduced by the generated data. By evaluating the embedding for different combinations of input signals, a likely explanation is that β_n causes this extremum. The reason why the generated data are not able to reproduce this extremum in the eigenspace is due to the multi-modality of the distribution around the drop in β_n in the range 2.75–3.00 s. This is also one limitation of the presented model as it is not

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

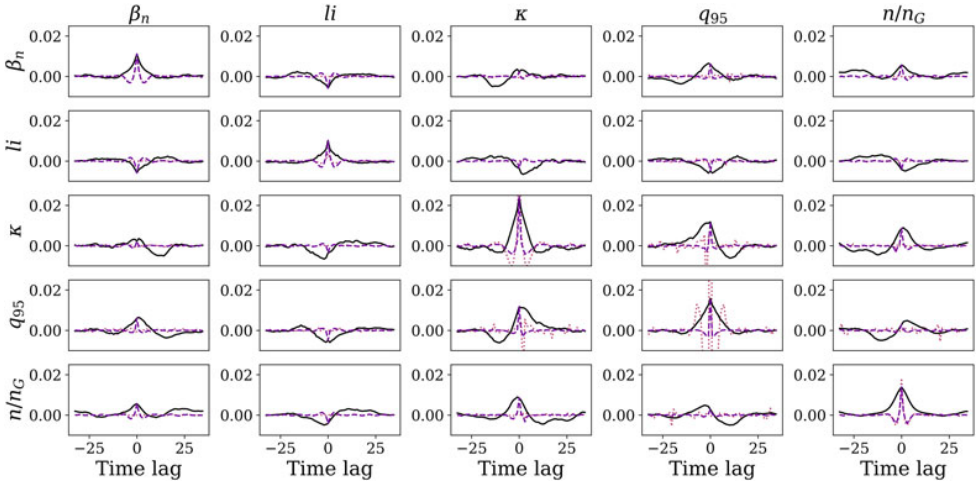


FIGURE 4. Comparison of the cross-covariance in the training and generated data with cross-covariance (solid lines on top of each other, numerical error of order 10^{-16}), covariance or sampled covariance post-processing (dashed lines) and uncorrelated model (dotted line) for test case (i).

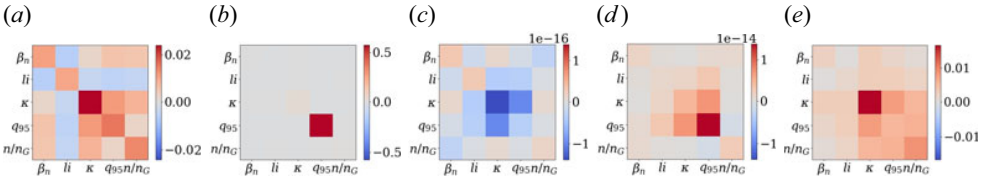


FIGURE 5. Comparison of the covariance of training data (*a*) and the difference from the generated data (*b*) with uncorrelated model, (*c*) empirical covariance, (*d*) cross-covariance and (*e*) sampled covariance post-processing for test case (i). Note the different scaling in the colour scale.

able to represent multi-modality of a cluster correctly. One possibility is to further refine the considered clusters to augment the data base (in the extreme case, down to one single discharge). In general, the number of available training data samples is very limited, as we are working with manually labelled disruptive data from DIII-D. Therefore, the results here only give an idea of whether the features apparent in the training data are also apparent in the generated data.

Besides the Wasserstein distance, DTW is difficult to interpret without context. Again, we calculate the metric between nearby non-disruptive and disruptive discharges for test case (ii) and obtain DTW $= 2.9 \pm 1.6$. The large error is due to averaging over all signals. Overall, the distances between the generated and training data for this disruption class lie below the distance between nearby non-disruptive and disruptive discharges for this test case. In test cases (i) and (iii), where non-disruptive data from the same campaigns are not available, DTW distances between generated and training data with included correlations are of the same order as in test case (ii).

The training data were also reconstructed using the multivariate functional PCA with 5 components. We observe the following reconstruction mean squared errors for test case (i) 0.006, (ii) 0.003 and (iii) 0.008. We use the first five eigenfunctions of the training data

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

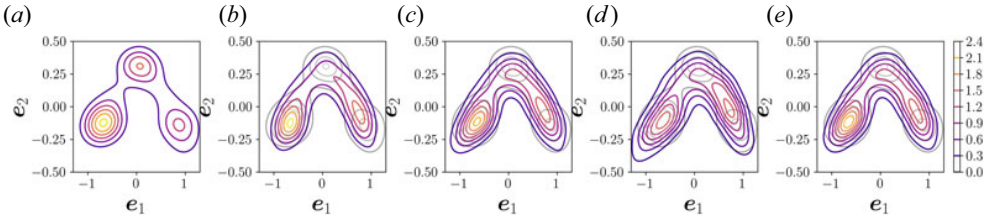


FIGURE 6. Kernel density estimation of the 2-D kernel PCA embedding of the (*a*) training data and generated data via (*b*) uncorrelated model, (*c*) empirical covariance, (*d*) cross-covariance and (*e*) sampled covariance post-processing for test case (i). The embedded training data are shown in grey in all plots. The colour scale representing the density is the same in all plots.

Train	Test	Training	Uncorrelated	Emp. cov	Emp. crosscov	Sample cov
original	generated	0.75	0.74	0.75	0.74	0.78
generated	original	0.88	0.86	0.90	0.90	0.89
mix	mix	0.89	0.89	0.89	0.89	0.89

TABLE 3. The $F1$ score for DTW SOM clustering of different post-processing methods for test case (i).

as a basis to project the generated data of each test case. The reconstruction error of the generated data with included correlations in the post-processing is still of the same order.

Finally, we use SOMs for time series clustering to evaluate if the label prediction works similarly well for the generated data. Here, we only use three classes, as the training data look quite similar for different signals. The results for three different experiments are given in table 3. Between the four post-processing methods, no significant difference is evident. The clustering algorithm performs as well on all methods as on the original training data.

6. Conclusion and outlook

We applied Student *t* process regression in a state space formulation to introduce robust data augmentation for multivariate time series. The state space formulation reduces the computational complexity and is thus suitable for high-resolution time series. We used the model to learn the distribution of time series coming from a given disruption class. From the estimated posterior, time series were generated to augment the training database. To evaluate if the original and generated data share key statistical properties, multiple statistical analyses and classic machine learning clustering algorithms have been carried out. We found that, within the scope of the used metrics, the generated time series resemble the training data to a sufficient extent. An important limitation of the method is multi-modality in the training data set which a Student *t* process cannot reproduce. In this case, the training data sets can be further split.

When the method is applied to augment the training database for the neural network disruption predictor, a thorough analysis of the existing (labelled) training database is necessary to decide which disruption classes are not available in sufficient quantity. For each of those classes, we will train the surrogate model and then be able to generate data to balance the data set. Subsequently, the performance of the neural network trained with the augmented training database will be evaluated. Due to the broad range of

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

evaluation metrics, we are optimistic that the generated data will improve and robustify the performance.

Another perspective regards disruption prediction of future devices, where little data will be available to train machine learning-based approaches. In this case, the surrogate model could be used and updated, as more data are being collected and can therefore update machine learning-driven models.

To improve the proposed method, the integration of correlations and cross-correlations on the level of a multivariate surrogate model instead of the colouring in post-processing will be investigated in future work (Boyle & Frean 2004; Vandenberg-Rodes & Shahbaba 2015). Another possible extension of the current method could also take spatial information of profiles into account (Wilkinson *et al.* 2020).

However, the approach developed here is sufficiently generic to be used for data augmentation in a broad range of applications, e.g. time series in climate research.

Acknowledgements

Editor William Dorland thanks the referees for their advice in evaluating this article.

Funding

The present contribution is supported by the Helmholtz Association of German Research Centers under the joint research school HIDSS-0006 'Munich School for Data Science – MUDS' (K.R.) and the MIT-Germany Lockheed Martin Seed Fund (K.R., C.R., A.M., R.G., U.v.T.). This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 – EUROfusion). Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them (C.A.). This work was supported by the Federal Ministry of Education and Research (BMBF) of Germany by Grant No. 01IS18036A (D.R., B.B.). This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences, using the DIII-D National Fusion Facility, a DOE Office of Science user facility, under Award(s) DE- SC0014264, and DE-FC02-04ER54698 (C.R., A.M., R.G.).

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process or service by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation or favouring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Declaration of interests

The authors report no conflict of interest.

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

Appendix A. Algorithm

Algorithm 1 Multivariate Student-t filter (Solin & Särkkä 2015)

Init:

$$\hat{\boldsymbol{y}}_{0|0} = \boldsymbol{y}_0, \quad \boldsymbol{P}_{0|0} = \boldsymbol{P}_0, \quad \boldsymbol{v}_0 = \boldsymbol{v}, \quad \gamma_0 = \boldsymbol{I}_D \tag{A 1}$$

for $t = 1, 2, ..., T$ **do**

Filter prediction:

$$\hat{\boldsymbol{y}}_{t|t-1} = \boldsymbol{A}_{t-1}\hat{\boldsymbol{y}}_{t-1} \tag{A 2}$$

$$\boldsymbol{P}_{t|t-1} = \boldsymbol{A}_{t-1}\boldsymbol{P}_{t-1}\boldsymbol{A}_{t-1}^{\top} + \gamma_{t-1}\boldsymbol{Q}_{t-1}, \tag{A 3}$$

where $\boldsymbol{A}_t = \exp(\boldsymbol{F}\Delta t)$ and $\boldsymbol{Q}_t = \boldsymbol{P}_0 - \boldsymbol{A}_t\boldsymbol{P}_0\boldsymbol{A}_t^{\top}$.

Filter update (if measurement $\boldsymbol{y}_t$ with mean $\bar{\boldsymbol{y}}_t$ is available):

$$\boldsymbol{v}_t = \bar{\boldsymbol{y}}_t - \boldsymbol{H}_t\hat{\boldsymbol{y}}_t \tag{A 4}$$

$$\boldsymbol{S}_t = \boldsymbol{H}_t\boldsymbol{P}_{t|t-1}\boldsymbol{H}_t^{\top} + \boldsymbol{R} \tag{A 5}$$

$$\gamma_t = \frac{\gamma_{t-1}}{\nu_t - 2}(\nu_{t-1} - 2 + \boldsymbol{v}_t\boldsymbol{S}_t^{-1}\boldsymbol{v}_t) \tag{A 6}$$

$$\boldsymbol{K}_t = \boldsymbol{P}_{t|t-1}\boldsymbol{H}_t^{\top}\boldsymbol{S}_t^{-1} \tag{A 7}$$

$$\hat{\boldsymbol{y}}_{t|t} = \hat{\boldsymbol{y}}_{t|t-1} + \boldsymbol{K}_t\boldsymbol{v}_t \tag{A 8}$$

$$\boldsymbol{P}_{t|t} = \frac{\gamma_t}{\gamma_{t-1}}(\boldsymbol{P}_{t|t-1} - \boldsymbol{K}_t\boldsymbol{S}_t\boldsymbol{K}_t^{\top}) \tag{A 9}$$

end for

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

Algorithm 2 Multivariate Student-t smoother (Solin & Särkkä 2015)

Init:

$$\hat{\boldsymbol{y}}_T = \hat{\boldsymbol{y}}_{T|t}, \quad \boldsymbol{P}_T = \boldsymbol{P}_{T|t} \tag{A 10}$$

for $t = T-1, T-2, ..., 1$ **do**

Smoother prediction:

$$\hat{\boldsymbol{y}}_{t+1|t} = \boldsymbol{A}_t \hat{\boldsymbol{y}}_{t|t} \tag{A 11}$$

$$\boldsymbol{P}_{t+1|t} = \boldsymbol{A}_t \boldsymbol{P}_{t|t} \boldsymbol{A}_t^\top + \gamma_t \boldsymbol{Q}_t \tag{A 12}$$

Smoother update:

$$\boldsymbol{G}_t = \boldsymbol{P}_{t|t} \boldsymbol{A}_t^\top \boldsymbol{P}_{t+1|t}^{-1} \tag{A 13}$$

$$\hat{\boldsymbol{y}}_{t|T} = \hat{\boldsymbol{y}}_{t|t} + \boldsymbol{G}_t (\hat{\boldsymbol{y}}_{t+1|T} - \hat{\boldsymbol{y}}_{t+1|t}) \tag{A 14}$$

$$\boldsymbol{P}_{t|T} = \frac{\gamma_T}{\gamma_t} (\boldsymbol{P}_{t|t} - \boldsymbol{G}_t \boldsymbol{P}_{t+1|T} \boldsymbol{G}_t^\top) + \boldsymbol{G}_t \boldsymbol{P}_{t+1|T} \boldsymbol{G}_t^\top \tag{A 15}$$

end for

Appendix B. Hyperparameters

For the different test cases, we used the hyperparameters given in table 4.

