# Structure-preserving Gaussian Process Dynamics

Katharina Ensinger[1,2] [0000−0001−7315−093X] ✉, Friedrich Solowjow[2] [0000−0003−2623−5652], Sebastian Ziesche[1], Michael Tiemann[1] [0000−0003−3454−8472], and Sebastian Trimpe[2] [0000−0002−2785−2487]

[1] Bosch Center for Artificial Intelligence, Renningen, Germany
[2] Institute for Data Science in Mechanical Engineering, RWTH Aachen University, Aachen, Germany
katharina.ensinger@bosch.com

**Abstract.** Most physical processes possess structural properties such as constant energies, volumes, and other invariants over time. When learning models of such dynamical systems, it is critical to respect these invariants to ensure accurate predictions and physically meaningful behavior. Strikingly, state-of-the-art methods in Gaussian process (GP) dynamics model learning are not addressing this issue. On the other hand, classical numerical integrators are specifically designed to preserve these crucial properties through time. We propose to combine the advantages of GPs as function approximators with structure-preserving numerical integrators for dynamical systems, such as Runge-Kutta methods. These integrators assume access to the ground truth dynamics and require evaluations of intermediate and future time steps that are unknown in a learning-based scenario. This makes direct inference of the GP dynamics, with embedded numerical scheme, intractable. As our key technical contribution, we enable inference through the implicitly defined Runge-Kutta transition probability. In a nutshell, we introduce an implicit layer for GP regression, which is embedded into a variational inference model learning scheme.

**Keywords:** Gaussian Processes · Bayesian Methods · Time-Series.

## 1 Introduction

Many physical processes can be described by an autonomous continuous-time dynamical system

$$\dot{x}(t) = f(x(t)) \text{ with } f : \mathbb{R}^d \to \mathbb{R}^d. \tag{1}$$

Dynamics model learning deals with the problem of estimating the function $f$ from sampled data. In practice, it is not possible to observe state trajectories in continuous time since data is typically collected on digital sensors and hardware. Thus, we obtain noisy discrete-time observations

$$\{\hat{x}_n\}_{1:N} = \{x_1 + \nu_1, \dots, x_N + \nu_N\}, \nu_n \sim \mathcal{N}(0, \mathrm{diag}(\sigma_{n,1}^2, \dots, \sigma_{n,d}^2)). \tag{2}$$

Accordingly, models of dynamical systems are typically learned as of one-step ahead-predictions

$$x_{n+1} = g(x_n). \tag{3}$$

Especially, Gaussian processes (GPs) have been popular for model learning and are predominantly applied to one step ahead predictions (3) [5, 7, 8]. However, there is a discrepancy between the continuous (1) and discrete-time (3) systems. Importantly, (1) often possesses invariants that represent physical properties. Thus, naively chosen discretizations (such as Eq. (3)) might lead to poor models.

Numerical integrators provide sophisticated tools to efficiently discretize continuous-time dynamics (1). Strikingly, one step ahead predictions (3) correspond to the explicit Euler integrator $x_{n+1} = x_n + hf(x_n)$ with step size $h$. This follows immediately by identifying $g(x_n)$ with $x_n + hf(x_n)$. It is well-known that the explicit Euler method might lead to problematic behavior and suboptimal performance [13]. Clearly, this raises the immediate question: *can superior numerical integrators be leveraged for dynamics model learning?*

For the numerical integration of dynamical systems, the function $f$ is assumed to be known. The explicit Euler is a popular and straightforward method, which thrives due to its simplicity. No intermediate evaluations of the dynamics are necessary, which makes the integrator also attractive for model learning. While this behavior is tempting when implementing the algorithm, there are theoretical issues [13]. In particular, important physical and geometrical structure is not preserved. In contrast to the explicit Euler, there are also implicit and higher-order methods. However, these generalizations require the evaluation at intermediate and future time steps, which leads to a nonlinear system of equations that needs to be solved. While these schemes become more involved, they yield advantageous theoretical guarantees. In particular, Runge-Kutta (RK) schemes define a rich class of powerful integrators. Despite assuming the dynamics function $f$ to be unknown, we can still benefit from the discretization properties of numerical integrators for model learning. To this end, we propose to combine GP dynamics learning with arbitrary RK integrators, in particular, implicit RK integrators.

Depending on the problem that is addressed, the specific RK method has to be tailored to the system. As an example, we consider structure-preserving integrators, i.e., geometric and symplectic ones. We develop our arguments based on Hamiltonian systems [28, 29]. These are an important class of problems that preserve a generalized notion of energy and volume. Symplectic integrators are designed to cope with this type of problems, providing volume-preserving trajectories and accurate approximation of the total energy [12]. In order to demonstrate the flexibility of our method, we also introduce a geometric integrator that is consistent with a mass moving on a surface. For both examples, we show in the experiments section that the predictions with our tailored GP model are indeed preserving the desired structure.

