Learning Dynamics from Input-Output Data with Hamiltonian Gaussian Processes

Jan-Hendrik Ewering[1,2] JAN-HENDRIK.EWERING@IMES.UNI-HANNOVER.DE
Robin E. Herrmann[1] ROBIN.ERIK.HERRMANN@STUD.UNI-HANNOVER.DE
Niklas Wahlström[2] NIKLAS.WAHLSTROM@IT.UU.SE
Thomas B. Schön[2] THOMAS.SCHON@UU.SE
Thomas Seel[1] THOMAS.SEEL@IMES.UNI-HANNOVER.DE

[1] *Leibniz Universität Hannover, 30823 Garbsen, Germany*
[2] *Uppsala University, 751 05 Uppsala, Sweden*

Editors: G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

Abstract

Embedding non-restrictive prior knowledge, such as energy conservation laws, into learning methods is a key motive to construct physically consistent dynamics models from limited data, relevant for, e. g., model-based control. Recent work incorporates Hamiltonian dynamics into Gaussian Processes (GPs) to obtain uncertainty-quantifying, energy-consistent models, but these methods rely on—rarely available—velocity or momentum data. In this paper, we study dynamics learning using Hamiltonian GPs and focus on learning solely from input–output data, without relying on velocity or momentum measurements. Adopting a non-conservative formulation, energy exchange with the environment, e. g., through external forces or dissipation, can be captured. We provide a fully Bayesian scheme for estimating probability densities of unknown hidden states, GP hyperparameters, as well as structural hyperparameters, such as damping coefficients. The proposed method is evaluated in a nonlinear simulation case study and compared to a state-of-the-art approach that relies on momentum measurements.

Keywords: Nonlinear System Identification, Gaussian Processes, Physics-informed Learning

1. Introduction

Learning the dynamics of a system is a key approach for enabling high-performance model-based control and system insight, even with limited prior model knowledge. To this end, recent physics-informed machine learning approaches provide well-generalizing and data-efficient learning-based models by incorporating non-restrictive prior knowledge (Geist and Trimpe, 2021). Notable examples include embedding energy conservation laws into neural networks (Cranmer et al., 2019) or GPs (Beckers et al., 2022) to learn models that yield physically interpretable predictions. In this context, choosing a GP is a good idea for (at least) two reasons. First, recent work has shown that kernel-based methods, such as GPs, are particularly well-suited to represent dynamical systems (Ziegler et al., 2024; Scampicchio et al., 2025). Second, GPs inherently provide an uncertainty quantification of the predictions that can benefit the application, e. g., through stochastic control or safe learning (Brunke et al., 2022).

In particular, fusing Hamiltonian inductive biases into GPs is a vital approach for obtaining uncertainty-quantifying dynamics models while ensuring physically plausible, energy-consistent predictions. However, to the best of our knowledge, all existing work relies on measurements of

the entire system state, meaning that, e. g., beyond position measurements, momentum or velocity data is required for training, which is a restrictive assumption in many practical settings. While ad hoc approaches approximate velocities from position data via numerical differentiation, this is highly sensitive to measurement noise (Chartrand, 2011). Moreover, generalized momenta, required within Hamiltonian mechanics, cannot generally be constructed from position data.

In this article, we consider learning of non-conservative dynamics with Hamiltonian GPs. In contrast to previous research—which relies on measurements of momenta or velocities—we address the more realistic problem setting of learning from input-output data only. We propose a novel Hamiltonian GP model, which is linear in its parameters, features simple closed-form gradient expressions, and exhibits computational complexity independent of the training data dimension, thereby improving its applicability in practical settings. For learning the model, we present a fully Bayesian system identification scheme that estimates probability densities of unknown latent states, GP hyperparameters, as well as structural hyperparameters, such as damping coefficients.

2. Related Work

Integrating continuous-time dynamics into learning-based models has attracted significant attention over the last decade, with early work focusing on learning a vector-valued flow map that determines the evolution of a system's state, e. g., using neural ordinary differential equations (Chen et al., 2018). A more contemporary approach is to compute this flow map from an approximation of the system's Hamiltonian or Lagrangian, which reflects the exchange of energy within a system. In these approaches, a learning-based model, e. g., a neural network (Cranmer et al., 2019; Lutter et al., 2019; Hansen et al., 2025) or a GP (Evangelisti and Hirche, 2022; Giacomuzzo et al., 2024; Dai et al., 2024; Beckers et al., 2022) approximates the scalar Hamiltonian or Lagrangian. By applying deterministic operations to the model, such as the Euler-Lagrange equations (Evangelisti and Hirche, 2022; Giacomuzzo et al., 2024; Dai et al., 2024) or a Hamiltonian system structure (Beckers et al., 2022; Greydanus et al., 2019), a physically consistent representation—obeying the underlying energy conservation laws by construction—is obtained. Importantly, incorporating this "algebraic" physics information (Watson et al., 2025) contrasts with simulation-based learning methods, which feature a physics loss based on a known partial differential equation (Raissi et al., 2019).

Table 1: Comparison of recent Hamiltonian GPs ($\boldsymbol{q}$: positions/coordinates; $\boldsymbol{p}$: momenta)

	Exogenous inputs	Dissipative systems	Variables required for training	Reduced-rank GP (comp. efficiency)	GP hyperparameter learning	Structural hyperparameter learning (e. g., damping)
Bertalan et al. (2019)	✗	✗	$\dot{\boldsymbol{q}}, \dot{\boldsymbol{p}}$	✗	✗	✗
Rath et al. (2021); Offen and Ober-Blöbaum (2022)	✗	✗	$\boldsymbol{q}, \boldsymbol{p}$	✗	frequentist	✗
Tanaka et al. (2022)	✗	✓	$\boldsymbol{q}, \boldsymbol{p}$	✓	frequentist	frequentist
Ensinger et al. (2023)	✗	✗	$\boldsymbol{q}, \boldsymbol{p}$	✗	frequentist	✗
Beckers et al. (2022)	✓	✓	$\boldsymbol{q}, \boldsymbol{p}$	✗	frequentist	frequentist
Ross and Heinonen (2023)	✗	✗	$\boldsymbol{q}, \boldsymbol{p}$	✓	frequentist	✗
Hu et al. (2025)	✗	✗	$\dot{\boldsymbol{q}}, \dot{\boldsymbol{p}}$	✗	frequentist	✗
Proposed	✓	✓	$\boldsymbol{q}$	✓	Bayesian	Bayesian

Considering Hamiltonian-based GP approaches (see Table 1), existing work often models the underlying system as conservative (Bertalan et al., 2019; Rath et al., 2021), meaning that there is no energy exchange with the environment. More recent methods consider energy dissipation (Tanaka et al., 2022) and exogenous inputs, such as driving forces and torques (Beckers et al., 2022; Li et al., 2024). Although this represents a step toward realistic problem settings, all existing work on Hamiltonian GPs relies on velocity or momentum measurements, which is a restrictive assumption in many applications. This issue has been noted by Hansen et al. (2025), who provide a *deterministic* neural network-based approach for energy-consistent learning from position data. While uncertainty quantification can be achieved through various frameworks—such as Bayesian neural networks (Jospin et al., 2022) or deep ensembles (Lakshminarayanan et al., 2017)—these probabilistic approaches do not inherently yield physically consistent dynamics. Hence, an open problem is to learn energy-consistent and non-conservative system models from input-output data while also providing an uncertainty quantification.

Moreover, all research on Hamiltonian GPs that considers hyperparameter learning relies on a frequentist assumption and does not provide uncertainty quantification over structural model properties, such as GP hyperparameters or damping coefficients. Lastly, predicting with GPs can be computationally costly when the number of training data points exceeds the "small-data" regime. In this light, only a few studies (Tanaka et al., 2022; Ross and Heinonen, 2023) address the computational burden with approximations to make the practical application of Hamiltonian GPs feasible.

3. Problem Formulation

We are concerned with learning physically consistent state-space models from input-output data. To this end, we consider systems whose dynamics can be described by forced Hamiltonian mechanics. Building on classical mechanics, this formulation provides a modeling paradigm that allows for describing energy storage and dissipation within a system, as well as energy exchange across systems, in a consistent and interpretable fashion. Formally, consider a continuous-time state-space system

$$\dot{\boldsymbol{x}} = (\boldsymbol{J}(\boldsymbol{x}) - \boldsymbol{R}(\boldsymbol{x}))\, \nabla_{\boldsymbol{x}} H(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{w}\,, \qquad \boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{e}\,, \tag{1}$$

where the energy is described by the *unknown* Hamiltonian function $H : \Omega \to \mathbb{R}$. The *unknown* system state $\boldsymbol{x} = [\boldsymbol{q}^\top, \boldsymbol{p}^\top]^\top \in \Omega \subset \mathbb{R}^{n_x}$—consisting of the generalized coordinates $\boldsymbol{q} \in \mathbb{R}^{n_q}$ and the conjugate momenta of the system $\boldsymbol{p} \in \mathbb{R}^{n_p}$—is driven by the exogenous inputs $\boldsymbol{u} \in \mathbb{R}^{n_u}$, and observed through the outputs $\boldsymbol{y} \in \mathbb{R}^{n_y}$. We consider a stochastic setting with normally distributed process noise $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{0}, \widetilde{\boldsymbol{\Sigma}}_w)$ and measurement noise $\boldsymbol{e} \sim \mathcal{N}(\boldsymbol{e} \mid \boldsymbol{0}, \widetilde{\boldsymbol{\Sigma}}_e)$, respectively.

Assumption 1 *The covariances $\widetilde{\boldsymbol{\Sigma}}_w$, $\widetilde{\boldsymbol{\Sigma}}_e$, and the map $\boldsymbol{g} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ are assumed to be known.*

While other noise models could be used, it is often sufficient in practical settings to consider Gaussian noise. The energy exchange in the system is described by the skew-symmetric interconnection matrix $\boldsymbol{J} \in \mathbb{R}^{n_x \times n_x}$, the symmetric, positive semi-definite dissipation matrix $\boldsymbol{R} \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{R} = \boldsymbol{R}^\top \succeq 0$, and the input matrix $\boldsymbol{G} \in \mathbb{R}^{n_x \times n_u}$.