Test case	hyp	β_n	li	κ	q_{95}	n/n_G
(i)	ν	2.19	2.58	2.15	2.21	2.1
	σ_n^2	0.024	0.01	0.056	0.029	0.02
	σ^2	1.74	1.76	1.96	1.60	1.60
	l	19.6	28.3	20.3	20.1	15.7
(ii)	ν	3.4	2.57	2.36	2.49	2.7
	σ_n^2	0.023	0.032	0.036	0.033	0.022
	σ^2	1.65	0.53	1.36	1.87	1.91
	l	17.7	19.8	9.63	19.1	16.8
(iii)	ν	2.14	2.12	2.01	2.71	2.55
	σ_n^2	0.163	0.044	0.493	0.016	0.011
	σ^2	0.62	1.73	1.24	1.21	0.58
	l	17.6	11.5	4.5	12.8	10.0

TABLE 4. Optimized hyperparameters for the state space Student t surrogate model for all test cases.

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

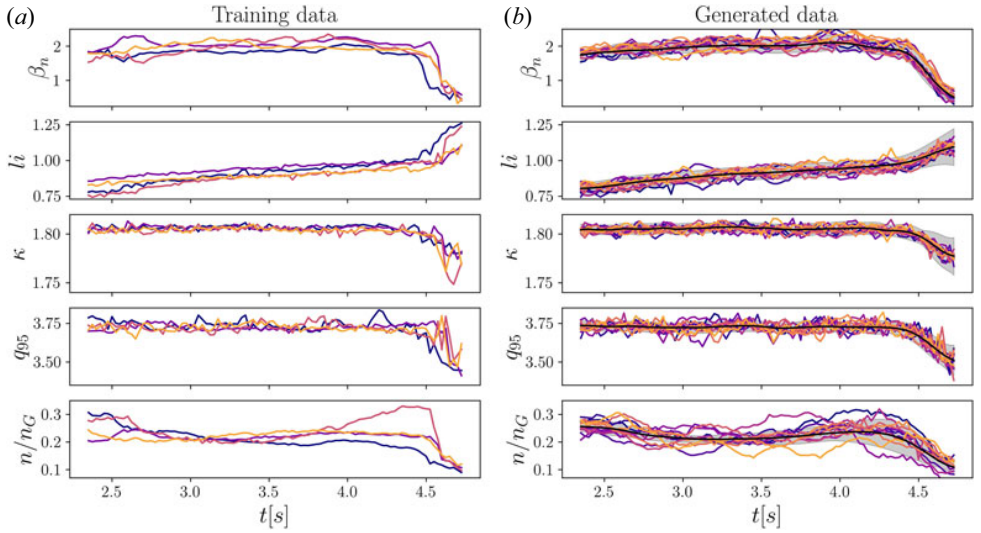


FIGURE 7. (*a*) Training data and (*b*) 10 generated data sets from the state space Student *t* surrogate model together with the estimated mean (black solid line) and 95 % confidence (grey shaded region) for test case (ii). Different colours correspond to different shots of training data and different samples of the generated data, respectively.

Metric	Uncorrelated	Emp. cov	Emp. crosscov	Sample cov
W_1	0.027 ± 0.013	0.020 ± 0.006	0.022 ± 0.007	$\mathbf{0.020 \pm 0.006}$
MMD p-value	0.617 ± 0.335	0.869 ± 0.153	$\mathbf{0.885 \pm 0.107}$	0.876 ± 0.15
MSE ρ_{rs}	0.013 ± 0.017	0.013 ± 0.017	$\mathbf{0.005 \pm 0.005}$	0.014 ± 0.017
W_{emb}	0.0458 ± 0.0003	0.0496 ± 0.0004	$\mathbf{0.0466 \pm 0.0004}$	0.0539 ± 0.0004
MSE PSD $[10^{-6}]$	7 ± 9	3 ± 4	$\mathbf{1 \pm 2}$	3 ± 4
DTW	0.86 ± 0.37	0.78 ± 0.32	$\mathbf{0.65 \pm 0.31}$	0.83 ± 0.39
MSE mfPCA	0.011	0.009	**0.007**	0.011

TABLE 5. Post-processing method comparison for test case (ii). Mean and standard deviation over five dimensions and $N = 1000$ samples generated from the trained model for statistical metrics described in § 3. Best values are highlighted in bold.

Appendix C. Results for other test cases

In the following sections, the results for test cases (ii) and (iii) are presented.

C.1. *Test case (ii): disruption due to MHD instability during RMP ELM control*

A visual comparison of the training and the generated data for test case (ii) is shown in figure 7. Here, the disruption occurs due to magnetohydrodynamic (MHD) instability induced by RMPs applied to control ELMs (shots 166452, 166454, 166457, 166460). The results of the statistical analysis are given in table 5 and are of the same order as for test case (i). Figures 8 and 9 show the cross-covariance and the covariance of the training and generated data. Figure 10 displays the kernel density of 2-D PCA embedding

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

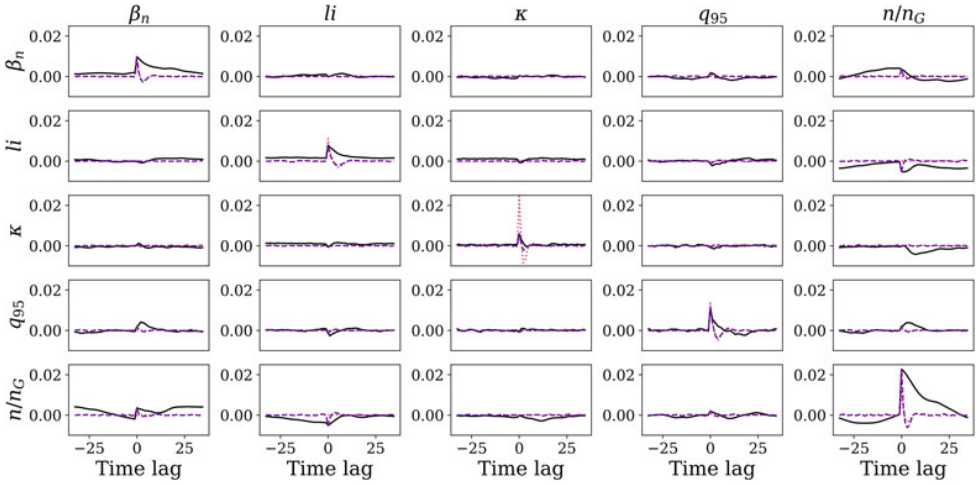


FIGURE 8. Comparison of cross-covariance of training data and generated data with cross-covariance (solid lines on top of each other, numerical error of order 10^{-16}), covariance or sampled covariance (dashed lines) post-processing and uncorrelated model (dotted line) for test case (ii).

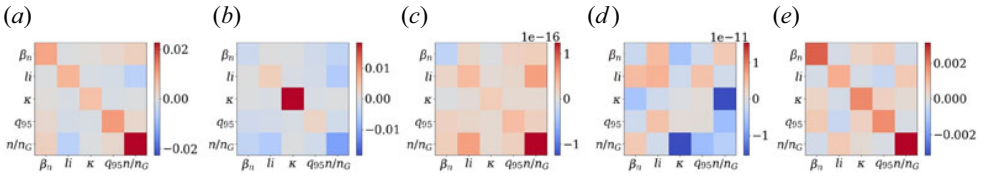


FIGURE 9. Comparison of covariance of training data (*a*) and difference of generated data (*b*) with uncorrelated model, (*c*) empirical covariance, (*d*) cross-covariance and (*e*) sampled covariance post-processing for test case (ii). Note the different scaling in the colour scale.

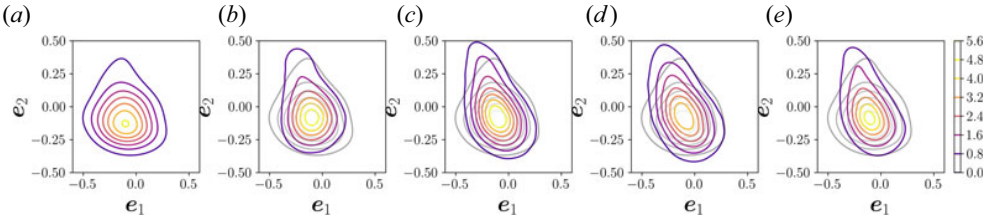


FIGURE 10. Kernel density estimation of the 2-D kernel PCA embedding of the (*a*) training data and generated data via (*b*) uncorrelated model, (*c*) empirical covariance, (*d*) cross-covariance and (*e*) sampled covariance post-processing for test case (ii). The embedded training data are shown in grey in all plots. The colour scale representing the density is the same in all plots.

of the generated data. Again, the results show that the generated data lives on the same submanifold for all four post-processing methods. In table 6, the $F1$ score for DTW SOM clustering is given.

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

Train	Test	Training	Uncorrelated	Emp. cov	Emp. crosscov	Sample cov
original	generated	1.0	1.0	1.0	0.94	1.0
generated	original	1.0	0.91	1.0	1.0	1.0
mix	mix	1.0	1.0	1.0	0.96	1.0

TABLE 6. The $F1$ score for DTW SOM clustering of different post-processing methods for test case (ii).

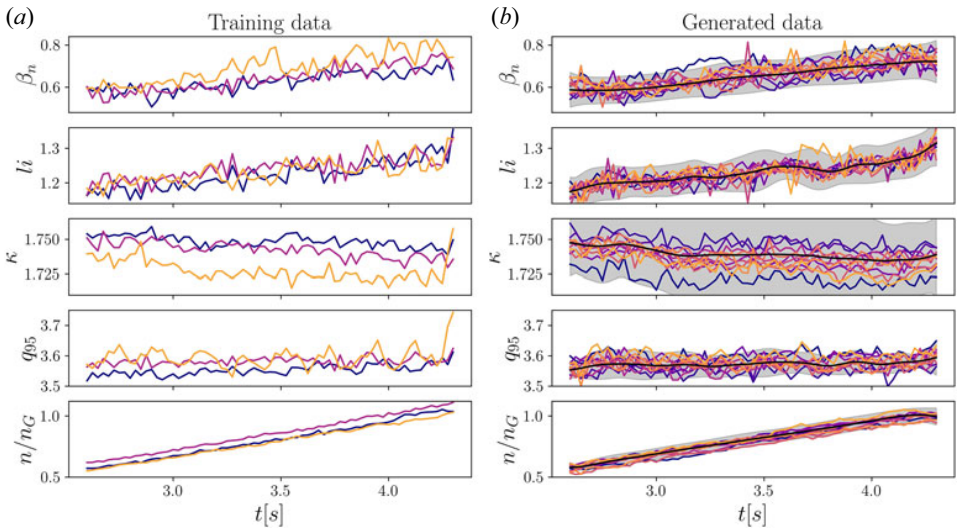


FIGURE 11. (*a*) Training data and (*b*) 10 generated data sets from the state space Student t surrogate model together with the estimated mean (black solid line) and 95 % confidence (grey shaded region) for test case (iii). Different colours correspond to different shots of training data and different samples of the generated data, respectively.

Metric	Uncorrelated	Emp. cov	Emp. crosscov	Sample cov
W_1	0.071 ± 0.088	0.030 ± 0.026	$\mathbf{0.025 \pm 0.019}$	0.03 ± 0.028
MMD p-value	0.43 ± 0.32	0.86 ± 0.17	0.84 ± 0.09	$\mathbf{0.87 \pm 0.13}$
MSE ρ_{rs}	0.0083 ± 0.0075	0.0076 ± 0.007	$\mathbf{0.0019 \pm 0.0019}$	0.0070 ± 0.007
W_{emb}	0.1808 ± 0.0034	0.0621 ± 0.0011	$\mathbf{0.0517 \pm 0.0008}$	0.0656 ± 0.0012
MSE PSD $[10^{-6}]$	240 ± 33	11 ± 19	$\mathbf{0.8 \pm 0.5}$	79 ± 13
DTW	1.7 ± 2.2	0.9 ± 0.6	$\mathbf{0.8 \pm 0.6}$	0.9 ± 0.6
MSE mfPCA	0.138	0.021	**0.010**	0.025

TABLE 7. Post-processing method comparison for test case (iii). Mean and standard deviation over five dimensions and $N = 1000$ samples generated from the trained model for statistical metrics described in § 3. Best values are highlighted in bold.