By generalizing to more sophisticated integrators, we have to address the issue of propagating implicitly defined distributions through the dynamics. This is due to the fact that evaluations of the GP at the next time step induce additional implicit evaluations of the dynamics. Depending on the integrator, these might be future or intermediate time steps that are also unknown. On a technical level, sparse GPs provide the necessary flexibility. A decoupled sampling approach allows consistent sampling of functions from the GP posterior [34]. In contrast to

previous GP dynamics modeling schemes [7], this yields consistency throughout the entire simulation pipeline. By leveraging these ideas, we derive a recurrent variational inference (VI) model learning scheme.

By addressing integrator-induced implicit transition probabilities, we are essentially proposing implicit layers for probabilistic models. Implicit layers in neural networks (NNs) are becoming increasingly popular and address outputs that can not be calculated explicitly [3, 10, 23]. Typically, implicit layers in NNs are defined in terms of an optimization problem. However, the idea of implicitly defined layers has (to the best of our knowledge) not yet been generalized to probabilistic models like GPs, introducing the technical challenge of dealing with implicitly defined probability distributions.

In summary, the main contributions of this paper are:

- a general and flexible probabilistic learning framework that combines arbitrary RK integrators with GP dynamics model learning;
- an inference scheme that is able to cope with implicitly defined distributions, thus extending the idea of implicit layers from NNs to GPs; and
- embedding geometrical and symplectic integrators, yielding GP dynamics models that are structure-preserving.

## 2 Related work

Dynamics model learning is a very broad field and has been addressed by various communities for decades, e.g., [9, 22]. Learning GP dynamics models can be addressed with a parameteric or non-parametric continuous-time GP model. Nonparametric continuous-time GP models were learned by applying sparse GPs [15, 16]. A common approach for learning discrete-time models are GP state-space models [32, 33]. In this work, we consider fully observable systems in contrast to common state-space models. However, we apply the tools of state-space model literature. In particular, we develop our ideas exemplary for the inference scheme proposed in [7]. At the same time, our contribution is not restricted to that choice of inference scheme and can be combined with other schemes as [19] as well. In constrast to [7], we sample consistent GPs from the posterior (cf. Sec. 4.)

Implicit transitions have become popular for NNs and provide useful tools that we leverage. Implicit layers in neural networks denote layers, in which the output is defined in terms of an optimization problem [10]. On a technical level, we implement related techniques based on the implicit function theorem and backpropagation. A NN with infinitely many layers was trained by implicitly defining the equilibrium of the network [3]. Look et al. [23] propose an efficient backpropagation scheme for implicitly defined neural network layers. We extend these approaches from deterministic NNs to probabilistic GP models.

Including Hamiltonian structure into learning in order to preserve physical properties of the system, is an important problem addressed by many sides. The problem can be tackled by approximating the continuous-time Hamiltonian system from data. This was addressed by applying a NN [11]. Since modern NN approaches provide a challenging benchmark, we compare our method against

[11]. In [20], the Hamiltonian structure is learned by stacking multiple symplectic modules. GPs have been combined with symplectic structure as well [4, 26]. In contrast to our approach, the focus lies on learning continuous-time dynamics with direct GP inference and afterwards unrolling the dynamics via certain symplectic or structure-preserving integrators. By learning and predicting the identical discrete-time system, we omit an additional numerical error and predictions are computational more efficient.

However, there is literature that addresses discrete-time Hamiltonian systems. The Hamiltonian neural network approach was extended by a recurrent NN coupled with a symplectic integrator [6]. In [27], the symplectic learning approach was extended to variational autoencoders, adding uncertainty to the initial value. Zhong et al. [35] extend previous approaches by adding control input. In contrast to previous approaches, we are able to apply implicit integrators. This allows us to address non-separable Hamiltonians and geometrical invariants. Further, our GP approach allows to sample trajectories from a structure-preserving distribution.

## 3 Technical background and main idea

Next, we make our problem precise and provide a summary of the preliminaries.

### 3.1 Gaussian process regression

A GP is a distribution over functions [25]. Similar to a normal distribution, a GP is determined by its mean function $m(x)$ and covariance function $k(x, y)$. We assume the prior mean to be zero.

**Standard GP inference:** For direct training, the GP predictive distribution is obtained by conditioning on $n$ observed data points. In addition to optimizing the hyperparameters, a system of equations has to be solved, which has a complexity of $\mathcal{O}(n^3)$. Clearly, this is problematic for large datasets.