Assumption 2 *The parametric structures of the matrices $\boldsymbol{J}(\boldsymbol{x})$, $\boldsymbol{R}(\boldsymbol{x})$, and $\boldsymbol{G}(\boldsymbol{x})$, i. e., the potentially state-dependent patterns of entries, are assumed to be known, but the parameters themselves are unknown.*

While the *unknown* Hamiltonian captures the potentially highly nonlinear energy landscape of a given system, the matrices $\boldsymbol{J}$, $\boldsymbol{R}$, and $\boldsymbol{G}$ typically encode the basic physical system topology, such as kinematic relationships, which are often available from coarse system understanding.

Specifically, the problem considered in this work is to learn an uncertainty-quantifying model, obeying the underlying physics (1), from sampled input-output data $\{\boldsymbol{u}_t, \boldsymbol{y}_t\}_{t=0}^{T}$. This amounts to estimating the joint conditional distribution[1] $p(\boldsymbol{x}_{0:T}, \boldsymbol{\xi} | \boldsymbol{u}_{0:T}, \boldsymbol{y}_{0:T})$ of hidden states and model parameters $\boldsymbol{\xi}$—describing the Hamiltonian H, as well as the matrices $\boldsymbol{J}(\boldsymbol{x})$, $\boldsymbol{R}(\boldsymbol{x})$, and $\boldsymbol{G}(\boldsymbol{x})$—in a fully Bayesian setting.

4. Reduced-Rank Hamiltonian Gaussian Processes

To construct a physically consistent, uncertainty-quantifying system representation, we model the Hamiltonian function as a zero-mean Gaussian Process (GP), i. e., $H(\boldsymbol{x}) \sim \mathcal{GP}(0, \kappa(\boldsymbol{x}, \boldsymbol{x}'))$, with covariance (kernel) function $k(\boldsymbol{x}, \boldsymbol{x}')$ (Rasmussen and Williams, 2005), as proposed by Beckers et al. (2022). Importantly, not the Hamiltonian function itself, but its gradient is needed for prediction within the model structure (1). Therefore, exploiting that GPs are closed under linear operations, we model the gradient of the Hamiltonian as

$$\nabla_{\boldsymbol{x}} H(\boldsymbol{x}) \sim \mathcal{GP}(\boldsymbol{0}, \nabla_{\boldsymbol{x}} \nabla_{\boldsymbol{x}'} \kappa(\boldsymbol{x}, \boldsymbol{x}')) \,, \tag{2}$$

where applying the differential operators to a differentiable scalar kernel yields a valid positive-definite matrix-valued kernel (Beckers et al., 2022). However, a common drawback of GPs is that their computational complexity and memory requirements scale poorly with the number of training data points, which becomes even more severe in (2) because the partial derivative induces a multi-output GP. Instead, for efficient learning in practical settings, it is desirable that a model features *(i)* a computational complexity independent of the training data dimension, *(ii)* a linear representation to facilitate closed-form learning, and *(iii)* access to computationally simple gradient expressions.

Interestingly, the reduced-rank GP presented by Solin and Särkkä (2020) exhibits all of these properties, which is why it has been exploited in various related papers on parameter learning and state inference (Svensson and Schön, 2017; Ewering et al., 2026). To retain conciseness, we refer to Svensson and Schön (2017) for a justification of the approach and introduce only the main concept subsequently. Loosely speaking, in the reduced-rank GP (Solin and Särkkä, 2020), a covariance function is approximated with a finite-dimensional eigenfunction expansion by encoding the kernel's spectral density S in the frequency domain. The chosen kernel is described by

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') \approx \sum_{k=1}^{M} S(\sqrt{\varrho_k}) \phi_k(\boldsymbol{x}) \phi_k(\boldsymbol{x}') \,, \tag{3}$$

where $\phi_k : \Omega \to \mathbb{R}$ are eigenfunctions of the Laplace operator on the domain $\Omega = [-L_1, L_1] \times \cdots \times [-L_{n_x}, L_{n_x}]$ where the system state resides, and ϱ_k are the corresponding eigenvalues. For this rectangular domain, the eigenfunctions have a closed form, that is

$$\phi_k(\boldsymbol{x}) = \prod_{i=1}^{n_x} \frac{1}{\sqrt{L_i}} \sin\left(\frac{\pi j_{k,i} \left(x_i + L_i\right)}{2L_i}\right) \,, \qquad \varrho_k = \sum_{i=1}^{n_x} \left(\frac{\pi j_{k,i}}{2L_i}\right)^2 \,, \tag{4}$$

1. To improve readability, we sometimes refer to a probability density function as a distribution, and use the short-hand notation $\boldsymbol{x}_{0:T} := \{\boldsymbol{x}_t\}_{t=0}^{T}$ to describe time-series data.

where $\boldsymbol{x} = [x_1, \ldots, x_{n_x}]^\top$, and the indices $j_{k,i}$ determine the frequency of the corresponding eigenfunction (Riutort-Mayol et al., 2023). Predictions of $H(\boldsymbol{x}) \sim \mathcal{GP}(0, \kappa(\boldsymbol{x}, \boldsymbol{x}'))$ can be performed with the reduced-rank GP at computational complexity $\mathcal{O}(M)$ using the basis function expansion

$$\widehat{H}(\boldsymbol{x}) = \sum_{k=1}^{M} a_k \phi_k(\boldsymbol{x}) = \boldsymbol{a}^\top \boldsymbol{\phi}(\boldsymbol{x}) \,, \tag{5}$$

where the vector-valued function $\boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \ldots, \phi_M(\boldsymbol{x})]^\top$, and the weights $\boldsymbol{a} = [a_1, \ldots, a_M]^\top$ follow a distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{V})$, with $\boldsymbol{V} = \operatorname{diag}(S(\sqrt{\varrho_1}), \ldots, S(\sqrt{\varrho_M}))$. Please note that we use the superscript $\widehat{\square}$ to denote the learned counterparts of the quantities in (1). The reduced-rank GP converges to the exact GP in the limit $M, L_1, \ldots, L_{n_x} \to \infty$ (Solin and Särkkä, 2020). Given this, we model the Hamiltonian gradient $\nabla_{\boldsymbol{x}} H$ by taking the partial derivative of (5), that is

$$\nabla_{\boldsymbol{x}} \widehat{H}(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \left(\boldsymbol{a}^\top \boldsymbol{\phi}(\boldsymbol{x}) \right) = \boldsymbol{D}_{\boldsymbol{\phi}}(\boldsymbol{x}) \boldsymbol{a} \,, \tag{6}$$

where $\boldsymbol{D}_{\boldsymbol{\phi}}(\boldsymbol{x}) \in \mathbb{R}^{n_x \times M}$ is the closed-form Jacobian of the transposed basis function vector $\boldsymbol{\phi}^\top(\boldsymbol{x})$. Note that this model is still linear in its parameters $\boldsymbol{a}$, which is convenient for computationally efficient training and prediction. Unfortunately, we do not have direct access to Hamiltonian measurements for learning. Instead, we use particle Gibbs (Svensson and Schön, 2017) to decouple learning the Hamiltonian from inferring unknown hidden variables (see Section 5). For this, we model the current Hamiltonian gradient $\nabla_{\boldsymbol{x}} \widehat{H}(\boldsymbol{x})$ at time t as an auxiliary quantity $\boldsymbol{h}_t \in \mathbb{R}^{n_x}$, i. e.,

$$\boldsymbol{h}_t = \nabla_{\boldsymbol{x}} \widehat{H}(\boldsymbol{x}) \Big|_{\boldsymbol{x}=\boldsymbol{x}_t} + \boldsymbol{v}_t \,, \qquad \boldsymbol{v}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}) \,, \tag{7}$$

which is estimated via Sequential Monte Carlo (SMC). For conjugacy reasons and to describe the model (6) & (7) with a single distribution (Svensson and Schön, 2017; Berntorp, 2021), we express the GP prior as a zero-mean multivariate normal $\mathcal{N}(\boldsymbol{a}|\mathbf{0}, \sigma^2 \boldsymbol{V})$, where the scale σ^2 reflects the noise, and the diagonal covariance $\boldsymbol{V}$ encodes the spectral density of the chosen kernel. While any isotropic kernel, i. e., $\kappa(\boldsymbol{x}, \boldsymbol{x}') = \kappa(||\boldsymbol{x} - \boldsymbol{x}'||)$, can be employed, we resort to using a common squared exponential kernel

$$\kappa_{\text{se}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\ell^2} \right) \,, \qquad S_{\text{se}}(\omega) = \sigma_f^2 (2\pi\ell^2)^{n_x/2} \exp\left(-\frac{\ell^2 \omega^2}{2} \right) \,, \tag{8}$$

in the following, with Euclidean norm $\|\cdot\|$, spectral density $S_{\text{se}}(\cdot)$, as well as σ_f^2 and length scale ℓ as hyperparameters $\boldsymbol{\vartheta}_{\text{K}} := \{\sigma_f^2, \ell\}$. Please note that using isotropic kernels is a valid assumption, provided we expect the Hamiltonian's smoothness not to change drastically from one region of state space to another.

The noise parameter σ^2 is unknown and must be estimated along with the basis function coefficients $\boldsymbol{a}$. To this end, we set an inverse Gamma prior $\mathcal{IG}(\sigma^2|\psi, \nu)$ on σ^2 (Murphy, 2007), with scale ψ and degrees of freedom ν, such that the overall GP prior for parameters $\boldsymbol{\theta} = \{\boldsymbol{a}, \sigma^2\}$ becomes

$$\boldsymbol{a}, \sigma^2 \sim \mathcal{NIG}(\boldsymbol{a}, \sigma^2 | \mathbf{0}, \boldsymbol{V}, \psi, \nu) = \mathcal{N}(\boldsymbol{a}|\mathbf{0}, \sigma^2 \boldsymbol{V}) \mathcal{IG}(\sigma^2|\psi, \nu) \,. \tag{9}$$

For further details on the distributions employed in this reduced-rank GP model, we refer to the Supplementary Material A and Volkmann et al. (2025). Using the gradient approximation $\nabla_{\boldsymbol{x}} \widehat{H}(\boldsymbol{x})$, the state transition function of a non-conservative Hamiltonian GP model can now be constructed as

$$\dot{\boldsymbol{x}} = \left(\widehat{\boldsymbol{J}}(\boldsymbol{x}) - \widehat{\boldsymbol{R}}(\boldsymbol{x}) \right) \nabla_{\boldsymbol{x}} \widehat{H}(\boldsymbol{x}) + \widehat{\boldsymbol{G}}(\boldsymbol{x}) \boldsymbol{u} + \boldsymbol{w} \,, \tag{10}$$

where energy is transferred across the system boundary by the exogenous inputs $\boldsymbol{u}$, and dissipation is reflected through the matrix $\widehat{\boldsymbol{R}}(\boldsymbol{x})$. To align with sampled data, a suitable integration scheme can be applied for discretizing the continuous-time dynamics (10).