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

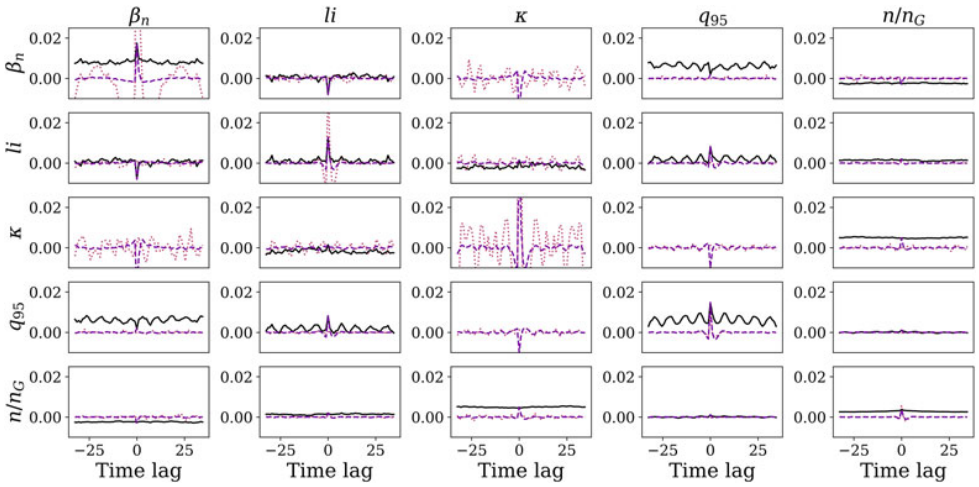


FIGURE 12. Comparison of cross-covariance of training data and generated data with cross-covariance (solid lines on top of each other, numerical error of order 10^{-16}), covariance or sampled covariance (dashed lines) post-processing and uncorrelated model (dotted line) for test case (iii).

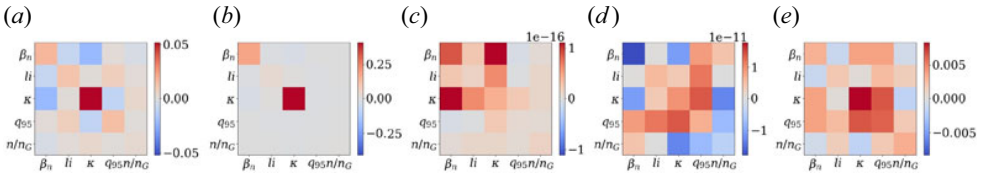


FIGURE 13. Comparison of covariance of training data (*a*) and difference of generated data (*b*) with uncorrelated model, (*c*) empirical covariance, (*d*) cross-covariance and (*e*) sampled covariance post-processing for test case (iii). Note the different scaling in the colour scale.

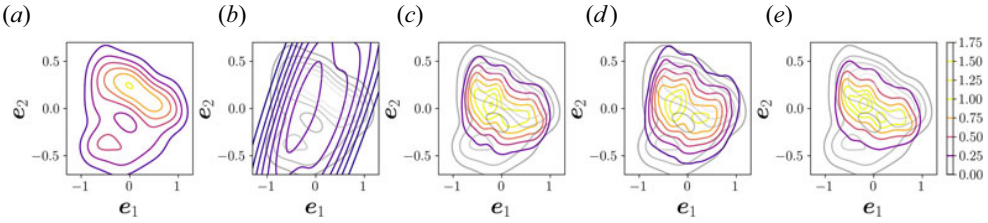


FIGURE 14. Kernel density estimation of the 2-D kernel PCA embedding of the (*a*) training data and generated data via (*b*) uncorrelated model, (*c*) empirical covariance, (*d*) cross-covariance and (*e*) sampled covariance post-processing for test case (iii). The embedded training data are shown in grey in all plots. The colour scale representing the density is the same in all plots.

C.2. *Test case (iii): density accumulation*

For the third test case with a disruption occurring due to density accumulation (shots 166933, 166934, 166937), the visual comparison is given in figure 11 followed by the results of the statistical analysis in table 7. The cross-covariance and covariance are

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

Train	Test	Training	Uncorrelated	Emp. cov	Emp. crosscov	Sample cov
original	generated	0.81	0.98	1.0	0.85	0.99
generated	original	0.99	0.92	0.93	0.93	0.93
mix	mix	0.96	0.96	0.96	0.94	0.97

TABLE 8. The $F1$ score for DTW SOM clustering of different post-processing methods for test case (iii).

displayed in figures 12 and 13, respectively. The embedding is shown in figure 14. Here, the skew of the embedding caused by the broad distribution of κ is not perfectly reproduced by the generated data. However, the results should be regarded with caution as only 3 training data samples are available in this test case. This presents also a limit to this metric. However, when looking at samples of the generated data shown in figure 11, this broad range present in the training data is still well reproduced by the generated data. In this test case, the uncorrelated model performs worst as correlations are not reproduced. The results for generated data with included correlations are again of the same order of magnitude as for test cases (i) and (ii). The results obtained for the $F1$ score for DTW SOM clustering are given in table 8.

REFERENCES

AGGARWAL, C.C. 2018 *Neural Networks and Deep Learning*. Springer.

AYMERICH, E., SIAS, G., PISANO, F., CANNAS, B., CARCANGIU, S., SOZZI, C., STUART, C., CARVALHO, P.J., FANNI, A. & JET CONTRIBUTORS 2022 Disruption prediction at JET through deep convolutional neural networks using spatiotemporal information from plasma profiles. *Nucl. Fusion* **62** (6), 066005.

BERKERY, J.W., SABBAGH, S.A., BELL, R.E., GERHARDT, S.P. & LEBLANC, B.P. 2017 A reduced resistive wall mode kinetic stability model for disruption forecasting. *Phys. Plasmas* **24** (5), 056103.

BERNDT, D.J. & CLIFFORD, J. 1994 Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, *AAAIWS'94*, pp. 359–370. AAAI.

BONNEEL, N., RABIN, J., PEYRÉ, G. & PFISTER, H. 2015 Sliced and radon wasserstein barycenters of measures. *J. Math. Imag. Vis.* **1** (51), 22–45.

BOYLE, P. & FREAN, M. 2004 Dependent Gaussian processes. In *Advances in Neural Information Processing Systems* (ed. L. Saul, Y. Weiss & L. Bottou), vol. 17. MIT.

CLEVELAND, R.B., CLEVELAND, W.S., MCRAE, J.E. & TERPENNING, I. 1990 STL: a seasonal-trend decomposition procedure based on loess (with discussion). *J. Off. Stat.* **6**, 3–73.

DURBIN, J. & KOOPMAN, S.J. 2002 A simple and efficient simulation smoother for state space time series analysis. *Biometrika* **89** (3), 603–615.

FLAMARY, R., COURTY, N., GRAMFORT, A., ALAYA, M.Z., BOISBUNON, A., CHAMBON, S., CHAPEL, L., CORENFLOS, A., FATRAS, K.F., NEMO, G., *et al.* 2021 POT: Python optimal transport. *J. Mach. Learn. Res.* **22** (78), 1–8.

GRETTON, A., BORGWARDT, K.M., RASCH, M.J., SCHÖLKOPF, B. & SMOLA, A. 2012 A kernel two-sample test. *J. Mach. Learn. Res.* **13**, 723–773.

HAPP, C. & GREVEN, S. 2018 Multivariate functional principal component analysis for data observed on different (dimensional) domains. *J. Am. Stat. Assoc.* **113** (522), 649–659.

IWANA, B.K. & UCHIDA, S. 2021 An empirical survey of data augmentation for time series classification with neural networks. *PLoS ONE* **16** (7), 1–32.

KANG, Y., HYNDMAN, R.J. & LI, F. 2020 Gratis: Generating time series with diverse and controllable characteristics. *Stat. Anal. Data Min.* **13** (4), 354–376.

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

KATES-HARBECK, J., SVYATKOVSKIY, A. & TANG, W. 2019 Predicting disruptive instabilities in controlled fusion plasmas through deep learning. *Nature* **568** (7753), 526–531.

KESSY, A., LEWIN, A. & STRIMMER, K. 2018 Optimal whitening and decorrelation. *Am. Stat.* **72** (4), 309–314.

MONTES, K.J., REA, C., TINGUELY, R.A., SWEENEY, R., ZHU, J. & GRANETZ, R.S. 2021 A semi-supervised machine learning detector for physics events in tokamak discharges. *Nucl. Fusion* **61** (2), 026022.

MURPHY, K.P. 2022 *Probabilistic Machine Learning: An Introduction*. MIT.

NEAL, R.M. 1997 Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification. (Technical report, University of Toronto, Department of Statistics). University of Toronto.

PARK, K.I. 2017 *Fundamentals of Probability and Stochastic Processes with Applications to Communications*. Springer.

PAU, A., FANNI, A., CARCANGIU, S., CANNAS, B., SIAS, G., MURARI, A. & RIMINI, F. 2019 A machine learning approach based on generative topographic mapping for disruption prevention and avoidance at JET. *Nucl. Fusion* **59** (10), 106017.

REA, C. & GRANETZ, R.S. 2018 Exploratory machine learning studies for disruption prediction using large databases on DIII-D. *Fusion Sci. Technol.* **74** (1–2), 89–100.

REA, C., MONTES, K.J., ERICKSON, K.G., GRANETZ, R.S. & TINGUELY, R.A. 2019 A real-time machine learning-based disruption predictor in DIII-D. *Nucl. Fusion* **59** (9), 096016.

REA, C., MONTES, K.J., PAU, A., GRANETZ, R.S. & SAUTER, O. 2020 Progress toward interpretable machine learning–based disruption predictors across tokamaks. *Fusion Sci. Technol.* **76** (8), 912–924.

ROTH, M., ARDESHIRI, T., ÖZKAN, E. & GUSTAFSSON, F. 2017 Robust Bayesian filtering and smoothing using student's *t* distribution. CoRR abs/1703.02428, arXiv:1703.02428.

SÄRKKÄ, S. 2013 *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge University Press.

SÄRKKÄ, S. & SOLIN, A. 2019 *Applied Stochastic Differential Equations*. Institute of Mathematical Statistics Textbooks. Cambridge University Press.

SCHÖLKOPF, B., SMOLA, A. & MÜLLER, K.-R. 1998 Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **10** (5), 1299–1319.

SHAH, A., WILSON, A.G. & GHAHRAMANI, Z. 2014 Student-*t* processes as alternatives to Gaussian processes. *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics* **PMLR: 33**, 877–885.

SHORTEN, C. & KHOSHGOFTAAR, T.M. 2019 A survey on image data augmentation for deep learning. *J. Big Data* **6**, 1–48.

SOLIN, A. & SÄRKKÄ S. 2015 State space methods for efficient inference in student- process regression. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics* ***PMLR:*** **38**, 885–893.

VANDENBERG-RODES, A. & SHAHBABA, B. 2015 Dependent matérn processes for multivariate time series. https://arxiv.org/abs/1502.03466.

VANHATALO, J., JYLÄNKI, P. & VEHTARI, A. 2009 Gaussian process regression with student-*t* likelihood. In *Advances in Neural Information Processing Systems* (ed. Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams & A. Culotta), vol. 22. Curran Associates, Inc.

VETTIGLI, G. 2018 Minisom: minimalistic and numpy-based implementation of the self organizing map. https://github.com/JustGlowing/minisom/.

VILLANI, C. 2008 *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer.

VIRTANEN, P., GOMMERS, R., OLIPHANT, T.E., HABERLAND, M., REDDY, T., COURNAPEAU, D., BUROVSKI, E., PETERSON, P., WECKESSER, W., BRIGHT, J., *et al.* 2020 SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Meth.* **17**, 261–272.

WEN, Q., GAO, J., SONG, X., SUN, L., XU, H. & ZHU, S. 2019 RobustSTL: a robust seasonal-trend decomposition algorithm for long time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5409–5416.

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

WEN, Q., SUN, L., YANG, F., SONG, X., GAO, J., WANG, X. & XU, H. 2021 Time series data augmentation for deep learning: a survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization.

WILKINSON, W.J., CHANG, P.E., ANDERSEN, M.R. & SOLIN, A. 2020 State space expectation propagation: efficient inference schemes for temporal Gaussian processes. *Proceedings of the 37th International Conference on Machine Learning* **PMLR: 119**, 10270–10281.

WILLIAMS, C.K.I. & RASMUSSEN, C.E. 1996 Gaussian processes for regression. In *Advances in Neural Information Processing Systems*, vol. 8, pp. 514–520. MIT.

YOON, J., JARRETT, D. & VAN DER SCHAAR, M. 2019 Time-series generative adversarial networks. In *Advances in Neural Information Processing Systems* (ed. H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox & R. Garnett), vol. 32. Curran Associates, Inc.

https://doi.org/10.1017/S0022377822000769 Published online by Cambridge University Press

3.5 Dependent state space Student-*t* processes for imputation and data augmentation in plasma diagnostics

Main novelty:
We introduce a fully multivariate state space Student-t process model for imputing and augmenting time-series data in plasma diagnostics. Correlations between input signals are directly included in the model.

Contributing article:
Rath, K., Rügamer, D., Bischl, B., von Toussaint, U., and Albert, C. G. (2023). Dependent state space Student-t processes for imputation and data augmentation in plasma diagnostics. *Contributions to Plasma Physics*, 63(5-6):e202200175

Author contributions:
Katharina Rath developed the model, implemented the software, designed and performed the experiments and evaluation routines, and wrote the paper. David Rügamer, Christopher Albert, Bernd Bischl, and Udo von Toussaint added valuable input, suggested notable modifications, and reviewed the manuscript.

Received: 18 November 2022 | Revised: 22 March 2023 | Accepted: 13 April 2023 | Published on: 10 May 2023

DOI: 10.1002/ctpp.202200175

ORIGINAL ARTICLE

Contributions to Plasma Physics

Dependent state space Student-t processes for imputation and data augmentation in plasma diagnostics

Katharina Rath[1,2] | **David Rügamer**[2,3] | **Bernd Bischl**[2,4] | **Udo von Toussaint**[1] | **Christopher G. Albert**[5]

[1]Department Numerical Methods of Plasma Physics, Max-Planck-Institut für Plasmaphysik, Garching, Germany

[2]Department of Statistics, Ludwig-Maximilians-Universität München, München, Germany

[3]Department of Statistics, TU Dortmund University, Dortmund, Germany

[4]Munich Center for Machine Learning (MCML), München, Germany

[5]Fusion@OEAW, Institute of Theoretical and Computational Physics, Technische Universität Graz, Graz, Austria

Correspondence
Katharina Rath, Department Numerical Methods of Plasma Physics, Max-Planck-Institut für Plasmaphysik, Garching, Germany.
Email: katharina.rath@ipp.mpg.de

Funding information
EUROfusion, Grant/Award Number: 101052200; Munich School for Data Science - MUDS, Grant/Award Number: HIDSS-0006

Abstract

Multivariate time series measurements in plasma diagnostics present several challenges when training machine learning models: the availability of only a few labeled data increases the risk of overfitting, and missing data points or outliers due to sensor failures pose additional difficulties. To overcome these issues, we introduce a fast and robust regression model that enables imputation of missing points and data augmentation by massive sampling while exploiting the inherent correlation between input signals. The underlying Student-t process allows for a noise distribution with heavy tails and thus produces robust results in the case of outliers. We consider the state space form of the Student-t process, which reduces the computational complexity and makes the model suitable for high-resolution time series. We evaluate the performance of the proposed method using two test cases, one of which was inspired by measurements of flux loop signals.

KEYWORDS

data augmentation, data imputation, Gaussian processes, multivariate time series, state space models, Student-t processes, surrogate models

1 | INTRODUCTION

Artificial neural networks[1] are a flexible machine learning tool applied in many areas to solve tasks such as nonlinear regression and classification. To produce robust predictions and avoid overfitting, neural networks require a sufficient amount of training data. In common applications of these networks such as image classification, data sets usually contain tens or hundreds of thousands of training samples. This number is then increased artificially by shifted and distorted versions of the existing images—a process known as data augmentation.[2,3] In fusion plasma experiments, diagnostic data consist of correlated high-resolution time series for each discharge. However, the number of discharges of a certain class is small (usually much less than 100). For the straightforward application of artificial neural networks and other "big-data" machine learning methods in this area, it is thus even more important to augment training data by statistically equivalent artificial samples. One possible method has been presented based on the application of Student-t process (TP) surrogates to correlated multi-channel diagnostics data.[4]

An important practical issue has remained unresolved in this previous work; when working with plasma diagnostics data, otherwise regularly spaced data points are partially missing due to sensor failures or non-converging calculation routines and outliers caused by, for example, neutron impact on sensors. Up to now, basic imputation by linear interpolation has been used to insert these points. Due to correlations, signal channels can be reconstructed more intelligently. A multivariate surrogate model should include the solution to this problem naturally: it can capture the dynamics of these series data, impute missing data points, and could serve as a model for data augmentation to generate quasi-realistic training data. So far, the correlation between signal channels has only been accounted for in post-processing via coloring transformations.[4] The present aim is to include correlations directly in the surrogate.

Here, we rely on stochastic processes—of which, the Gaussian process (GP) is the most widely used.[5] In regression with GPs, the posterior distribution of unknown values is inferred, making it a powerful tool for nonlinear multivariate interpolation.[6] Correlations over time are captured by a covariance function or kernel, which may be designed for specific tasks (e.g., to fulfill laws of physics[7,8]). A generalization of GPs are TPs[9] allowing a more robust estimation of mean and variance for data with outliers. Regression with usual GPs or TPs suffers from extensive computing times—scaling cubically in the number of samples—thus making regression with high-resolution time series prohibitively expensive. However, this computational complexity can be reduced when using the state space formulation of the stochastic process by solving a resulting stochastic differential equation via Bayesian filtering and smoothing.[10,11] Previous work on multivariate time series inference with stochastic processes already uses GPs in a state space formulation.[12]

In this paper, we present a state space formulation of a full multivariate TP model using a Matérn cross-covariance kernel[13] and its application to data representative for typical plasma diagnostics signals. Due to the particular structure of the kernel, the underlying correlations of the training data are directly included in the model. Thus, information from other input signals can be used to impute missing data points, resulting in more reliable posterior estimates.

2 | METHODS

2.1 | Multivariate Gaussian and Student-t processes

A multivariate function $\boldsymbol{f}(t) \in \mathbb{R}^d$ that follows a multivariate GP with zero mean is given by[5,6]

$$\boldsymbol{f} \sim \mathcal{GP}\left(\mathbf{0}, \mathbf{K}\left(t, t'\right)\right), \tag{1}$$

where $\mathbf{K}\left(t, t'\right)$ is the positive semidefinite matrix-valued covariance function. Its entries give the covariance between the dimensions of $\boldsymbol{f}(t)$. When performing regression, we aggregate input values into a $d \times n$ design matrix $\mathbf{T}$. Using n observed function values in each dimension d, $\mathbf{Y} \in \mathbb{R}^{d \times n}$ with components $\boldsymbol{y}(t) = \boldsymbol{f}(t) + \boldsymbol{\epsilon}$ that contain local uncorrelated noise $\boldsymbol{\epsilon}$, the posterior mean and covariance for a test data point at t_* are given by

$$\mathbb{E}\left[\boldsymbol{f}(t_*)\right] = \mathbf{k}_*^\top \mathbf{K}_n^{-1} \mathbf{Y}, \tag{2a}$$

$$\mathbb{V}\left[\boldsymbol{f}(t_*)\right] = \mathrm{k}_{**} - \mathbf{k}_*^\top \mathbf{K}_n^{-1} \mathbf{k}_*, \tag{2b}$$

where $\mathbf{k}_* = \mathbf{K}\left(\mathbf{T}, t_*\right)$, $\mathrm{k}_{**} = \mathbf{K}\left(t_*, t_*\right)$, and $\mathbf{K}_n = \mathbf{K}(\mathbf{T}, \mathbf{T}) + \boldsymbol{\Sigma}_n$, where $\boldsymbol{\Sigma}_n \in \mathbb{R}^{nd \times nd}$ is the output noise covariance matrix (in the simplest case, diagonal with entries σ_n^2).

A generalization of multivariate GP regression is given via the Student-t process (TP),[9,14–16]

$$\boldsymbol{f} \sim \mathcal{TP}\left(\mathbf{0}, \mathbf{K}\left(t, t'\right), \tilde{\nu}\right), \tag{3}$$

where the additional hyperparameter $\tilde{\nu} > 2$ controls the tail of the process. When $\tilde{\nu} \to \infty$, we recover a GP, whereas small values of $\tilde{\nu}$ correspond to heavy tails.

Realizations $\boldsymbol{y}(t) = \boldsymbol{f}(t) + \boldsymbol{\epsilon}$ are multivariate Student-t distributed with the density

$$T(\boldsymbol{y} \mid \mathbf{0}, \mathbf{K}, \tilde{\nu}) = \frac{\Gamma\left(\frac{\tilde{\nu}+d}{2}\right)}{\Gamma\left(\frac{\tilde{\nu}}{2}\right)} \frac{1}{((\tilde{\nu}-2)\pi)^{d/2}} \frac{1}{|\mathbf{K}|^{1/2}} \left(1 + \frac{1}{\tilde{\nu}-2} \boldsymbol{y}\mathbf{K}^{-1}\boldsymbol{y}\right)^{-\frac{\tilde{\nu}+d}{2}}, \tag{4}$$

with $\mathbf{K}$ evaluated at the respective t values. In TP regression, the prediction is given in closed form similar to GP regression,[14]

$$\mathbb{E}\left[\boldsymbol{f}(t_*)\right] = \mathbf{k}_*^\top \mathbf{K}_n^{-1} \mathbf{Y}, \tag{5a}$$

$$\mathbb{V}\left[\boldsymbol{f}(t_*)\right] = \frac{\tilde{\nu} - 2 + \mathbf{Y}^\top \mathbf{K}_n^{-1} \mathbf{Y}}{\tilde{\nu} - 2 + d} \left(\mathbf{k}_{**} - \mathbf{k}_*^\top \mathbf{K}_n^{-1} \mathbf{k}_*\right). \tag{5b}$$

For both stochastic processes, the choice of the covariance function is crucial, as it defines the process' behavior.[6] Here, we use a form of multivariate covariance functions,[13]

$$\mathbf{C} = \begin{pmatrix} C_{11} & \cdots & C_{1d} \\ \vdots & \ddots & \vdots \\ C_{d1} & \cdots & C_{dd} \end{pmatrix}, \tag{6}$$

where each C_{ii} is a univariate covariance function and C_{ij} is the cross-covariance function between dimensions. Each univariate covariance between measurements taken at two points separated by distance $\boldsymbol{h}$ is given by a Matérn function $M(\cdot)$,

$$C_{ii} = \sigma_{ii}^2 M\left(\boldsymbol{h} | \nu_{ii}, l_{ii}\right), \tag{7}$$

with variance parameter σ_{ii}^2, smoothness parameter ν_{ii}, and length scale l_{ii}. The Matérn function $M(\cdot)$ reads

$$M_{\nu_{ii}}(\boldsymbol{h}) = \sigma_{ii}^2 \frac{2^{1-\nu_{ii}}}{\Gamma(\nu_{ii})} \left(\sqrt{2\nu_{ii}} \frac{\boldsymbol{h}}{l_{ii}}\right)^{\nu_{ii}} K_{\nu_{ii}} \left(\sqrt{2\nu_{ii}} \frac{\boldsymbol{h}}{l_{ii}}\right), \tag{8}$$

where $\Gamma(\cdot)$ is the gamma function and $K_{\nu_{ii}}(\cdot)$ is the modified Bessel function (second kind). For $\nu_{ii} \to \infty$, we obtain the squared exponential covariance function. The cross-covariances are of the following form:

$$C_{ij} = C_{ji} = \rho_{ij} \sigma_i \sigma_j M\left(\boldsymbol{h} | \nu_{ij}, l_{ij}\right), \tag{9}$$

where ρ_{ij} is the correlation between input dimension i and j. The choice of the hyperparameters ν_{ij} and l_{ij} is crucial, as it is necessary to ensure that the covariance matrix is positive definite.[13] In the following, we set ν_{ij} fixed for all input dimensions to 3/2. This simplifies the transformation into a stochastic differential equation (discussed in Section 2.2) as there exists an exact representation for half-integer values of ν_{ii}.[10] In the present case of plasma diagnostic data with individual and complicated physical processes, abrupt changes can happen over short times. We still want to retain first-order differentiability over time in the results.

This choice simplifies the expression in Equation (8) as now the covariance function can be written as a product of an exponential and polynomial of order $p = 1$:

$$M_{\nu_{ii}=3/2}(\boldsymbol{h}) = \left(1 + \frac{\sqrt{3}\boldsymbol{h}}{l_{ii}}\right) \exp\left(-\frac{\sqrt{3}\boldsymbol{h}}{l_{ii}}\right). \tag{10}$$

In addition, the smoothness of the individual input channels is similar. Therefore, ν_{ii} is the same for all dimensions. In general, however, it is possible to use different smoothness parameters for different dimensions in the presented framework as long as the validity conditions are satisfied to generate a valid covariance matrix. This is explained in detail

in Gneiting et al.[13] Further, we choose $l_{ij}^2 = \frac{1}{2}\left(l_i^2 + l_j^2\right)$. This choice reduces the computational effort of the hyperparameter optimization while still producing satisfying results. In general, however, it is also possible to optimize the cross-covariance lengthscale parameters individually. This choice of hyperparameters in the bivariate full Matérn model simplifies the validity conditions as outlined in Theorem 4 in Gneiting et al.[13] The resulting covariance matrix is valid if the correlation ρ_{12} satisfies[13]

$$|\,\rho_{12}\,| \leq \frac{l_1^{\nu_1} l_2^{\nu_2}}{l_{12}^{\nu_1+\nu_2}} \frac{\Gamma\left(\frac{1}{2}(\nu_1+\nu_2)\right)}{\Gamma(\nu_1)^{1/2}\Gamma(\nu_2)^{1/2}}. \tag{11}$$

All other hyperparameters are optimized by minimizing the negative log-likelihood.