To learn the proposed model, the (hyper-)parameters $\boldsymbol{\xi} := \{\boldsymbol{\theta}, \boldsymbol{\vartheta}_{\mathrm{K}}, \boldsymbol{\vartheta}_{\mathrm{S}}\}$ must be determined. In particular, we have *(i)* the Hamiltonian model parameters $\boldsymbol{\theta}$, *(ii)* the GP kernel hyperparameters $\boldsymbol{\vartheta}_{\mathrm{K}}$, and *(iii)* the structural hyperparameters $\boldsymbol{\vartheta}_{\mathrm{S}}$, i. e., the unknown entries of $\widehat{\boldsymbol{J}}(\boldsymbol{x})$, $\widehat{\boldsymbol{R}}(\boldsymbol{x})$, and $\widehat{\boldsymbol{G}}(\boldsymbol{x})$.

5. Bayesian Inference and Learning from Input-Output Data

This section introduces a Bayesian inference and learning scheme for estimating—under the given model structure (10)—the joint distribution $p(\boldsymbol{z}_{0:T}, \boldsymbol{\xi} | \boldsymbol{u}_{0:T}, \boldsymbol{y}_{0:T})$ of latent variables $\boldsymbol{z}_t = \{\boldsymbol{x}_t, \boldsymbol{h}_t\}$ and (hyper-)parameters $\boldsymbol{\xi}$ from input-output[2] data $\{\boldsymbol{u}_t, \boldsymbol{y}_t\}_{t=0}^{T}$. For a fully Bayesian treatment, we model the parameters $\boldsymbol{\theta}$ and the hyperparameters $\boldsymbol{\vartheta} := \{\boldsymbol{\vartheta}_{\mathrm{K}}, \boldsymbol{\vartheta}_{\mathrm{S}}\}$ as random variables and set a prior $p(\boldsymbol{\theta} | \boldsymbol{\vartheta}_{\mathrm{K}})$ and a hyper-prior $p(\boldsymbol{\vartheta})$, respectively.

Given this potentially high-dimensional target distribution, performing state inference and parameter learning can be challenging. To tackle this task, we rely on a Particle Markov Chain Monte Carlo (PMCMC) approach (Andrieu et al., 2010) and break down the overall problem into simpler sub-problems using a particle Gibbs scheme (Lindsten et al., 2014; Volkmann et al., 2025). The overall inference and learning method is illustrated in Figure 1 and detailed in Algorithm 1. Loosely speaking, the learning procedure consists of sequentially sampling from the conditional distributions

i. of latent variables $\boldsymbol{z}_{0:T}$, given the measurements $\boldsymbol{y}_{0:T}$ and (hyper-)parameters $\boldsymbol{\xi}$, and

ii. of (hyper-)parameters $\boldsymbol{\xi} = \{\boldsymbol{\theta}, \boldsymbol{\vartheta}_{\mathrm{K}}, \boldsymbol{\vartheta}_{\mathrm{S}}\}$, given the latent variables $\boldsymbol{z}_{0:T}$.

To perform Step i., we employ a conditional SMC procedure (Lindsten et al., 2014). It provides samples from the density $p(\boldsymbol{z}_{0:T} | \boldsymbol{y}_{0:T}, \boldsymbol{\xi})$, and resembles a standard particle filter with one trajectory fixed to a previous reference trajectory. In Step ii., finding the posterior density of $\boldsymbol{\theta}$ is done in closed form, exploiting the conjugate prior and the parameter-linearity of the model. The posterior density of $\boldsymbol{\vartheta} = \{\boldsymbol{\vartheta}_{\mathrm{K}}, \boldsymbol{\vartheta}_{\mathrm{S}}\}$ is targeted by Metropolis Hastings (MH) steps within the particle Gibbs scheme.

Remark 1 *The proposed inference and learning method resembles the particle Gibbs scheme for GP learning by Svensson and Schön (2017). Algorithm 1 thus represents a valid PMCMC sampler that is guaranteed to asymptotically sample from the true distribution $p(\boldsymbol{z}_{0:T}, \boldsymbol{\xi} | \boldsymbol{y}_{0:T})$, as the number of iterations $k \to \infty$ (Andrieu et al., 2010).*

Remark 2 *While we focus on Bayesian inference and learning in this paper, the reduced-rank Hamiltonian GP model (10) also enables efficient learning in a frequentist setting, exploiting the closed-form expressions for gradient-based hyperparameter optimization. The computational complexity of learning the covariance function parameters is $\mathcal{O}(TM^2)$ for initialization and $\mathcal{O}(M^3)$ per evaluation of the marginal likelihood and its gradient (Solin and Särkkä, 2020). This compares to a complexity of $\mathcal{O}(T^3)$ for each optimizer step using an exact GP (Rasmussen and Williams, 2005).*

2. As commonly done in Bayesian estimation literature (Särkkä and Svensson, 2023), the dependence on exogenous inputs $\boldsymbol{u}_{0:T}$ is not explicitly stated for the remainder of the paper to avoid notational clutter.

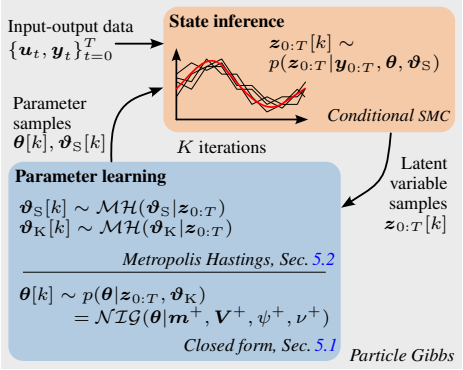

Figure 1: Algorithm overview.

Algorithm 1 Inference and Learning of Hamiltonian GP

Input: Data $\{\boldsymbol{u}_t, \boldsymbol{y}_t\}_{t=0}^T$, state density $p(\boldsymbol{x}_0)$, parameter prior $p(\boldsymbol{\theta}|\boldsymbol{\vartheta}_\mathrm{K})$, hyper-prior $p(\boldsymbol{\vartheta})$
Output: K samples of invariant distr. $p(\boldsymbol{z}_{0:T}, \boldsymbol{\xi}|\boldsymbol{y}_{0:T})$
Initialize $\boldsymbol{z}_{0:T}[0]$ arbitrarily ;
Draw $\boldsymbol{\vartheta}[0] \mid \boldsymbol{z}_{0:T}[0]$; // Sec. 5.2
Draw $\boldsymbol{\theta}[0] \mid \boldsymbol{z}_{0:T}[0], \boldsymbol{\vartheta}[0]$; // Sec. 5.1
for $k = 1$ **to** K **do**
  Draw $\boldsymbol{z}_{0:T}[k] \mid \boldsymbol{y}_{0:T}, \boldsymbol{z}_{0:T}[k-1], \boldsymbol{\theta}[k-1], \boldsymbol{\vartheta}_\mathrm{S}[k-1]$
  Draw $\boldsymbol{\vartheta}[k] \mid \boldsymbol{z}_{0:T}[k]$; // Sec. 5.2
  Draw $\boldsymbol{\theta}[k] \mid \boldsymbol{z}_{0:T}[k], \boldsymbol{\vartheta}_\mathrm{K}[k]$; // Sec. 5.1

5.1. Closed-form Parameter Learning

To construct the posterior of the paramters $\boldsymbol{\theta}$ from the estimated trajectories $\boldsymbol{z}_{0:T}$, we decompose $p(\boldsymbol{\theta}|\boldsymbol{z}_{0:T}, \boldsymbol{\vartheta}) \propto p(\boldsymbol{z}_{0:T}|\boldsymbol{\theta}, \boldsymbol{\vartheta}_\mathrm{S}) p(\boldsymbol{\theta}|\boldsymbol{\vartheta}_\mathrm{K})$ using Bayes' rule. Considering individual time steps, we can write the likelihood as

$$p(\boldsymbol{z}_{0:T}|\boldsymbol{\theta}, \boldsymbol{\vartheta}_\mathrm{S}) = p(\boldsymbol{z}_0) \cdot \prod_{t=1}^{T} p(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1}, \boldsymbol{\theta}, \boldsymbol{\vartheta}_\mathrm{S}) \propto \prod_{t=0}^{T} p(\boldsymbol{h}_t \mid \boldsymbol{x}_t, \boldsymbol{\theta}) \,, \tag{11}$$

where the proportionality is regarding the parameters $\boldsymbol{\theta}$. The factors $p(\boldsymbol{h}_t \mid \boldsymbol{x}_t, \boldsymbol{\theta})$ are normal distributions, defined by (7), and the overall likelihood is, thus, a normal distribution

$$p(\boldsymbol{z}_{0:T}|\boldsymbol{\theta}, \boldsymbol{\vartheta}_\mathrm{S}) = \prod_{t=0}^{T} b_t \exp\left(\sum_{i=1}^{2} \boldsymbol{\alpha}_i^\top \boldsymbol{s}_i(\boldsymbol{z}_t) - \mathrm{Tr}\left(\boldsymbol{P}_i^\top(\boldsymbol{\alpha}) \boldsymbol{r}_i(\boldsymbol{x}_t) \right) \right) , \tag{12}$$

expressed in the canonical form of the restricted exponential family with the natural parameters $\boldsymbol{\alpha}_1 = \boldsymbol{a}/\sigma^2$ and $\alpha_2 = -1/(2\sigma^2)$, respectively, log-partition functions $\boldsymbol{P}_i(\boldsymbol{\alpha})$, as well as (sufficient) statistics $\boldsymbol{s}_i(\boldsymbol{z}_t)$ and $\boldsymbol{r}_i(\boldsymbol{x}_t)$. Please note that $\mathrm{Tr}(\cdot)$ is the trace operator, and that the base measures b_t contain the $\boldsymbol{\theta}$-independent state dynamics $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1}, \boldsymbol{\vartheta}_\mathrm{S})$, defined by (10). Conveniently, the chosen $\mathcal{NIG}$ prior with distribution parameters $\boldsymbol{\eta} = \{\boldsymbol{0}, \boldsymbol{V}, \psi, \nu\}$, i. e.,

$$p(\boldsymbol{\theta}|\boldsymbol{\vartheta}_\mathrm{K}) = \mathcal{NIG}(\boldsymbol{a}, \sigma^2 \mid \boldsymbol{0}, \boldsymbol{V}, \psi, \nu) = n(\boldsymbol{\eta}) \exp\left(\sum_{i=1}^{2} \boldsymbol{\alpha}_i^\top \tilde{\boldsymbol{s}}_i(\boldsymbol{\eta}) - \mathrm{Tr}\left(\boldsymbol{P}_i^\top(\boldsymbol{\alpha}) \tilde{\boldsymbol{r}}_i(\boldsymbol{\eta}) \right) \right) , \tag{13}$$

is a conjugate prior to the likelihood (12), with normalizing factor $n(\boldsymbol{\eta})$, as well as prior statistics $\tilde{\boldsymbol{s}}_i(\boldsymbol{\eta})$, and $\tilde{\boldsymbol{r}}_i(\boldsymbol{\eta})$, respectively. The parameter posterior $p(\boldsymbol{\theta}|\boldsymbol{z}_{0:T}, \boldsymbol{\vartheta})$ is available in closed form by summation of the prior statistics and the new statistics obtained from the estimates $\boldsymbol{z}_{0:T}$, that is $\boldsymbol{s}_i^+ = \tilde{\boldsymbol{s}}_i(\boldsymbol{\eta}) + \sum_{t=0}^{T} \boldsymbol{s}_i(\boldsymbol{z}_t)$ and $\boldsymbol{r}_i^+ = \tilde{\boldsymbol{r}}_i(\boldsymbol{\eta}) + \sum_{t=0}^{T} \boldsymbol{r}_i(\boldsymbol{x}_t)$. The resulting posterior density is

$$p(\boldsymbol{\theta}|\boldsymbol{z}_{0:T}, \boldsymbol{\vartheta}_\mathrm{K}) = \mathcal{NIG}(\boldsymbol{a}, \sigma^2 \mid \boldsymbol{m}^+, \boldsymbol{V}^+, \psi^+, \nu^+) \,, \tag{14}$$

with the posterior distribution parameters $\boldsymbol{\eta}^+ = \{\boldsymbol{m}^+, \boldsymbol{V}^+, \psi^+, \nu^+\}$ being

$$\boldsymbol{m}^+ = \left(\boldsymbol{r}_1^+\right)^{-1} \boldsymbol{s}_1^+ \,, \qquad \boldsymbol{V}^+ = \left(\boldsymbol{r}_1^+\right)^{-1} \,, \qquad \psi^+ = s_2^+ - \boldsymbol{s}_1^{+\top} \left(\boldsymbol{r}_1^+\right)^{-1} \boldsymbol{s}_1^+ \,, \qquad \nu^+ = r_2^+ \,. \tag{15}$$

The full derivation can be found in the Supplementary Material A.

5.2. Hyperparameter Learning: Metropolis-within-Gibbs

For the hyperparameters $\boldsymbol{\vartheta} = \{\boldsymbol{\vartheta}_\mathrm{K}, \boldsymbol{\vartheta}_\mathrm{S}\}$, similar closed-form results are not available. Therefore, we employ Metropolis-within-Gibbs steps for learning, and, in each iteration k, generate a hyperparameter sample $\boldsymbol{\vartheta}^*$ from a proposal distribution $\zeta(\boldsymbol{\vartheta}^*|\boldsymbol{\vartheta}[k])$, e. g., a random walk. The proposals are accepted with probability $\min\left(1, \frac{p(\boldsymbol{\vartheta}^*|\boldsymbol{z}_{0:T})}{p(\boldsymbol{\vartheta}[k]|\boldsymbol{z}_{0:T})} \frac{\zeta(\boldsymbol{\vartheta}[k]|\boldsymbol{\vartheta}^*)}{\zeta(\boldsymbol{\vartheta}^*|\boldsymbol{\vartheta}[k])}\right)$, and rejected otherwise, i. e., $\boldsymbol{\vartheta}[k+1] = \boldsymbol{\vartheta}[k]$. To evaluate $p(\boldsymbol{\vartheta}^*|\boldsymbol{z}_{0:T}) \propto p(\boldsymbol{z}_{0:T}|\boldsymbol{\vartheta}^*)p(\boldsymbol{\vartheta}^*)$, we use the hyperparameter prior $p(\boldsymbol{\vartheta}^*)$ and compute the likelihood $p(\boldsymbol{z}_{0:T}|\boldsymbol{\vartheta}^*)$ for the kernel and structural hyperparameters individually.

Kernel Hyperparameters For the likelihood of the GP kernel hyperparameters $\boldsymbol{\vartheta}_\mathrm{K}$, we have

$$p(\boldsymbol{z}_{0:T}|\boldsymbol{\vartheta}_\mathrm{K}^*) = \frac{p(\boldsymbol{\theta}|\boldsymbol{\vartheta}_\mathrm{K}^*)p(\boldsymbol{z}_{0:T}|\boldsymbol{\theta}, \boldsymbol{\vartheta}_\mathrm{K}^*)}{p(\boldsymbol{\theta}|\boldsymbol{z}_{0:T}, \boldsymbol{\vartheta}_\mathrm{K}^*)} = \frac{n(\boldsymbol{\eta}^*)\prod_{t=0}^{T} b_t}{n(\boldsymbol{\eta}^{+*})}\,, \tag{16}$$

using Bayes' rule, and we note that all terms are known in closed form. The numerator and the denominator in (16) are proportional to each other. In fact, both follow a $\mathcal{NIG}$ distribution, and the $\boldsymbol{\theta}$-related components cancel. Therefore, the likelihood for the kernel hyperparameters is a quotient of normalizing constants, and the first term of the acceptance probability can be computed as

$$\frac{p(\boldsymbol{\vartheta}_\mathrm{K}^*|\boldsymbol{z}_{0:T})}{p(\boldsymbol{\vartheta}_\mathrm{K}[k]|\boldsymbol{z}_{0:T})} = \frac{p(\boldsymbol{z}_{0:T}|\boldsymbol{\vartheta}_\mathrm{K}^*)p(\boldsymbol{\vartheta}_\mathrm{K}^*)}{p(\boldsymbol{z}_{0:T}|\boldsymbol{\vartheta}_\mathrm{K}[k])p(\boldsymbol{\vartheta}_\mathrm{K}[k])} = \frac{n(\boldsymbol{\eta}^*)p(\boldsymbol{\vartheta}_\mathrm{K}^*)}{n(\boldsymbol{\eta}^{+*})} \frac{n(\boldsymbol{\eta}^+[k])}{n(\boldsymbol{\eta}[k])p(\boldsymbol{\vartheta}_\mathrm{K}[k])}\,, \tag{17}$$

where the base measure products $\prod_{t=0}^{T} b_t$ in the numerator and the denominator cancel.

Structural Hyperparameters For the structural hyperparameters $\boldsymbol{\vartheta}_\mathrm{S}$, the likelihood can be constructed by integrating out the parameters $\boldsymbol{\theta}$ from $p(\boldsymbol{z}_{0:T}, \boldsymbol{\theta}|\boldsymbol{\vartheta}_\mathrm{S})$. This is done by noting that the result is proportional regarding $\boldsymbol{\vartheta}_\mathrm{S}$ to the Gaussian density $p(\boldsymbol{x}_{0:T} \mid \boldsymbol{h}_{0:T-1}, \boldsymbol{\vartheta}_\mathrm{S})$, that is

$$\begin{aligned} p(\boldsymbol{z}_{0:T}|\boldsymbol{\vartheta}_\mathrm{S}) &= \int \left[p(\boldsymbol{x}_0) \cdot \prod_{t=0}^{T} p(\boldsymbol{h}_t \mid \boldsymbol{x}_t, \boldsymbol{\theta}) \prod_{t=1}^{T} p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1}, \boldsymbol{\vartheta}_\mathrm{S}) p(\boldsymbol{\theta}) \right] \mathrm{d}\boldsymbol{\theta} \\ &\propto p(\boldsymbol{x}_0) \cdot \prod_{t=1}^{T} p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1}, \boldsymbol{\vartheta}_\mathrm{S}) = p(\boldsymbol{x}_{0:T} \mid \boldsymbol{h}_{0:T-1}, \boldsymbol{\vartheta}_\mathrm{S})\,. \end{aligned} \tag{18}$$

Thus, we compute the first quotient of the acceptance probability as

$$\frac{p(\boldsymbol{\vartheta}_\mathrm{S}^*|\boldsymbol{z}_{0:T})}{p(\boldsymbol{\vartheta}_\mathrm{S}[k]|\boldsymbol{z}_{0:T})} = \frac{p(\boldsymbol{x}_{0:T} \mid \boldsymbol{h}_{0:T-1}, \boldsymbol{\vartheta}_\mathrm{S}^*)p(\boldsymbol{\vartheta}_\mathrm{S}^*)}{p(\boldsymbol{x}_{0:T} \mid \boldsymbol{h}_{0:T-1}, \boldsymbol{\vartheta}_\mathrm{S}[k])p(\boldsymbol{\vartheta}_\mathrm{S}[k])}\,, \tag{19}$$

where the normalizing constants cancel.

6. Simulation and Results

To evaluate the proposed method, we conduct a simulation case study with a non-harmonic oscillator, governed by the Hamiltonian $H(q,p) = \frac{q^2}{2} + \frac{p^2}{2} + 2\cos q$, with the position q and the momentum p (see Figure 2). The system is driven by a known input force u and dissipates

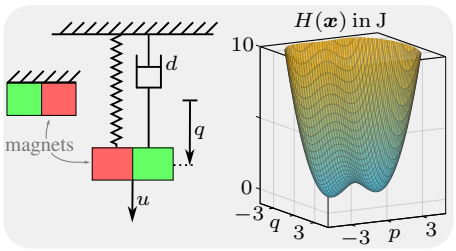

Figure 2: Test system & Hamiltonian.