2.2 | Dependent state space Student-t processes

It is possible to represent the multivariate random processes $\boldsymbol{f}$ of Equation (1) with covariance $\mathbf{K}\left(t, t'\right)$ as the solution of a linear time invariant stochastic differential equation,[10,11,17]

$$\frac{d\widehat{\boldsymbol{f}}(t)}{dt} = \mathbf{F}\widehat{\boldsymbol{f}}(t) + \mathbf{L}\boldsymbol{w}(t), \tag{12}$$

$$f\left(t_i\right) = \mathbf{H}\widehat{\boldsymbol{f}}\left(t_i\right), \tag{13}$$

with $\widehat{\boldsymbol{f}}(t) = \left(f_1(t), \frac{df_1(t)}{dt}, \ldots, \frac{d^{m-1}f_1(t)}{dt^{m-1}}, \ldots, f_d(t), \frac{df_d(t)}{dt}, \ldots, \frac{d^{m-1}f_d(t)}{dt^{m-1}}\right)^\top$ containing d processes and their corresponding first m derivatives. In the following part, $\mathbf{Q}$ is the spectral density of a white noise process $\boldsymbol{w}(t)$. $\mathbf{H} = (1, 0, \ldots, 0) \otimes \mathbf{I}_d$ is the observation matrix, $\mathbf{L} = (0, 1, \ldots, 0) \otimes \mathbf{I}_d$ is the noise effect matrix (where $\otimes$ denotes the Kronecker product), and $\mathbf{I}_d$ is the d-dimensional identity matrix. Applying $\mathbf{H}\widehat{\boldsymbol{f}}(t_i)$ yields the observed measurements $(f_1(t), \ldots, f_d(t))$. $\mathbf{F}$ is the state transition matrix, which is derived from the underlying TP we want to transform.[10,11,18]

For discrete time points t_k, the closed-form solution of Equation (13) is[10]

$$\boldsymbol{f}(t_k) = \boldsymbol{f}_k = \mathbf{A}_{k-1}\boldsymbol{f}_{k-1} + \boldsymbol{q}_{k-1}, \tag{14}$$

with normally distributed $\boldsymbol{f}_0 \sim \mathcal{N}(0, \gamma\mathbf{P}_0)$ and $\boldsymbol{q}_{k-1} \sim \mathcal{N}(0, \gamma\mathbf{Q}_{k-1})$, and with γ being an inverse gamma random variable.[14] The state transition covariance matrix is given by $\mathbf{Q}_k = \mathbf{P}_0 - \mathbf{A}_k\mathbf{P}_0\mathbf{A}_k^\top$ and $\mathbf{A}_k = \exp(\mathbf{F}\Delta t_k)$, where Δt_k is the time between subsequent measurements. The initial state covariance $\mathbf{P}_0$ is given by the stationary covariance, which is given in turn by the Lyapunov equation $\mathbf{F}\mathbf{P}_0 + \mathbf{P}_0\mathbf{F}^\top + \mathbf{L}\mathbf{Q}_k\mathbf{L}^\top = \mathbf{0}$.

For the choice of covariance function as described in Equation (6) with $\nu_{ii} = 3/2$, the state transition and the initial covariance matrices are composed of block matrices $\mathbf{F}_{ij}$ and $\mathbf{P}_{0,ij}$, respectively, and are of the following forms:

$$\mathbf{F}_{ij} = \rho_{ij}\begin{pmatrix} 0 & 1 & 0 \\ -\lambda_{ij}^2 & -2\lambda_{ij} & 0 \\ 0 & 0 & -\infty \end{pmatrix}, \tag{15}$$

$$\mathbf{P}_{0,ij} = \rho_{ij}\begin{pmatrix} \sigma_i\sigma_j & 0 & 0 \\ 0 & \sigma_i\sigma_j\lambda_i\lambda_j & 0 \\ 0 & 0 & \sigma_{n,i}\sigma_{n,j} \end{pmatrix}, \tag{16}$$

with $\lambda_{ij} = \frac{\sqrt{6}}{\sqrt{l_i^2 + l_j^2}}$. The measurement noise $\sigma_{n,i}$ is augmented into the state as an additional component.[14] The current choice of the covariance function allows to immediately infer $df(t)/dt$ from the given observations.[11] It is also possible to use derivative observations to regularize the regression. In this case, the observation matrix $\mathbf{H}$ changes to $(1, 1, \ldots, 0) \otimes \mathbf{I}_d$.

Equation (14) can be solved recursively using a Bayesian filter and smoother that estimates the joint posterior distribution given all (noisy) measurements using Bayes' rule[10]:

$$p\left(\boldsymbol{f}_{0:n}|\mathbf{Y}_{1:n}\right)=\frac{p\left(\mathbf{Y}_{1:n}|\boldsymbol{f}_{0:n}\right)\mid p\left(\boldsymbol{f}_{0:n}\right)}{p\left(\mathbf{Y}_{1:n}\right)}. \tag{17}$$

Since the calculation of the full posterior is not computationally feasible, we consider marginal distributions estimated by the Bayesian filter and smoother[10,11,14]:

- filtering distribution $p\left(\boldsymbol{f}_k|\mathbf{Y}_{1:k}\right)$ estimated by a Bayesian filter taking into account the current and all previous measurements;
- predictive distribution of the future state $\boldsymbol{f}_{k+n}$ estimated by a Bayesian filter taking into account all previous measurements: $p\left(\boldsymbol{f}_{k+n}|\mathbf{Y}_{1:k}\right)$;
- smoothing distribution $p\left(\boldsymbol{f}_k|\mathbf{Y}_{1:n}\right)$ estimated by a Bayesian smoother taking into account all measurements.

The marginal distributions for GPs and TPs are calculated as closed-form solutions via a Kalman filter and Rauch–Tung–Striebel smoother[10,11,18] and Student-t filter and smoother,[14] respectively. The smoothing distribution is equivalent to the prediction of a GP or TP given in Equations (2a) and (2b) and Equations (5a) and (5b), respectively.

The advantage of reformulating Equation (1) into Equation (14) is the reduction of computational complexity, as the GP regression scales with $\mathcal{O}\left(N^3\right)$, where $N=nd$ is the total number of observations, while Bayesian filtering and smoothing is only of complexity $\mathcal{O}\left(m^3N\right)$, where m is the number of included derivatives ($m\gg N$).[14]

3 | NUMERICAL EXPERIMENTS

We apply the presented method to two synthetic test cases and evaluate its performance in comparison to an independent state space TP.

The state space TP is trained by optimizing the hyperparameters $l_{ii}, \sigma_{ii}^2, \sigma_{n,ii}^2$ for each univariate Matérn process individually by minimizing the negative log-likelihood.[12,14] The correlation ρ_{ij} as in Equation (9) is the empirical correlation calculated from the training data. Equation (11) is satisfied. Estimated hyperparameters for both test cases are given in Table 1.

3.1 | Synthetic example

We first apply the presented method to 100 data points sampled from two identical time series,

$$f(t+\Delta t)=\sin(0.04\pi t)+\sin(0.07\pi t)+0.2\mathcal{T}(3), \tag{18}$$

TABLE 1 Optimized hyperparameters for the state space Student-t surrogate model for both test cases.

Test case	Hyp	f_1	f_2
(1)	ν_{ii}	3.08	3.08
	$\sigma_{n,ii}^2$	0.15	0.15
	σ_{ii}^2	1.61	1.61
	l_{ii}	18.08	18.08
(2)	ν_{ii}	3.0	3.0
	$\sigma_{n,ii}^2$	0.09	0.09
	σ_{ii}^2	2.06	2.06
	l_{ii}	27.08	25.08

with $t \in [0,100]$ that are shifted in time with an offset of $\Delta t = -0.15$. Both functions are corrupted by individually randomly sampled Student-t distributed noise, $\mathcal{T}(3)$. The correlation between the observed function values is $\rho_{12} = -0.57$. To create artificial missing values, 15 measurements are removed from signal 2 in range $t \in [40, 65]$.

3.2 | Plasma diagnostics data

The second test case is inspired by two correlated flux loop signals during an edge localized mode[19] (ELM). Here, we use a function similar to the plotted signals in the reference, which is overlayed with t-distributed noise: $\boldsymbol{f}(t) = (f_1(t), f_2(t))^\top = \widetilde{\boldsymbol{f}}(t) + \sigma_n \mathcal{T}(3)$, where $\sigma_n = 0.2$. The correlation between f_1 and f_2 is given by $\rho_{12} = -0.68$. In this test case, measurements are missing for f_1 in the range $t \in [8.4,33.6]$ (60 missing points) and for f_2 in the range $t \in [117.6,168.0]$ (120 missing points).

4 | RESULTS AND ANALYSIS

4.1 | Data imputation

For test case (1), the mean and 95% confidence band estimated by the dependent state space TP together with the training data and the true underlying function is shown in the left panels in Figure 1. In comparison to the independent model shown in the right panels in Figure, the mean of f_2 is better captured as the correlation is taken into account. As the

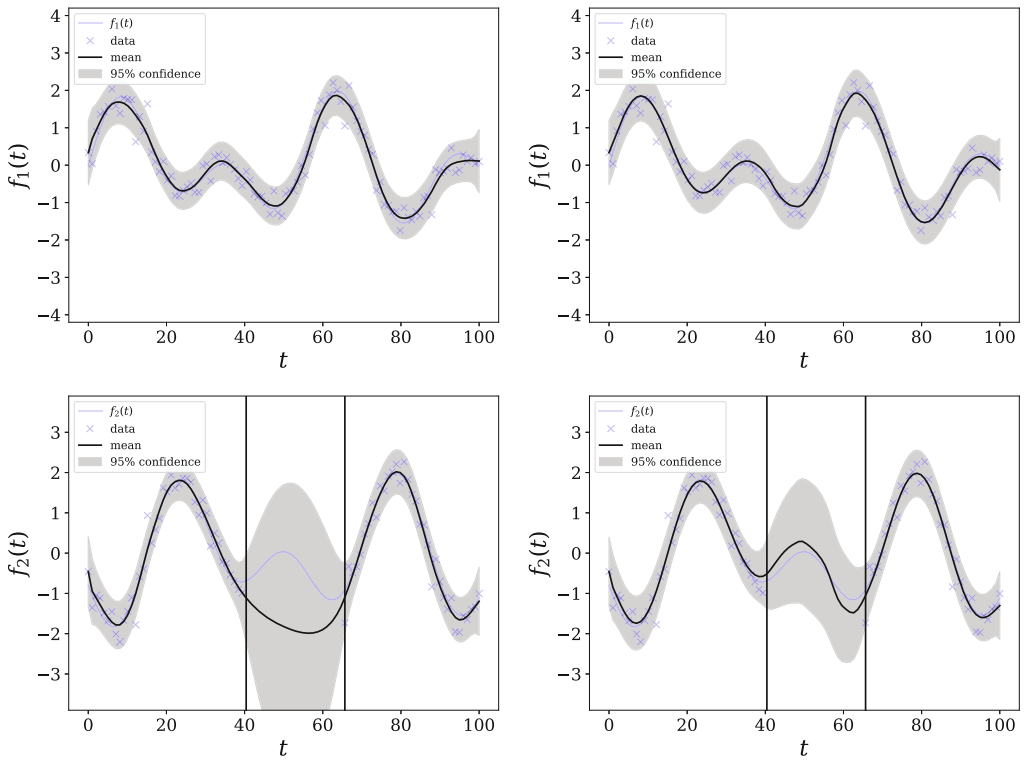


FIGURE 1 Independent (left) and dependent (right) state space TP model applied to test case (1). Vertical bars indicate regions where measurements are missing.