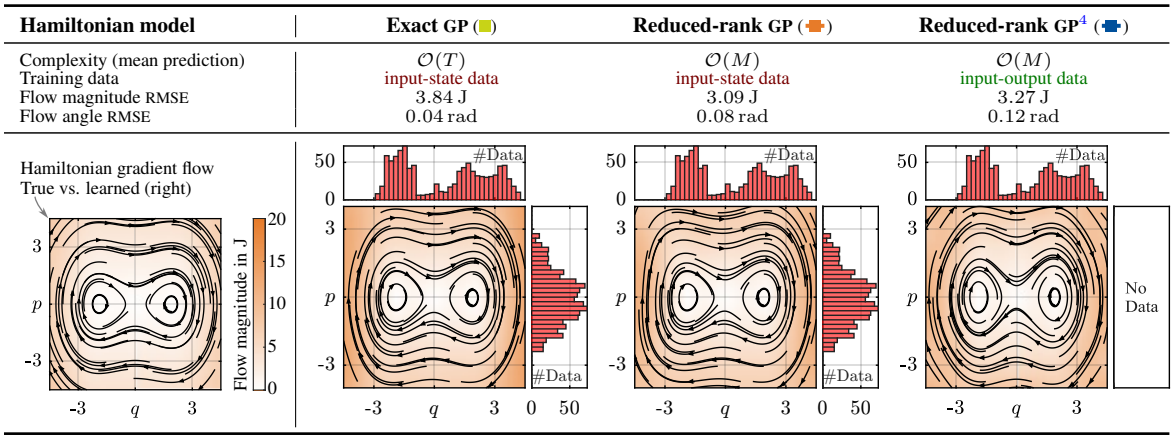

Figure 3: Flow maps following from the true and learned Hamiltonians. The reduced-rank Hamiltonian GP yields similar approximation accuracy while allowing for computationally efficient prediction. Despite only having access to input-output data, the proposed method (right) enables learning a Hamiltonian GP with accuracy comparable to methods with access to full-state measurements.

energy through a damping coefficient $d =: \vartheta_{\mathrm{S}}$, to be estimated. To generate training data, we simulate a single trajectory by applying an input signal u_t for time steps $t = 0, \ldots, T$, and storing only noisy position measurements $y_t^{(\mathrm{i/o})} = q_t + e_t$ as outputs.[3] For comparison with existing work that relies on full-state measurements, we also consider an input-state setting, i. e., $\boldsymbol{y}_t^{(\mathrm{i/s})} = [q_t, p_t]^\top + \boldsymbol{e}_t$. In particular, we compare the proposed Algorithm 1 for physically consistent learning from input-output data to *(i)* itself in an input-state setting, and to *(ii)* an exact Hamiltonian GP model based on Beckers et al. (2022). Please note that we set non-informative priors on all parameters, for instance, broad Gaussian distributions for the hyper-prior $p(\boldsymbol{\vartheta})$.

In Figure 3, the flow map resulting from the true Hamiltonian (left) is compared to the learned Hamiltonian models in terms of their Root Mean Squared Error (RMSE) and computational complexity. While the algorithms in the middle columns have access to state measurements, the proposed approach (right) uses only noisy position measurements, i. e., input-output data. Despite only having access to input-output data, the proposed method learns the flow map with accuracy comparable to approaches that have access to full-state measurements. Please note that learning the considered non-harmonic oscillator system from input-output data required incorporating a symmetry constraint.[4] In this regard, future work may examine conditions for the Hamiltonian's identifiability.

Figure 4 depicts the Bayesian inference and learning results on the training data set, showcasing that the method provides density estimates for the model (hyper-)parameters, and accurate smoothing estimates for all state variables (including unknown hidden states). If we run Algorithm 1 with measurements $\boldsymbol{y}_{0:T}^{(\mathrm{i/s})}$, the hyperparameter density estimates are pronounced, and the density $p(d|\boldsymbol{y}_{0:T}^{(\mathrm{i/s})})$ accurately reflects the true damping coefficient. In comparison, the density $p(d|y_{0:T}^{(\mathrm{i/o})})$ is slightly biased, which we attribute to identifiability issues in the chosen simulation example.

3. Further details on the simulation (including the matrices $\boldsymbol{J}, \boldsymbol{R}, \boldsymbol{G}$), the algorithmic setup, as well as the training/testing scenarios can be found in the Suppl. Material B. Convergence metrics are given in Suppl. Material C.

4. Due to identifiability issues in the input-output setting, we impose a symmetry constraint on the Hamiltonian GP, i. e., $\widehat{H}(\boldsymbol{x}) = \widehat{H}(-\boldsymbol{x})$, which induces an anti-symmetric gradient $\nabla_{\boldsymbol{x}} \widehat{H}(\boldsymbol{x}) = -\nabla_{\boldsymbol{x}} \widehat{H}(-\boldsymbol{x})$. Results without symmetry constraint are presented in the Suppl. Material D. No symmetry constraint is used in the input-state setting.

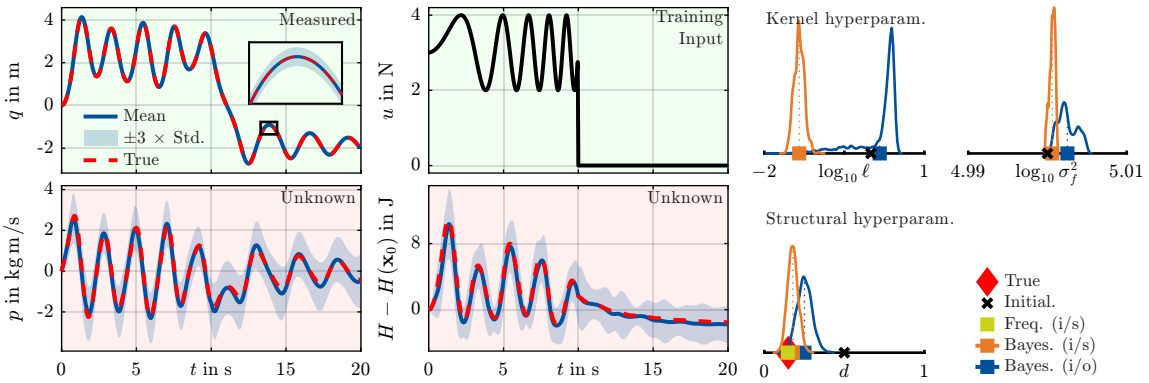

Figure 4: True and estimated system behavior in the training data set (left), as well as density estimates for the GP kernel and structural hyperparameters $\boldsymbol{\vartheta} = \{\boldsymbol{\vartheta}_\mathrm{K}, \vartheta_\mathrm{S}\}$ (right). From input-output data, the proposed method infers densities of unknown hidden states, the current system energy, and hyperparameters in a fully Bayesian fashion.

To evaluate the physical consistency of the learned Hamiltonian GP, we draw models from the sample-based posterior distribution, provided by Algorithm 1, and compare their forward simulations from an initial value with a true system trajectory (see Figure 5 and details in the Supplementary Material B). Despite learning from input-output data, Algorithm 1 yields a physically consistent, probabilistic model whose samples closely resemble the actual system behavior. In fact, the method can represent multimodal densities in phase space, as can be seen in a single sample converging to the system's upper equilibrium (compare Figure 2 and Figure 5, top left). Notably, the method consistently accounts for dissipation and energy intake through exogenous inputs, as visible in the monotonically decreasing Hamiltonian H after the input signal reaches zero.

7. Conclusion

In this article, we present a method for learning a physically consistent dynamics model using non-conservative Hamiltonian GPs. In contrast to existing work, which relies on measurements of momenta or velocities, we consider learning solely based on input-output data. The proposed reduced-rank Hamiltonian GP is linear in its parameters and comes with closed-form gradient expressions. While we exploit these properties for fully Bayesian learning of (hyper-)parameters, we emphasize that the model structure can also speed up training and prediction *significantly* in frequentist settings. However, we acknowledge that the computational complexity of the particle-based scheme and the reduced-rank GP increases rapidly with the problem dimension (Riutort-Mayol et al., 2023), which future work should consider. Moreover, the presented empirical case study necessitated prior knowledge in the form of symmetry constraints to learn a complex Hamiltonian function, highlighting the relevance of a future identifiability analysis.

Taking a step back, we present an approach to bridge physics-informed dynamics learning (Beckers et al., 2022) and physically consistent Bayesian system identification. Using the proposed method, an uncertainty-quantifying, physically consistent nonlinear system model can be learned solely from input-output data, potentially enabling model-based control and insights in complex application systems without extensive prior knowledge.

Acknowledgments

This research was supported by the *Kjell och Märta Beijer Foundation* and the *Swedish Research Council (VR)* under the contract numbers 2021-04321 and 2025-04318. Jan-Hendrik Ewering was supported by the *German Academic Scholarship Foundation (Studienstiftung des Deutschen Volkes)*.

References

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov Chain Monte Carlo Methods. *Journal of the Royal Statistical Society*, 72(3):269–342, 2010. doi: 10.1111/j.1467-9868.2009.00736.x.

Thomas Beckers, Jacob Seidman, Paris Perdikaris, and George J. Pappas. Gaussian Process Port-Hamiltonian Systems: Bayesian Learning with Physics Prior. In *Conf. on Decision and Control*, pages 1447–1453. IEEE, 2022. doi: 10.1109/CDC51059.2022.9992733.

Karl Berntorp. Online Bayesian inference and learning of Gaussian-process state–space models. *Automatica*, 129:109613, 2021. doi: 10.1016/j.automatica.2021.109613.

Karl Berntorp and Marcel Menner. Online Constrained Bayesian Inference and Learning of Gaussian-Process State-Space Models. In *American Control Conf.*, pages 940–945. IEEE, 2022. doi: 10.23919/ACC53348.2022.9867635.

Tom Bertalan, Felix Dietrich, Igor Mezić, and Ioannis G. Kevrekidis. On learning Hamiltonian systems from data. *Chaos*, 29(12):121107, 2019. doi: 10.1063/1.5128231.

Lukas Brunke, Melissa Greeff, Adam W. Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P. Schoellig. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):411–444, 2022. doi: 10.1146/annurev-control-042920-020211.

Rick Chartrand. Numerical Differentiation of Noisy, Nonsmooth Data. *ISRN Applied Mathematics*, 2011:1–11, 2011. doi: 10.5402/2011/164564.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, 2018.

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian Neural Networks. In *ICLR Workshop on Integration of Deep Neural Models and Differential Equations*, 2019.

Rui Dai, Giulio Evangelisti, and Sandra Hirche. Physically consistent modeling & identification of nonlinear friction with dissipative Gaussian processes. In *Learning for Dynamics & Control Conference*, volume 242, pages 1415–1426. PMLR, 2024.

Katharina Ensinger, Friedrich Solowjow, Sebastian Ziesche, Michael Tiemann, and Sebastian Trimpe. Structure-Preserving Gaussian Process Dynamics. In *Machine Learning and Knowledge Discovery in Databases*, volume 13717, pages 140–156. Springer Nature, Cham, 2023.

Giulio Evangelisti and Sandra Hirche. Physically Consistent Learning of Conservative Lagrangian Systems with Gaussian Processes. In *Conf. on Decision and Control*, pages 4078–4085. IEEE, 2022. doi: 10.1109/CDC51059.2022.9993123.

Jan-Hendrik Ewering, Max Bartholdt, Simon F. G. Ehlers, Niklas Wahlström, Thomas B. Schön, and Thomas Seel. Simultaneous State Estimation and Online Model Learning in a Soft Robotic System. In *29th Int. Conf. on Information Fusion (FUSION)*. IEEE, 2026. doi: 10.48550/arXiv.2602.14092.

A. René Geist and Sebastian Trimpe. Structured learning of rigid–body dynamics: A survey and unified view from a robotics perspective. *GAMM-Mitteilungen*, 44(2), 2021. doi: 10.1002/gamm.202100009.

Giulio Giacomuzzo, Ruggero Carli, Diego Romeres, and Alberto Dalla Libera. A Black-Box Physics-Informed Estimator Based on Gaussian Process Regression for Robot Inverse Dynamics Identification. *IEEE Transactions on Robotics*, 40:4820–4836, 2024. doi: 10.1109/TRO.2024.3474851.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian Neural Networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, 2019.

Martine Dyring Hansen, Elena Celledoni, and Benjamin Kwanen Tapley. Learning mechanical systems from real-world data using discrete forced Lagrangian dynamics. *preprint, arXiv: 2505.20370*, 2025. doi: 10.48550/arXiv.2505.20370.

Jianyu Hu, Juan-Pablo Ortega, and Daiying Yin. A structure-preserving kernel method for learning Hamiltonian systems. *Mathematics of Computation*, 2025. doi: 10.1090/mcom/4106.

Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. Hands-On Bayesian Neural Networks—A Tutorial for Deep Learning Users. *IEEE Computational Intelligence Magazine*, 17(2):29–48, 2022. doi: 10.1109/MCI.2022.3155327.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, 2017.

Peilun Li, Kaiyuan Tan, and Thomas Beckers. PyGpPHs: A Python Package for Bayesian Modeling of Port-Hamiltonian Systems. *IFAC-PapersOnLine*, 58(6):54–59, 2024. doi: 10.1016/j.ifacol.2024.08.256.

Fredrik Lindsten, Michael I. Jordan, and Thomas B. Schön. Particle Gibbs with Ancestor Sampling. *Journal of Machine Learning Research*, 15(63):2145–2184, 2014.

Michael Lutter, Christian Ritter, and Jan Peters. Deep Lagrangian Networks: Using Physics as Model Prior for Deep Learning. In *Int. Conf. on Learning Representations*, 2019.

Kevin P. Murphy. Conjugate Bayesian analysis of the Gaussian distribution. *Technical Report*, 2007.

C. Offen and S. Ober-Blöbaum. Symplectic integration of learned Hamiltonian systems. *Chaos*, 32 (1):013122, 2022. doi: 10.1063/5.0065913.

M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045.

Carl Edward Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2005. doi: 10.7551/mitpress/3206.001.0001.

Katharina Rath, Christopher G. Albert, Bernd Bischl, and Udo von Toussaint. Symplectic Gaussian process regression of maps in Hamiltonian systems. *Chaos*, 31(5):053121, 2021. doi: 10.1063/5.0048129.

Gabriel Riutort-Mayol, Paul-Christian Bürkner, Michael R. Andersen, Arno Solin, and Aki Vehtari. Practical Hilbert space approximate Bayesian Gaussian processes for probabilistic programming. *Statistics and Computing*, 33(1), 2023. doi: 10.1007/s11222-022-10167-2.

Gareth O. Roberts and Osnat Stramer. Langevin Diffusions and Metropolis-Hastings Algorithms. *Methodology And Computing In Applied Probability*, 4(4):337–357, 2002. doi: 10.1023/A:1023562417138.

Magnus Ross and Markus Heinonen. Learning Energy Conserving Dynamics Efficiently with Hamiltonian Gaussian Processes. *Transactions on Machine Learning Research*, 2023.

Simo Särkkä and Lennart Svensson. *Bayesian filtering and smoothing*. Cambridge University Press, New York, 2 edition, 2023. ISBN 9781108926645.

Anna Scampicchio, Elena Arcari, Amon Lahr, and Melanie N. Zeilinger. Gaussian processes for dynamics learning in model predictive control. *Annual Reviews in Control*, 60:101034, 2025. doi: 10.1016/j.arcontrol.2025.101034.

Arno Solin and Simo Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, 30(2):419–446, 2020. doi: 10.1007/s11222-019-09886-w.

Andreas Svensson and Thomas B. Schön. A flexible state–space model for learning nonlinear dynamical systems. *Automatica*, 80:189–199, 2017. doi: 10.1016/j.automatica.2017.02.030.

Yusuke Tanaka, Tomoharu Iwata, and naonori ueda. Symplectic Spectrum Gaussian Processes: Learning Hamiltonians from Noisy and Sparse Data. In *Advances in Neural Information Processing Systems*, volume 35, pages 20795–20808. Curran Associates, 2022.

Björn Volkmann, Jan-Hendrik Ewering, Michael Meindl, Simon F. G. Ehlers, Matthias A. Müller, and Thomas Seel. Bayesian Inference and Learning in Nonlinear Dynamical Systems: A Framework for Incorporating Explicit and Implicit Prior Knowledge. *preprint, arXiv: 2508.15345*, 2025. doi: 10.48550/arXiv.2508.15345.

Joe Watson, Chen Song, Oliver Weeger, Theo Gruner, Le Thai an , Kay Hansel, Ahmed Hendawy, Oleg Arenz, Will Trojak, Miles Cranmer, Carlo D'Eramo, Fabian Buelow, Tanmay Goyal, Jan Peters, and Martin W. Hoffmann. Machine Learning with Physics Knowledge for Prediction: A Survey. *Transactions on Machine Learning Research*, 2025.

Anna Wigren, Riccardo Sven Risuleo, Lawrence Murray, and Fredrik Lindsten. Parameter elimination in particle Gibbs sampling. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, 2019.

Anna Wigren, Johan Wågberg, Fredrik Lindsten, Adrian G. Wills, and Thomas B. Schön. Nonlinear System Identification: Learning While Respecting Physical Models Using a Sequential Monte Carlo Method. *IEEE Control Systems*, 42(1):75–102, 2022. doi: 10.1109/MCS.2021.3122269.

Martin Ziegler, Andres Felipe Posada-Moreno, Friedrich Solowjow, and Sebastian Trimpe. On Foundation Models for Dynamical Systems from Purely Synthetic Data. *preprint. arXiv: 2412.00395*, 2024. doi: 10.48550/arXiv.2412.00395.

Supplementary Material

A. Proposed Model in the Canonical Form of the Restricted Exponential Family

To derive the distributions employed in Section 5, we express all densities in the canonical form of the restricted exponential family. This is convenient for Bayesian inference in state-space models, as probabilistic dependence on the previous state can be incorporated (Wigren et al., 2019).

Formulation The proposed model (10) can be written in the canonical form of the restricted exponential family as

$$p(\boldsymbol{\alpha}) = Z \exp\left(\sum_{i=1}^{2} \boldsymbol{\alpha}_i^\top \boldsymbol{s}_i - \operatorname{Tr}\left(\boldsymbol{P}_i^\top(\boldsymbol{\alpha})\boldsymbol{r}_i\right)\right), \tag{20}$$

where the density is expressed in terms of the *natural parameters* $\boldsymbol{\alpha}$ that are obtained by transforming the original model parameters $\boldsymbol{\theta} = \{\boldsymbol{a}, \sigma^2\}$. We have $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1^\top, \alpha_2]^\top$ with

$$\boldsymbol{\alpha}_1 = \frac{\boldsymbol{a}}{\sigma^2}, \qquad \alpha_2 = -\frac{1}{2\sigma^2}, \tag{21}$$

and the parameter-dependent parts of the log-partition function being

$$\boldsymbol{P}_1(\boldsymbol{\alpha}) = -\frac{1}{4}\boldsymbol{\alpha}_1\alpha_2^{-1}\boldsymbol{\alpha}_1^\top = \frac{1}{2\sigma^2}\boldsymbol{a}\boldsymbol{a}^\top, \qquad P_2(\boldsymbol{\alpha}) = -\frac{1}{2}\log|-2\alpha_2| = \frac{1}{2}\log\sigma^2, \tag{22}$$

where we denote $|\cdot|$ as the determinant. The remaining variables, Z, $\boldsymbol{s}_i$, and $\boldsymbol{r}_i$—depending on the employed prior and the data—are a normalization constant and statistics, respectively.

Likelihood To express the likelihood

$$\begin{aligned} p(\boldsymbol{z}_{0:T}|\boldsymbol{\theta}, \vartheta_{\mathrm{S}}) &= p(\boldsymbol{z}_0) \cdot \prod_{t=1}^{T} p(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1}, \boldsymbol{\theta}, \vartheta_{\mathrm{S}}) \\ &= p(\boldsymbol{x}_0)p(\boldsymbol{h}_0 \mid \boldsymbol{x}_0, \boldsymbol{\theta}) \cdot \prod_{t=1}^{T} p(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1}, \boldsymbol{\theta}, \vartheta_{\mathrm{S}}), \end{aligned} \tag{23}$$

in the form (20), we assume, for simplicity, that $p(\boldsymbol{x}_0)$ is known and consider each time step separately. For individual transitions $p(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1}, \boldsymbol{\theta}, \vartheta_{\mathrm{S}})$ we have

$$\begin{aligned} p(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1}, \boldsymbol{\theta}, \vartheta_{\mathrm{S}}) &= p(\boldsymbol{h}_t \mid \boldsymbol{x}_t, \boldsymbol{\theta})\, p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1}, \vartheta_{\mathrm{S}}) \\ &= \mathcal{N}(\boldsymbol{h}_t \mid \boldsymbol{D}_{\boldsymbol{\phi}}(\boldsymbol{x}_t)\boldsymbol{a}, \sigma^2\mathbf{I})\, \mathcal{N}(\boldsymbol{x}_t \mid \boldsymbol{f}(\boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1}, \boldsymbol{u}_{t-1}, \vartheta_{\mathrm{S}}), \boldsymbol{\Sigma}_w), \end{aligned} \tag{24}$$