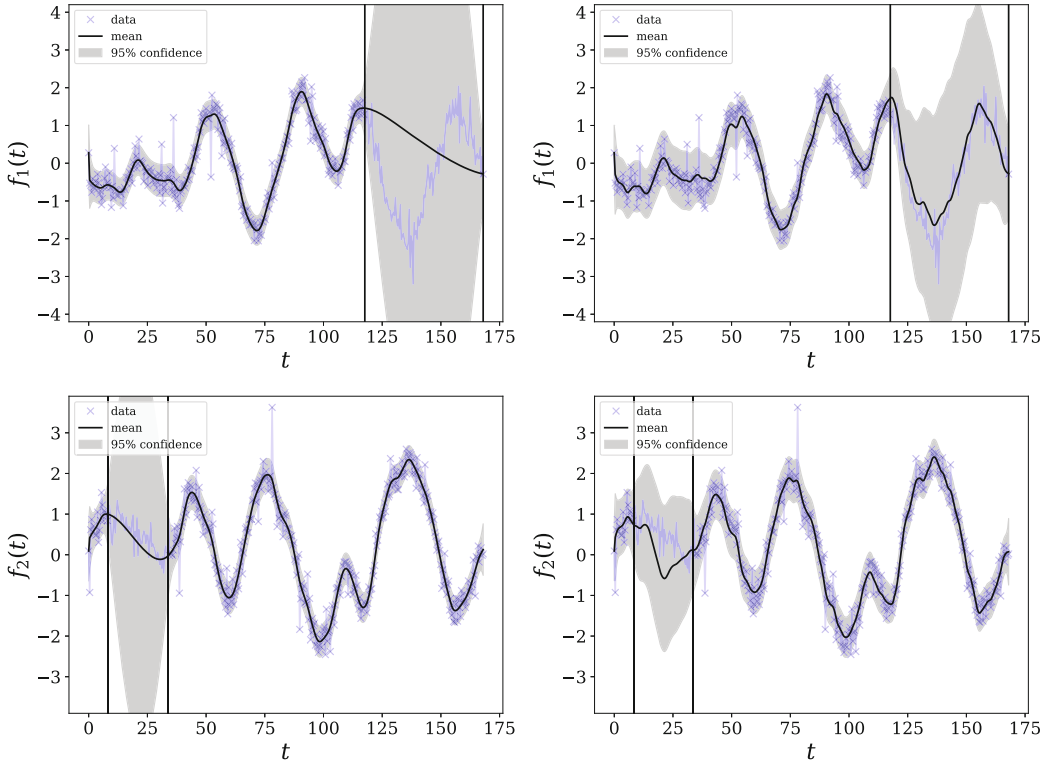


FIGURE 2 Independent (left) and dependent (right) state space TP model applied to test case (2). Vertical bars indicate regions where measurements are missing.

independent model does not have any information in the range $t \in [40, 65]$ where measurements are missing, the estimation of the mean is not as accurate as in the correlated model. We evaluate the mean absolute error (MAE) to the underlying ground truth for both models and find $\text{MAE}_{f_1} = 0.04$ and $\text{MAE}_{f_2} = 0.06$ when using the dependent model, whereas we obtain $\text{MAE}_{f_1} = 0.06$ and $\text{MAE}_{f_2} = 0.22$ for the independent model. The very different results for f_1 and f_2 of the independent model can be explained by the missing measurements in f_2, as the independent model only interpolates between existing measurement points.

The results for test case (2) are shown in Figure 2. For the missing measurements for f_1, the results are similar to test case (1): the independent model is not able to capture the oscillating behavior of the underlying ground truth, and this behavior is only correctly represented by the dependent model. However, in the range $t \in$ [8.4,33.6] where measurements are missing in f_2, the independent model seems to outperform the dependent model, as the overall function behavior is almost linear in this region. Additionally, the local correlation between f_1 and f_2 in this time span seems to be smaller than the overall correlation. Therefore, the dependent model overestimates the correlation. However, the underlying ground truth still lies within the 95% confidence bands estimated by the dependent model. Again, we evaluate the MAE and obtain $\text{MAE}_{f_1} = 0.18$ and $\text{MAE}_{f_2} = 0.16$ when using the dependent model and $\text{MAE}_{f_1} = 0.46$ and $\text{MAE}_{f_2} = 0.08$ for the independent model.

4.2 | Data augmentation

For the two test cases, we further evaluate the performance of the presented method for data augmentation by drawing 1000 samples from the estimated posterior distribution.[20] One sample is shown in Figure 3. We then perform statistical

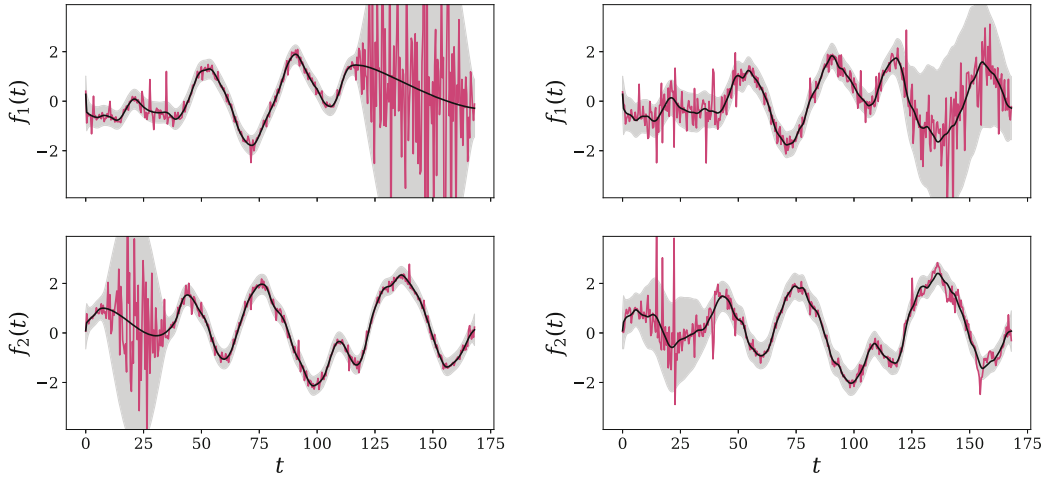


FIGURE 3 Data augmentation using the independent (left) and dependent (right) state space TP for test case (2). Mean and 95% covariance are shown as a black solid line and grey shaded regions, respectively. One sample drawn from the estimated posterior is depicted as a solid line in magenta.

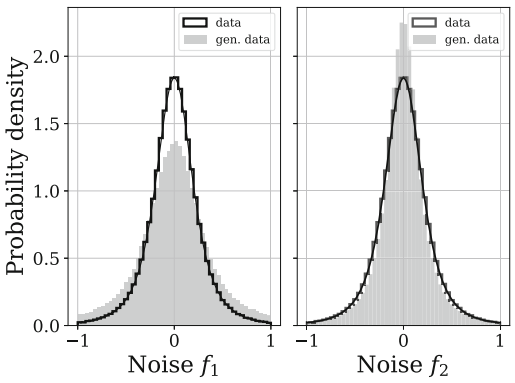


FIGURE 4 Noise distribution of training and generated data for test case (2). The probability density function of a Student-t distribution with $\tilde{\nu} = 3$ is shown as a solid black line.

analyses similar to those performed by Rath et al.[4] to evaluate the quality of the generated data. This is done by comparing the auto- and cross-correlation,

$$\rho_{rs}(t_1, t_2) = \frac{\mathbb{E}\left[\left(\boldsymbol{y}_{r,t_1} - \mu_{r,t_1}\right)\left(\boldsymbol{y}_{s,t_2} - \mu_{s,t_2}\right)\right]}{\sigma_{r,t_1}\sigma_{s,t_2}}, \tag{19}$$

where μ_{i,t_i} is the mean and σ_{i,t_i} is the standard deviation of $\boldsymbol{y}_{i,t_i}$ between the training and generated data using the mean squared error (MSE).

Additionally, we calculate the power spectral density (PSD) that estimates the power distribution across the frequency of a signal. Again, the MSE of the PSD of the training data is calculated.

To compare the distributions of training and generated data, we use the Wasserstein-1 (W_1) metric[21] for each dimension individually and globally over time. We use the W_1 metric also to assess the noise distribution ϵ of training and

TABLE 2 Data augmentation: model comparison for test case (1) for 1000 samples generated from the trained model for statistical metrics described in Section 4

	Independent		Dependent	
Metric	f_1	f_2	f_1	f_2
W_1	**0.03**	0.19	**0.03**	**0.05**
noise distr. W_1	0.08	0.06	**0.04**	**0.03**
MSE ρ_{rs}	**0.0012**		**0.0012**	
MSE PSD $[10^{-6}]$	**6**	14	**6**	**9**

Note: Best values are highlighted in bold.

TABLE 3 Data augmentation: model comparison for test case (2) for 1000 samples generated from the trained model for statistical metrics described in Section 4.

	Independent		Dependent	
Metric	f_1	f_2	f_1	f_2
W_1	0.43	0.11	**0.10**	**0.08**
noise distr. W_1	0.49	0.11	**0.18**	**0.02**
MSE ρ_{rs}	0.012		**0.003**	
MSE PSD $[10^{-5}]$	1494	24	**11**	**2**

Note: Best values are highlighted in bold.

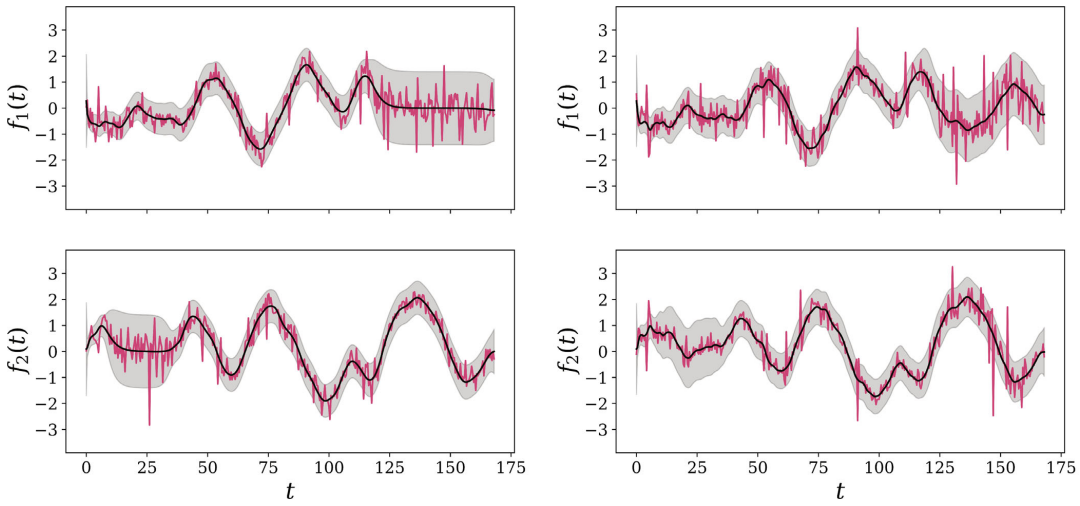


FIGURE 5 Data augmentation using the independent (left) and dependent (right) state space TP for test case (2) with different hyperparameters. Mean and 95% covariance are shown as a black solid line and grey shaded regions, respectively. One sample drawn from the estimated posterior is depicted as a solid line in magenta.

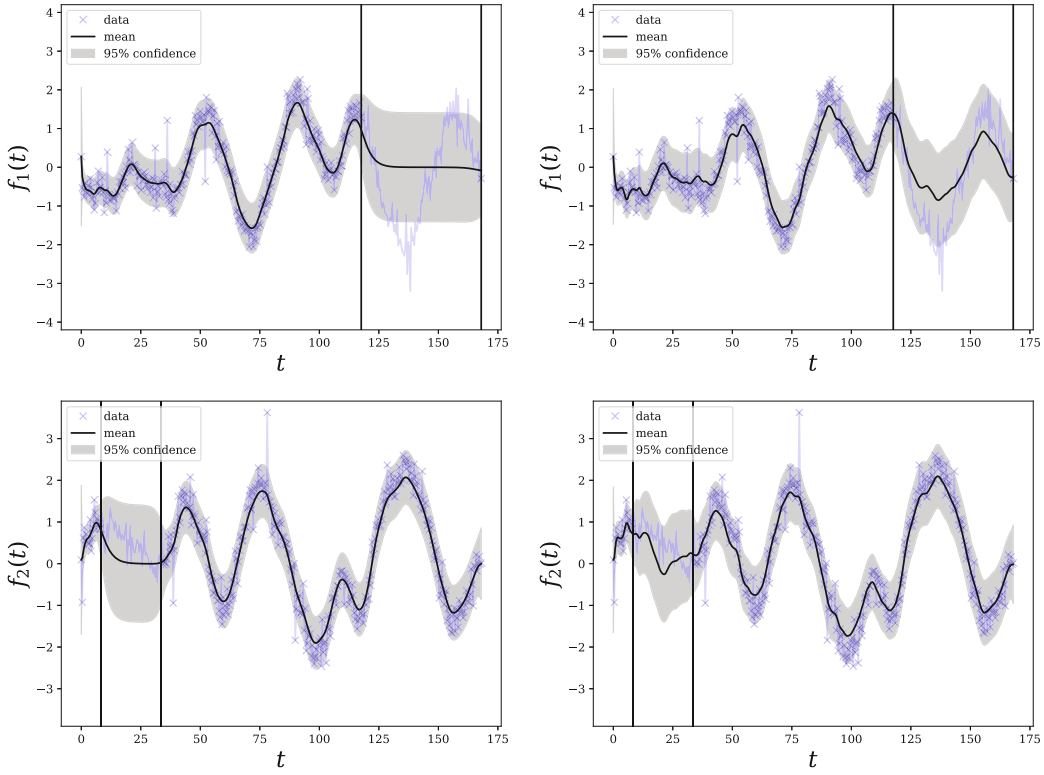


FIGURE 6 Independent (left) and dependent (right) state space TP models applied to test case (2) with different set of hyperparameters. Vertical bars indicate regions where measurements are missing.

generated data in order to see whether the latter follows a Student-t distribution. Additionally, a visual comparison is given in Figure 4 for test case (1). The noise distribution for f_1 has heavier tails, as there are missing measurements and the covariance is larger in this range. Therefore, the drawn samples contain more noise in comparison to the training data.