were the function $\boldsymbol{f}$ and the noise covariance $\boldsymbol{\Sigma}_w$ describe the discrete-time state dynamics, and can be obtained from integrating (10). Noting that only the first density is $\boldsymbol{\theta}$-dependent, the likelihood for one time step can be expressed as

$$\begin{aligned} p(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1}, \boldsymbol{\theta}, \vartheta_{\mathrm{S}}) &= \tilde{b}_t \cdot \mathcal{N}(\boldsymbol{h}_t \mid \boldsymbol{D}_{\boldsymbol{\phi}}(\boldsymbol{x}_t)\boldsymbol{a}, \sigma^2\mathbf{I}) \\ &= \frac{\tilde{b}_t}{(2\pi)^{n_x/2}(\sigma^2)^{n_x/2}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{h}_t - \boldsymbol{D}_{\boldsymbol{\phi}}(\boldsymbol{x}_t)\boldsymbol{a})^\top(\boldsymbol{h}_t - \boldsymbol{D}_{\boldsymbol{\phi}}(\boldsymbol{x}_t)\boldsymbol{a})\right) \\ &= b_t \exp\left(\sum_{i=1}^{2} \boldsymbol{\alpha}_i^\top \boldsymbol{s}_i(\boldsymbol{z}_t) - \operatorname{Tr}\left(\boldsymbol{P}_i^\top(\boldsymbol{\alpha})\boldsymbol{r}_i(\boldsymbol{x}_t)\right)\right), \end{aligned} \tag{25}$$

with the statistics

$$\begin{aligned} \boldsymbol{s}_1(\boldsymbol{z}_t) &= \boldsymbol{D}_{\boldsymbol{\phi}}(\boldsymbol{x}_t)^\top \boldsymbol{h}_t\,, & s_2(\boldsymbol{z}_t) &= \boldsymbol{h}_t^\top \boldsymbol{h}_t\,, \\ \boldsymbol{r}_1(\boldsymbol{x}_t) &= \boldsymbol{D}_{\boldsymbol{\phi}}(\boldsymbol{x}_t)^\top \boldsymbol{D}_{\boldsymbol{\phi}}(\boldsymbol{x}_t)\,, & r_2(\boldsymbol{x}_t) &= n_x\,, \end{aligned} \tag{26}$$

and the base measures

$$b_t = \begin{cases} (2\pi)^{-n_x/2} \cdot p(\boldsymbol{x}_0)\,, & \text{if } t = 0\,, \\ (2\pi)^{-n_x/2} \cdot \mathcal{N}(\boldsymbol{x}_t \mid \boldsymbol{f}(\boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1}, \boldsymbol{u}_{t-1}, \boldsymbol{\vartheta}_{\mathrm{S}}), \boldsymbol{\Sigma}_w)\,, & \text{otherwise}\,. \end{cases} \tag{27}$$

The overall likelihood for data $\boldsymbol{z}_{0:T}$ is constructed by multiplying the one-step densities, which amounts to summing the corresponding trajectory statistics (26).

Prior and Posterior To express the multivariate normal inverse Gamma ($\mathcal{NIG}$) (Murphy, 2007) prior and posterior densities in the form (20), we first define the normal distribution

$$\begin{aligned} p(\boldsymbol{a} \mid \boldsymbol{m}, \boldsymbol{V}, \sigma^2) &= \mathcal{N}(\boldsymbol{a} \mid \boldsymbol{m}, \sigma^2 \boldsymbol{V}) \\ &= \frac{1}{(2\pi)^{M/2}(\sigma^2)^{M/2}|\boldsymbol{V}|^{1/2}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{a}-\boldsymbol{m})^\top \boldsymbol{V}^{-1}(\boldsymbol{a}-\boldsymbol{m})\right)\,, \end{aligned} \tag{28}$$

and inverse Gamma ($\mathcal{IG}$) distribution

$$p(\sigma^2 \mid \psi, \nu) = \mathcal{IG}(\sigma^2 \mid \psi, \nu) = \frac{(\psi/2)^{\nu/2}}{\Gamma\left(\frac{\nu}{2}\right)} (\sigma^2)^{-\nu/2-1} \exp\left(-\frac{1}{2\sigma^2}\psi\right)\,, \tag{29}$$

where Γ is the scalar Gamma function. Given this, the prior and posterior densities can be expressed in the canonical form of the restricted exponential family as

$$\begin{aligned} p(\boldsymbol{a}, \sigma^2 \mid \boldsymbol{m}, \boldsymbol{V}, \psi, \nu) &= \mathcal{NIG}(\boldsymbol{a} \mid \boldsymbol{m}, \boldsymbol{V}, \psi, \nu) \\ &= \frac{(\psi/2)^{\nu/2}\left(\frac{1}{\sigma^2}\right)^{\nu/2+1+M/2}}{(2\pi)^{M/2}|\boldsymbol{V}|^{1/2}\Gamma\left(\frac{\nu}{2}\right)} \exp\left(-\frac{1}{2\sigma^2}\left[\psi + (\boldsymbol{a}-\boldsymbol{m})^\top \boldsymbol{V}^{-1}(\boldsymbol{a}-\boldsymbol{m})\right]\right) \\ &= n(\boldsymbol{\eta}) \exp\left(\sum_{i=1}^{2} \boldsymbol{\alpha}_i^\top \tilde{\boldsymbol{s}}_i(\boldsymbol{\eta}) - \mathrm{Tr}\left(\boldsymbol{P}_i^\top(\boldsymbol{\alpha})\tilde{\boldsymbol{r}}_i(\boldsymbol{\eta})\right)\right)\,, \end{aligned} \tag{30}$$

where the statistics, dependent on the distribution parameters $\boldsymbol{\eta} = \{\boldsymbol{m}, \boldsymbol{V}, \psi, \nu\}$, are

$$\begin{aligned} \tilde{\boldsymbol{s}}_1(\boldsymbol{\eta}) &= \boldsymbol{V}^{-1}\boldsymbol{m}\,, & \tilde{s}_2(\boldsymbol{\eta}) &= \psi + \boldsymbol{m}^\top \boldsymbol{V}^{-1}\boldsymbol{m}\,, \\ \tilde{\boldsymbol{r}}_1(\boldsymbol{\eta}) &= \boldsymbol{V}^{-1}\,, & \tilde{r}_2(\boldsymbol{\eta}) &= \nu + 2 + M\,, \end{aligned} \tag{31}$$

and the normalizing factor

$$n(\boldsymbol{\eta}) = \frac{(\psi/2)^{\nu/2}}{(2\pi)^{M/2}|\boldsymbol{V}|^{1/2}\Gamma\left(\frac{\nu}{2}\right)}\,. \tag{32}$$

The parameter posterior $p(\boldsymbol{\theta}|\boldsymbol{z}_{0:T}, \boldsymbol{\vartheta})$ is available in closed form by summation of the prior statistics and new statistics, obtained from the data trajectories $\boldsymbol{z}_{0:T}$, that is $\boldsymbol{s}_i^+ = \tilde{\boldsymbol{s}}_i(\boldsymbol{\eta}) + \sum_{t=0}^{T} \boldsymbol{s}_i(\boldsymbol{z}_t)$ and $\boldsymbol{r}_i^+ = \tilde{\boldsymbol{r}}_i(\boldsymbol{\eta}) + \sum_{t=0}^{T} \boldsymbol{r}_i(\boldsymbol{x}_t)$. The resulting parameter posterior density is

$$p(\boldsymbol{\theta}|\boldsymbol{z}_{0:T}, \boldsymbol{\vartheta}_{\mathrm{K}}) = \mathcal{NIG}(\boldsymbol{a}, \sigma^2 | \boldsymbol{m}^+, \boldsymbol{V}^+, \psi^+, \nu^+)\,, \tag{33}$$

with the new distribution parameters $\boldsymbol{\eta}^+ = \{\boldsymbol{m}^+, \boldsymbol{V}^+, \psi^+, \nu^+\}$ being

$$\begin{aligned} \boldsymbol{m}^+ &= \left(\boldsymbol{r}_1^+\right)^{-1} \boldsymbol{s}_1^+\,, & \boldsymbol{V}^+ &= \left(\boldsymbol{r}_1^+\right)^{-1}\,, \\ \psi^+ &= s_2^+ - \boldsymbol{s}_1^{+\top}\left(\boldsymbol{r}_1^+\right)^{-1}\boldsymbol{s}_1^+\,, & \nu^+ &= r_2^+\,. \end{aligned} \tag{34}$$

B. Simulation Case Study

Simulation Setup We conduct a simulation case study with a non-harmonic oscillator, governed by the Hamiltonian function

$$H\left(\begin{bmatrix} q \\ p \end{bmatrix}\right) = \frac{q^2}{2} + \frac{p^2}{2} + 2\cos q\,, \tag{35}$$

with position q and momentum p (see Figure 2). The system is driven by a known input force u and dissipates energy through a damping coefficient d, to be estimated. We consider the system

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x} = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} q \\ p \end{bmatrix} &= \left(\underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{\boldsymbol{J}} - \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & d \end{bmatrix}}_{\boldsymbol{R}}\right)\nabla_{\boldsymbol{x}} H(\boldsymbol{x}) + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\boldsymbol{G}} u + \boldsymbol{w}\,, \\ \boldsymbol{y}^{(\mathrm{i/s})} &= \boldsymbol{x} + \boldsymbol{e}^{(\mathrm{i/s})}\,, \\ y^{(\mathrm{i/o})} &= q + e^{(\mathrm{i/o})}\,, \end{aligned} \tag{36}$$

with damping coefficient $d = 0.15$. The zero-mean discrete-time process and measurement noise terms are Gaussian with standard deviation 10^{-4} and 10^{-3}, respectively. To generate training data, we simulate the system from the initial value $\boldsymbol{x}_0 = [0, 0]^\top$, using the input signal illustrated in Figure 4, for $T = 1,000$ time steps with a symplectic Euler integrator at step size $\delta = 0.02\,\mathrm{s}$.

Inference and Learning To perform Bayesian inference and learning, we run Algorithm 1 for $K = 20,000$ iterations. We discard the first $10,000$ samples to ignore the burn-in period of the Markov chain. For discretizing the state dynamics (10) in Algorithm 1, we use Euler integration. To generate proposals in the Metropolis-within-Gibbs steps, a random walk is employed for the kernel hyperparameters $\boldsymbol{\vartheta}_{\mathrm{K}}$. For the structural hyperparameter $\vartheta_{\mathrm{S}} := d$, we use refined proposals based on the gradient and Hessian of the likelihood (Roberts and Stramer, 2002). The chosen parameters are summarized in Table 2.