The results for test cases (1) and (2) are given in Tables 2 and 3, respectively. The local correlation achieved by the dependent model is -0.68 for test case (1) and -0.60 for test case (2). The independent model generates data with a correlation of -0.47 for test case (1) and -0.09 for test case (2). A similar analysis with an independent GP model yields a comparable mean, but overestimates the variance due to outliers. Overall, the dependent TP model outperforms the independent one.

The current choice of hyperparameters produces a quite extensive confidence band due to the estimated combination of noise covariance and length scale. This can be suppressed by a different choice of hyperparameters ($\nu_{11} = \nu_{22} = 3.0$, $\sigma^2_{n,ii} = 0.5$, $\sigma^2_{ii} = 2.0$, $l_{11} = l_{22} = 4$) as depicted in Figure 5. Here, the mean is not captured as well as before when there are missing measurements (shown in Figure 6). In addition, it is possible to use time derivatives in the current framework to further regularize the regression if independent observations by diagnostics dedicated to derivative observation are available.

5 | CONCLUSION

In this paper, we have presented a dependent state space Student-t process model, which directly includes the correlation of multivariate time series by using a multivariate Matérn kernel. The heavy-tailed noise distribution of the TP

allows more robust results when challenging outliers are present in the measurements. The advantage over traditional TP regression is the reduced computational complexity. On the basis of two test cases inspired by real-world problems in plasma diagnostics, we have shown that the dependent model is more accurate for missing data points in comparison to an independent model that does not take correlations between input signals into account. The included correlations do not cause any additional computational effort with the presented choice of hyperparameters. One detail left open for possible future work is the global optimization of hyperparameters without the simplifications of using each signal independently for this purpose. Depending on the complexity of the application, it could be beneficial also to optimize cross-covariance hyperparameters. Here, the trade-off between accuracy and increased computational effort has to be considered. In addition, more complex kernels could be incorporated, for example, different smoothness for different input channels in the Matérn kernel if necessary for the considered application. Overall, we consider the presented approach to be well-suited for data imputation as well as data augmentation in multichannel time series sensor data, in particular for plasma diagnostics. The next step would be to incorporate the augmented samples in the training of black-box machine learning routines for disruption prediction. Another application could be as a tool to augment shot databases by typical samples to be used for uncertainty quantification and their propagation to numerical models that rely on these data.

ACKNOWLEDGMENTS

The present contribution is supported by the Helmholtz Association of German Research Centers under the joint research school HIDSS-0006 “Munich School for Data Science (MUDS)” (KR). This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200-EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them (CA). Open Access funding enabled and organized by Projekt DEAL.

CONFLICT OF INTEREST STATEMENT

The authors have stated explicitly that there are no conflicts of interest in connection with this article.

DATA AVAILABILITY STATEMENT

Data available on request from the authors.

REFERENCES

[1] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag, Berlin, Heidelberg **2006**.
[2] P. Y. Simard, D. Steinkraus, J. C. Platt. Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings. 2003, pp. 958–963.
[3] C. Shorten, T. M. Khoshgoftaar, *J. Big Data* **2019**, *6*, 1.
[4] K. Rath, D. Rügamer, B. Bischl, U. von Toussaint, C. Rea, A. Maris, R. Granetz, C. G. Albert, *J. Plasma Phys.* **2022**, *88*, 895880502.
[5] C. K. I. Williams, C. E. Rasmussen. Advances in Neural Information Processing Systems 8, MIT Press, 1996, pp. 514–520.
[6] M. A. Alvarez, L. Rosasco, N. D. Lawrence. Kernels for Vector-Valued Functions: A Review, 2011. https://arxiv.org/abs/1106.6251
[7] C. G. Albert, K. Rath, *Entropy* **2020**, *22*, 2.
[8] K. Rath, C. G. Albert, B. Bischl, U. von Toussaint, *J. Nonlinear Sci.* **2021**, *31*, 053121.
[9] A. Shah, Andrew Gordon Wilson, Zoubin Ghahramani, Artificial Intelligence and Statistics **2014**.
[10] S. Särkkä, Bayesian Filtering and Smoothing, Cambridge University Press, Cambridge, **2013**.
[11] S. Särkkä, A. Solin, Applied Stochastic Differential Equations, Cambridge University Press, Cambridge, **2019**.
[12] A. Vandenberg-Rodes, B. Shahbaba. Dependent Matérn Processes for Multivariate Time Series, 2015. https://arxiv.org/abs/1502.03466
[13] T. Gneiting, W. Kleiber, M. Schlather, *J. Am. Stat. Assoc.* **2010**, *105*, 1167.
[14] A. Solin, S. Särkkä. State Space Methods for Efficient Inference in Student-t Process Regression, 2015.
[15] B. D. Tracey, D. Wolpert. 2018 AIAA Non-Deterministic Approaches Conference, American Institute of Aeronautics and Astronautics. 2018. https://doi.org/10.2514/2F6.2018-1659
[16] S. Yu, V. Tresp, K. Yu. Proceedings of the 24th International Conferenceon Machine Learning, Association for Computing Machinery, New York, NY, USA, of ICML'07, 2007, p. 1103–1110. https://doi.org/10.1145/1273496.1273635
[17] J. Hartikainen, S. Sarkka. IEEE International Workshop on Machine Learning for Signal Processing 2010, 379–384. **2010**.
[18] S. Sarkka, A. Solin, J. Hartikainen, *IEEE Signal Process. Magaz.* **2013**, *30*, 51.
[19] S. C. Chapman, R. O. Dendy, T. N. Todd, N. W. Watkins, A. J. Webster, F. A. Calderon, J. Morris, *Phys. Plasmas* **2014**, *21*, 062302.

[20] J. Durbin, S. J. Koopman, *Biometrika* **2002**, *89*, 603.
[21] C. Villani, Optimal Transport: Old and New, of Grundlehren der mathematischen Wissenschaften, Springer, Berlin Heidelberg **2008**.

How to cite this article: K. Rath, D. Rügamer, B. Bischl, U. von Toussaint, C. G. Albert, *Contrib. Plasma Phys.* **2023**, *63*(5-6), e202200175. https://doi.org/10.1002/ctpp.202200175

4 Conclusion and Future Work

This thesis comprises five contributing publications addressing different problems for incorporating physical knowledge into machine learning models. Three contributions addressed the direct incorporation of laws of physics into GP regression models using a specialized kernel. The construction of specialized kernels exactly fulfilling underlying PDEs with singular sources is presented in Albert and Rath (2020). The surrogate model representing Hamiltonian flow and Poincaré maps has an exact symplectic property and can be used as a fast orbit emulator and for early classification of chaotic orbits, especially when a closed form expression for Poincaré maps is not available (Rath et al., 2021b,a). The contributions on data augmentation and imputation address the challenge of few (labeled) training data for disruption prediction by presenting two approaches for learning (noisy) multivariate time-series data (Rath et al., 2022, 2023).

While the presented publications answer some questions, there still remain open points that require further exploration and are worth addressing in future work. Several of these have already been mentioned in the respective contributions. In the following, we will examine them further.

Specialized model architectures As discussed in Rath et al. (2021b), SympGPR is naturally limited in the sense that the generating function might be non-unique. We addressed this by splitting the GP into several sub-steps which requires more training but leads to satisfying and stable results in the presented case of field line following in a tokamak with a non-axisymmetric perturbation. However, these intermediate sub-steps can not be identified easily for some Hamiltonian systems. A possibility to tackle the non-uniqueness of the generating function is to consider an unwinding transformation in order to get a unique generating function allowing predictions by the SympGPR. Another path could be the utilization of local experts in phase space: for different regions in phase space, local models are trained to make predictions. However, the combination with symplecticity is non-trivial.
A natural further development is the extension to higher-order integration schemes, e.g., Störmer-Verlet, midpoint, or Gauss-Legendre to improve the prediction accuracy of orbits as well as the Hamiltonian H. Especially, the fast explicit SympGPR scheme in combination with a Störmer-Verlet scheme seems promising to improve the obtained accuracy while still leveraging the fast computation.
The predictive variance is directly available from the SympGPR and could be used for uncertainty quantification. The variance should be considered especially when it comes to orbit classification to distinguish regular from chaotic orbits. In future work, the application case is the distinction between confined and lost alpha particles in fusion devices. Here, a thorough investigation of the transition barrier is needed for different device geometries, similar as in Albert et al. (2023). Additionally, benchmarking the presented approach against existing methods such as the topological classifier (Albert et al., 2023), using the fractal dimension (Albert et al., 2020) or level set learning method (Ruth and Bindel, 2023) provides insight into its performance.

Data augmentation and imputation Both contributions on data augmentation and imputation consider only temporal data. A promising path to extend the proposed model is to consider spatial information of profiles. The state space model can be extended seamlessly to spatio-temporal models (Wilkinson et al., 2020; Aymerich et al., 2022).
The proposed models could further be improved by globally optimizing hyperparameters and also numerically optimizing cross-correlation parameters in the dependent model. The possibly higher accuracy comes, however, with increased computational complexity.
Depending on the application case, another possibility to enhance the flexibility of the model is the incorporation of more complex kernel functions, e.g., Matérn kernels with different smoothness parameters for different input dimensions.
As the training database is imbalanced, especially with regard to different disruption classes, a thorough analysis is crucial for assessing which disruption classes are not available in sufficient quantity. Then, different local models are trained on the input channels under investigation for the different disruption classes. If incoming data is gappy, the correlated model should be used first to impute missing data. From the trained local models, samples are then drawn that augment the database. A limitation of the proposed method is that it cannot handle multi-modality in the training data. This is circumvented by further splitting the training data set.
For the final analysis of the performance of the proposed data augmentation algorithm, the generated samples should be incorporated in the training of black-box ML models for disruption prediction. Here, the robustness of the predictions is the driving criterion.
With regard to future fusion devices such as ITER or SPARC, only little data will be available to train machine learning-based approaches for disruption prediction when operation starts. However, it is of utter importance to mitigate disruptions as they are potentially harmful for the devices. Here, the presented data augmentation model could be utilized to generate samples to train ML models. Whenever new measurement data are available, the local models could be updated.

Contributing Publications

Albert, C. G. and Rath, K. (2020). Gaussian Process Regression for Data Fulfilling Linear Differential Equations with Localized Sources. *Entropy*, 22(2)

Rath, K., Albert, C. G., Bischl, B., and von Toussaint, U. (2021b). Symplectic Gaussian process regression of maps in Hamiltonian systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(5):053121

Rath, K., Albert, C. G., Bischl, B., and von Toussaint, U. (2021a). Orbit Classification and Sensitivity Analysis in Dynamical Systems Using Surrogate Models. *Physical Sciences Forum*, 3(1)

Rath, K., Rügamer, D., Bischl, B., von Toussaint, U., Rea, C., Maris, A., Granetz, R., and Albert, C. G. (2022). Data augmentation for disruption prediction via robust surrogate models. *Journal of Plasma Physics*, 88(5):895880502

Rath, K., Rügamer, D., Bischl, B., von Toussaint, U., and Albert, C. G. (2023). Dependent state space Student-t processes for imputation and data augmentation in plasma diagnostics. *Contributions to Plasma Physics*, 63(5-6):e202200175

Further References

A. Papoulis (1991). *Probability, random variables, and stochastic processes*. McGraw-Hill.

Abarbanel, H., Brown, R., and Kennel, M. (1991). Variation of Lyapunov exponents on a strange attractor. *Journal of Nonlinear Science*, 1:175–199.

Abarbanel, H. D. I. (1992). Local Lyapunov Exponents Computed From Observed Data. *Journal of Nonlinear Science*, 2(3):343–365.

Abdullaev, S. S. (2006). *Construction of Mappings for Hamiltonian Systems and Their Applications*. Springer.

Adler, R. J. (2010). *The Geometry of Random Fields*. Society for Industrial and Applied Mathematics.

Albert, C. G., Buchholz, R., Kasilov, S. V., Kernbichler, W., and Rath, K. (2023). Alpha particle confinement metrics based on orbit classification in stellarators. *Journal of Plasma Physics*, 89(3):955890301.

Albert, C. G., Kasilov, S. V., and Kernbichler, W. (2020). Accelerated methods for direct computation of fusion alpha particle losses within, stellarator optimization. *Journal of Plasma Physics*, 86(2):815860201.

Albert, C. G. and Rath, K. (2020). Gaussian Process Regression for Data Fulfilling Linear Differential Equations with Localized Sources. *Entropy*, 22(2).

Alvarez, M. A., Rosasco, L., and Lawrence, N. D. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266.

Arnold, V. (1989). *Mathematical methods of classical mechanics*, volume 60. Springer.

Aymerich, E., Sias, G., Pisano, F., Cannas, B., Carcangiu, S., Sozzi, C., Stuart, C., Carvalho, P., Fanni, A., and Contributors, J. (2022). Disruption prediction at JET through deep convolutional neural networks using spatiotemporal information from plasma profiles. *Nuclear Fusion*, 62(6):066005.

Benettin, G., Galgani, L., Giorgilli, A., and Strelcyn, J. (1980a). Lyapunov Characteristic Exponents for smooth dynamical systems and for Hamiltonian systems; A method for computing all of them. Part 1: Theory. *Meccanica*, 15:9–20.

Benettin, G., Galgani, L., Giorgilli, A., and Strelcyn, J. (1980b). Lyapunov Characteristic Exponents for smooth dynamical systems and for Hamiltonian systems; A method for computing all of them. Part 2: Numerical application. *Meccanica*, 15:21–30.

Berkery, J. W., Sabbagh, S. A., Bell, R. E., Gerhardt, S. P., and LeBlanc, B. P. (2017). A reduced resistive wall mode kinetic stability model for disruption forecasting. *Physics of Plasmas*, 24(5):056103.

Bertalan, T., Dietrich, F., Mezić, I., and Kevrekidis, I. G. (2019). On learning Hamiltonian systems from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(12):121107.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Boyle, P. and Frean, M. (2004). Dependent Gaussian Processes. In Saul, L., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press.

Brantner, B., de Romemont, G., Kraus, M., and Li, Z. (2023). Structure-Preserving Transformers for Learning Parametrized Hamiltonian systems.

Brantner, B. and Kraus, M. (2023). Symplectic Autoencoders for Model Reduction of Hamiltonian Systems.

Burby, J. W., Tang, Q., and Maulik, R. (2020). Fast neural Poincaré maps for toroidal magnetic fields.

Chen, Z., Zhang, J., Arjovsky, M., and Bottou, L. (2019). Symplectic recurrent neural networks.

Chirikov, B. V. (1979). A universal instability of many-dimensional oscillator systems. *Physics Reports*, 52(5):263–379.

Cockayne, J., Oates, C., Sullivan, T., and Girolami, M. (2017). Probabilistic Numerical Methods for Partial Differential Equations and Bayesian Inverse Problems.

Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., and Ho, S. (2020). Lagrangian neural networks.

Csató, L. and Opper, M. (2002). Sparse On-Line Gaussian Processes. *Neural Computation*, 14(3):641–668.

Duruisseaux, V., Burby, J., and Tang, Q. (2023). Approximation of nearly-periodic symplectic maps via structure-preserving neural networks. *Scientific Reports*, 13.

Eckhardt, B. and Yao, D. (1993). Local Lyapunov exponents in chaotic systems. *Physica D: Nonlinear Phenomena*, 65(1):100–108.

Eckmann, J. P. and Ruelle, D. (1985). Ergodic theory of chaos and strange attractors. *Rev. Mod. Phys.*, 57:617–656.

Ensinger, K., Solowjow, F., Ziesche, S., Tiemann, M., and Trimpe, S. (2023). Structure-Preserving Gaussian Process Dynamics. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part V*, page 140–156, Berlin, Heidelberg. Springer-Verlag.

Eriksson, D., Lee, E., Dong, K., Bindel, D., and Wilson, A. (2018). Scaling Gaussian process regression with derivatives. *Advances in Neural Information Processing Systems*, 2018-December:6867–6877.

Finzi, M., Wang, K. A., and Wilson, A. G. (2020). Simplifying Hamiltonian and Lagrangian Neural Networks via Explicit Constraints. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Fischer, R., Dinklage, A., and Pasch, E. (2003). Bayesian modelling of fusion diagnostics. *Plasma Physics and Controlled Fusion*, 45(7):1095.

Glad, T. and Ljung, L. (2000). *Control Theory*. Control Engineering. Taylor & Francis.

Goldstein, H. (1980). *Classical Mechanics*. Addison-Wesley, 2nd edition.

Greydanus, S., Dzamba, M., and Yosinski, J. (2019). Hamiltonian Neural Networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Hairer, E., Lubich, C., and Wanner, G. (2006). *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*. Springer.

Hartikainen, J. and Sarkka, S. (2010). Kalman filtering and smoothing solutions to temporal gaussian process regression models. *2010 IEEE International Workshop on Machine Learning for Signal Processing*, pages 379–384.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 282–290, Arlington, Virginia, USA. AUAI Press.

Jin, P., Zhang, Z., Zhu, A., Tang, Y., and Karniadakis, G. E. (2020). Sympnets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems. *Neural Networks*, 132:166–179.

José, J. V. and Saletan, E. J. (1998). *Classical Dynamics: A Contemporary Approach*. Cambridge University Press.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440.

Khoshnevisan, D. (2002). *Kolmogorov's Consistency Theorem*, pages 499–500. Springer New York, New York, NY.

Lee, J. (2003). *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics. Springer.

Lichtenberg, A. and Lieberman, M. (1992). *Regular and chaotic dynamics*. Applied mathematical sciences. Springer.

MacKay, D. J. C. (2003). *Information theory, inference and learning algorithms*. Cambridge University Press, Cambridge, England.

Mendes, F. M. and da Costa Júnior, E. A. (2012). Bayesian inference in the numerical solution of Laplace's equation. *AIP Conference Proceedings*, 1443(1):72–79.

Murphy, K. P. (2022). *Probabilistic Machine Learning: An introduction*. MIT Press.

Offen, C. and Ober-Blöbaum, S. (2022). Symplectic integration of learned Hamiltonian systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 32(1):013122.

O'Hagan, A. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(1):1–42.

O'Hagan, A. (1992). Some Bayesian numerical analysis. *Bayesian Statistics*, 4(345–363):4–2.

Ott, E. (2002). *Chaos in Dynamical Systems*. Cambridge University Press, 2 edition.

Pau, A., Fanni, A., Carcangiu, S., Cannas, B., Sias, G., Murari, A., and and, F. R. (2019). A machine learning approach based on generative topographic mapping for disruption prevention and avoidance at JET. *Nuclear Fusion*, 59(10):106017.

Peng, L. and Mohseni, K. (2016). Symplectic Model Reduction of Hamiltonian Systems. *SIAM Journal on Scientific Computing*, 38(1):A1–A27.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017). Inferring solutions of differential equations using noisy multi-fidelity data. *Journal of Computational Physics*, 335:736 – 746.

Rasmussen, C. E. (2003). Gaussian Processes to Speed up Hybrid Monte Carlo for Expensive Bayesian Integrals. In *Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting*. Oxford University Press.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. The MIT Press.

Rath, K., Albert, C. G., Bischl, B., and von Toussaint, U. (2021a). Orbit Classification and Sensitivity Analysis in Dynamical Systems Using Surrogate Models. *Physical Sciences Forum*, 3(1).

Rath, K., Albert, C. G., Bischl, B., and von Toussaint, U. (2021b). Symplectic Gaussian process regression of maps in Hamiltonian systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(5):053121.

Rath, K., Rügamer, D., Bischl, B., von Toussaint, U., and Albert, C. G. (2023). Dependent state space Student-t processes for imputation and data augmentation in plasma diagnostics. *Contributions to Plasma Physics*, 63(5-6):e202200175.

Rath, K., Rügamer, D., Bischl, B., von Toussaint, U., Rea, C., Maris, A., Granetz, R., and Albert, C. G. (2022). Data augmentation for disruption prediction via robust surrogate models. *Journal of Plasma Physics*, 88(5):895880502.

Rea, C. and Granetz, R. S. (2018). Exploratory Machine Learning Studies for Disruption Prediction Using Large Databases on DIII-D. *Fusion Science and Technology*, 74(1-2):89–100.

Rea, C., Montes, K., Erickson, K., Granetz, R., and Tinguely, R. (2019). A real-time machine learning-based disruption predictor in DIII-D. *Nuclear Fusion*, 59(9):096016.

Rea, C., Montes, K. J., Pau, A., Granetz, R. S., and Sauter, O. (2020). Progress Toward Interpretable Machine Learning–Based Disruption Predictors Across Tokamaks. *Fusion Science and Technology*, 76(8):912–924.

Ross, M. and Heinonen, M. (2023). Learning Energy Conserving Dynamics Efficiently with Hamiltonian Gaussian Processes.

Roth, M., Ardeshiri, T., Özkan, E., and Gustafsson, F. (2017). Robust Bayesian Filtering and Smoothing Using Student's t Distribution. *CoRR*, abs/1703.02428.

Ruth, M. and Bindel, D. (2023). Level Set Learning for Poincaré Plots of Symplectic Maps.

Schölkopf, B. and Smola, A. J. (2018). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press.

Shah, A., Wilson, A. G., and Ghahramani, Z. (2014). Student-t processes as alternatives to gaussian processes. *Artificial Intelligence and Statistics*.

Skokos, C. (2009). The Lyapunov Characteristic Exponents and Their Computation. *Lecture Notes in Physics*, page 63–135.

Skokos, C., Bountis, T., and Antonopoulos, C. (2007). Geometrical properties of local dynamics in Hamiltonian systems: The Generalized Alignment Index (GALI) method. *Physica D: Nonlinear Phenomena*, 231(1):30–54.

Slotani, M. (1964). Tolerance regions for a multivariate normal population. *Annals of the Institute of Statistical Mathematics*, 16:135–153.

Solak, E., Murray-smith, R., Leithead, W. E., Leith, D. J., and Rasmussen, C. E. (2003). Derivative observations in Gaussian process models of dynamic systems. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*, pages 1057–1064. MIT Press.

Solin, A. (2016). *Stochastic Differential Equation Methods for Spatio-Temporal Gaussian Process Regression*. PhD thesis, Aalto University, Finland.

Solin, A. and Särkkä, S. (2015). State Space Methods for Efficient Inference in Student-t Process Regression. In Lebanon, G. and Vishwanathan, S. V. N., editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 885–893, San Diego, California, USA. PMLR.

Stewart, I. (1987). The symplectic camel. *Nature*, 329(6134):17–18.

Swiler, L. P., Gulian, M., Frankel, A. L., Safta, C., and Jakeman, J. D. (2020). A survey of constrained Gaussian Process Regression: Approaches and Implementation Challenges. *Journal of Machine Learning for Modeling and Computing*, 1(2):119–156.

Särkkä, S. (2011). Linear operators and stochastic partial differential equations in gaussian process regression. In Honkela, T., Duch, W., Girolami, M., and Kaski, S., editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 151–158, Berlin, Heidelberg. Springer Berlin Heidelberg.

Särkkä, S. (2013). *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge University Press.

Särkkä, S. and Hartikainen, J. (2012). Infinite-Dimensional Kalman Filtering Approach to Spatio-Temporal Gaussian Process Regression. In Lawrence, N. D. and Girolami, M., editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 993–1001, La Palma, Canary Islands. PMLR.

Särkkä, S. and Solin, A. (2019). *Applied Stochastic Differential Equations*. Institute of Mathematical Statistics Textbooks. Cambridge University Press.

Särkkä, S., Solin, A., and Hartikainen, J. (2013). Spatiotemporal learning via infinite-dimensional bayesian filtering and smoothing: A look at gaussian process regression through kalman filtering. *IEEE Signal Processing Magazine*, 30(4):51–61.

Tanaka, Y., Iwata, T., and ueda, n. (2022). Symplectic Spectrum Gaussian Processes: Learning Hamiltonians from Noisy and Sparse Data. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 20795–20808. Curran Associates, Inc.

Tao, T. (2006). *Nonlinear Dispersive Equations: Local and Global Analysis*. Number Nr. 106 in Conference Board of the Mathematical Sciences. Regional conference series in mathematics. American Mathematical Soc.

Theiler, J. (1990). Estimating fractal dimension. *Journal of the Optical Society of America A*, 7(6):1055–1073.

Toth, P., Rezende, D. J., Jaegle, A., Racanière, S., Botev, A., and Higgins, I. (2019). Hamiltonian Generative Networks.

Tracey, B. D. and Wolpert, D. (2018). Upgrading from Gaussian Processes to Student's-T Processes. In *2018 AIAA Non-Deterministic Approaches Conference*. American Institute of Aeronautics and Astronautics.

Werndl, C. (2009). What are the new implications of chaos for unpredictability? *The British Journal for the Philosophy of Science*, 60(1):195–220.

Wilkinson, W. J., Chang, P. E., Andersen, M. R., and Solin, A. (2020). State Space Expectation Propagation: Efficient Inference Schemes for Temporal Gaussian Processes. Technical report.

Zaslavsky, G. (2007). *The Physics of Chaos in Hamiltonian Systems*. Imperial College Press.

Zhang, Y., Leithead, W., and Leith, D. (2005). Time-series Gaussian Process Regression Based on Toeplitz Computation of O(N2) Operations and O(N)-level Storage. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 3711–3716.

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12. Juli 2011, § 8 Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, den 24.01.2024 Katharina Röck