Table 2: Parameters of Algorithm 1 in the Case Study

Name	Symbol	Value
Number of eigenfunctions	M	20 without symmetry constraint 15 with symmetry constraint
Domain bounds	L_1, L_2	8, 8
Hyper-prior	$p(\boldsymbol{\vartheta})$	near uniform (broad Gaussian)
Scale	ψ	100
Degrees of freedom	ν	400

Testing For testing in Figure 5, we use the samples $\{\boldsymbol{\theta}[k], \vartheta_{\mathrm{S}}[k]\}$, $k = 10,000, \ldots, 20,000$, provided by Algorithm 1 to *(i)* generate a mean model and *(ii)* draw ten random instances, each representing a sample from the posterior model distribution. Using these models, we perform forward-predictions with a symplectic Euler integration scheme at step size $\delta = 0.01\,\mathrm{s}$. We simulate from the initial value $\boldsymbol{x}_0 = [-0.1, 0.5]^\top$ and apply a pre-set test input signal (depicted in Figure 5), different from the training input signal.

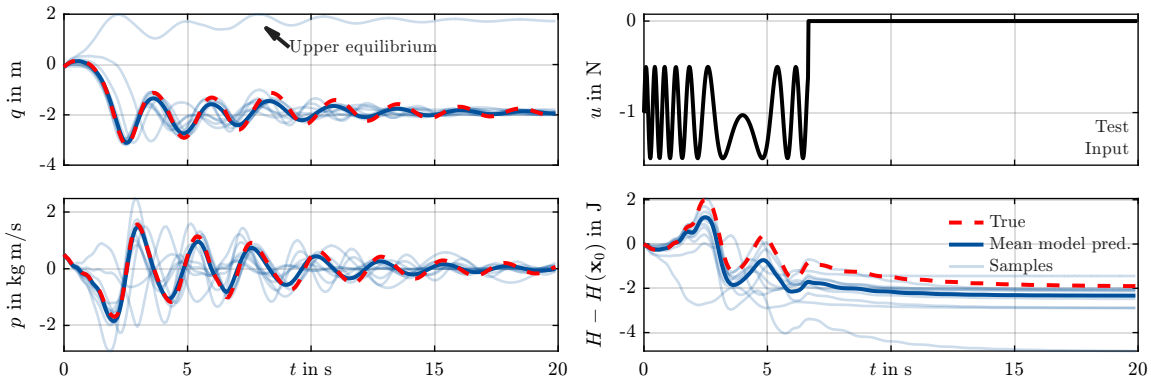

Figure 5: True system behavior and forward predictions of the learned Hamiltonian GP model. The proposed method provides a probabilistic Hamiltonian system, and—despite learning only from input-output data—*each* sampled model yields a physically consistent prediction that resembles the actual system behavior.

C. Convergence Metrics

For the simulation case study in Section 6, we evaluate the convergence of the PMCMC scheme in Algorithm 1. Specifically, we present the autocorrelation plots of individual states and parameters in Figure 6, which indicate the degree of correlation between successive draws from the sampler. A quickly decreasing autocorrelation indicates good "mixing" and, thus, an efficient exploration of the domain (Wigren et al., 2022).

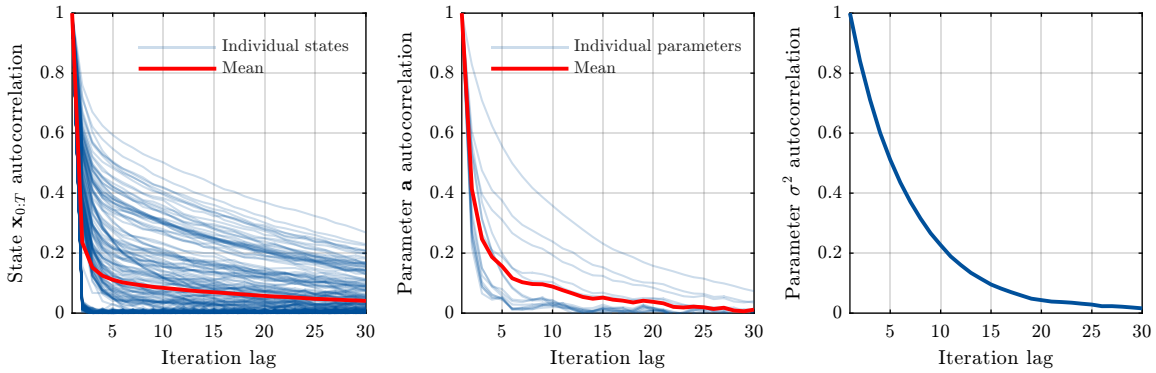

Figure 6: Autocorrelation of the state and parameter samples. In the plots, each state time step and each state/parameter dimension are represented by individual autocorrelation graphs.

Similarly, the update rate along the time axis provides a measure of how regularly state time step samples $\boldsymbol{x}_t[k]$ are updated throughout the sampling procedure. Ideally, the update rate should be close to 1 for each time step, which is the case in the present simulation example (see Figure 7).

The RMSE between the true measurements and the estimated output variables is given in the right plot of Figure 7. Although we discard the initial $10,000$ samples as burn-in, we see a significantly faster convergence after at most $1,000$ iterations.

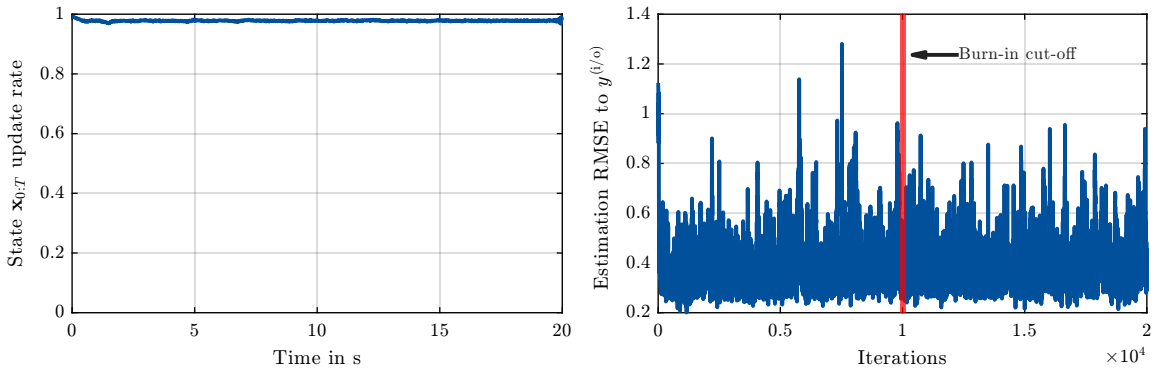

Figure 7: Update rate of the state samples (left) and RMSE between the estimated positions and the true position measurements $y^{(\mathrm{i/o})}$ (right).

D. Additional Simulation Results

Due to identifiability issues in the input-output setting, we impose a symmetry constraint on the Hamiltonian GP model (5). To this end, an advantage of the employed reduced-rank GP is that prior knowledge about the target function, such as symmetry, can be easily encoded in the basis function expansion (Volkmann et al., 2025). To impose the symmetry constraint, we follow the lines of Berntorp and Menner (2022) and index basis functions in (4) by selecting only odd indices $j_{k,i} = 1, 3, \ldots$ when constructing the reduced-rank GP. This ensures that the approximated Hamiltonian satisfies $\widehat{H}(\boldsymbol{x}) = \widehat{H}(-\boldsymbol{x})$, which induces the anti-symmetry $\nabla_{\boldsymbol{x}}\widehat{H}(\boldsymbol{x}) = -\nabla_{\boldsymbol{x}}\widehat{H}(-\boldsymbol{x})$ for its gradient. In Figure 8, we compare the proposed method in three different settings, i. e., learning with

- state measurements $\{\boldsymbol{y}_t^{(\mathrm{i/s})}\}_{t=0}^{T}$, without symmetry constraint (left column),
- output measurements $\{y_t^{(\mathrm{i/o})}\}_{t=0}^{T}$, with symmetry constraint (middle column), and
- output measurements $\{y_t^{(\mathrm{i/o})}\}_{t=0}^{T}$, without symmetry constraint (right column).

All approaches correctly learn the equilibrium positions along the q-dimension. In the input-output setting, i. e., without measurements of the momentum p available, learning along the p-dimension can result in ambiguous solutions, which we attribute to identifiability problems. In the present nonlinear simulation example, the symmetry constraint can facilitate learning from input-output data.

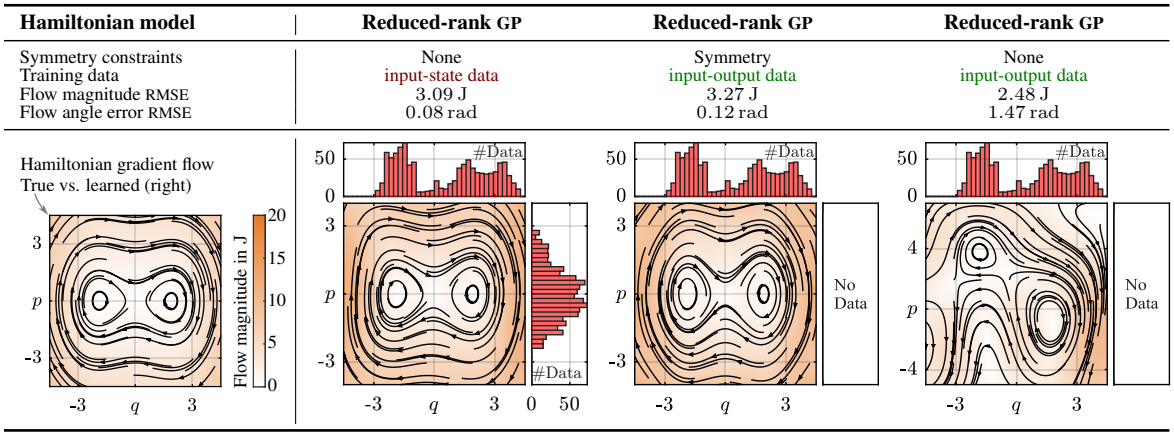

Figure 8: Flow maps following from the true and learned Hamiltonians of the non-harmonic oscillator system.