**Variational sparse GP:** The GP can be sparsified by introducing pseudo inputs [31]. Intuitively, we approximate the posterior with a lower number of training points. An elegant approximation strategy is based on casting Bayesian inference as an optimization problem. We consider pseudo inputs $\xi = [\xi_1, \dots, \xi_P]$ and targets $z = [z_1, \dots, z_P]$ as proposed in [17] and applied in [7, 19]. Intuitively, the targets can be interpreted as GP observations at $\xi$. The posterior of pseudo targets is approximated via a variational approximation $q(z) = \mathcal{N}(\mu, \Sigma)$, where $\mu$ and $\Sigma$ are adapted during training. The GP posterior distribution at inputs $x^*$ is conditioned on the pseudo inputs and targets resulting in a normal distribution $f(x^*|z, \xi) \sim \mathcal{N}(\mu(x^*), \Sigma(x^*))$ with

$$\mu(x^*) = k(x^\star, \xi)k(\xi, \xi)^{-1}z, \text{ and } \Sigma(x^\star) = k(x^\star, x^\star) - k(x^\star, \xi)k(\xi, \xi)^{-1}k(\xi, x^\star). \tag{4}$$

**Decoupled sampling:** In contrast to standard (sparse) GP conditioning (4) this approach allows to sample functions from the posterior that can be evaluated at arbitrary locations [34]. Thus, iterative sampling at multiple inputs is achieved

without conditioning on previous function evaluations. By applying Matheron's rule [18], the GP posterior is decomposed into two parts,

$$\begin{aligned} f(x^\star|z,\xi) &= \underbrace{f(x^\star)}_{\text{prior}} + \underbrace{k(x^\star,\xi)k(\xi,\xi)^{-1}(z-f_z)}_{\text{update}} \\ &\approx \sum_{i=1}^{S} w_i\phi_i(x^\star) + \sum_{j=1}^{P} v_j k(x^\star,\xi_j), \end{aligned} \tag{5}$$

where $S$ Fourier bases $\phi_i$ and $w_i \sim \mathcal{N}(0,1)$ represent the stationary GP prior [24]. For the update in Eq. (5) it holds that $v = k(\xi,\xi)^{-1}(z - \Phi w)$ with feature matrix $\Phi = \phi(\xi) \in \mathbb{R}^{P\times S}$ and weight vector $w \in \mathbb{R}^S$. The targets $z$ are sampled from the variational distribution $q(z)$ at the inputs $\xi$. We add technical details and details on the Fourier bases in our setting in the supplementary material.

### 3.2 Runge-Kutta integrators

A RK integrator $\psi_f$ for a continuous-time dynamical system $f$ (1) is designed to approximate the solution $x(t_n)$ at discrete time steps $t_n$ via $\bar{x}_n$. Hence,

$$\bar{x}_{n+1} = \psi_f(\bar{x}_n) = \bar{x}_n + h\sum_{j=1}^{s} b_j g_j,\; g_j = f(\bar{x}_n + h\sum_{l=1}^{s} a_{jl}g_l), j = 1,\dots,s, \tag{6}$$

where $g_j$ are the internal stages and $\bar{x}_0 = x(0)$. We use the notation $\bar{x}$ to indicate numerical error corrupted states and highlight the subtle difference to ground truth data. The parameters $a_{jl}, b_j \in \mathbb{R}$ determine the properties of the method, e.g., the stability radius of the method [14], the geometrical properties, or whether it is symplectic [12].

**Implicit integrators:** If $a_{jl} > 0$ for $l \geq j$, Eq. (6) takes evaluations at time steps into account where the state is not yet known. Therefore, the solution of a nonlinear system of equations using a numerical solver is required. A prominent example is the implicit Euler scheme $\bar{x}_{n+1} = \bar{x}_n + hf(\bar{x}_{n+1})$.

### 3.3 Main idea

We propose to embed RK methods (6) into GP regression. Since the underlying ground truth dynamics $f$ (1) are given in continuous time, the discretization matters. Naive methods, such as the explicit Euler method, are known to be inconsistent with physical behavior. Therefore, we investigate how to learn more sophisticated models that, by design, are able to preserve physical structure of the original system. Further, we will develop this idea into a tractable inference scheme. In a nutshell, we learn GP dynamics $\hat{f}$ that yield predictions $x_{n+1} = \psi_{\hat{f}}(x_n)$. This enforces the RK (6) instead of explicit Euler (3) structure. Thus, leading to properties like volume preservation. The main technical difficulty lies in making the implicitly defined transition probability $p(\psi_{\hat{f}}(x_n)|x_n)$ tractable.

## 4 Embbedding Runge-Kutta integrators in GP models

Next, we dive into the technical details of merging GPs with RK integrators. We demonstrate how to make the distribution of any (implicit) RK method tractable.

### 4.1 Efficient evaluation of the transition model

At its core, we consider the problem of evaluating implicitly defined distributions of RK integrators $\psi_{\hat{f}}$. To this end, we derive a sampling-based technique. Leveraging decoupled sampling (5) allows to sample a consistent dynamics function from the GP posterior distribution and thus, proper RK integrator steps. The procedure is illustrated in Fig. 1. We model the dynamics $\hat{f}$ via $d$ variational sparse GPs. Let $z \in \mathrm{R}^{P \times d}$ be a sample from the variational posterior $q(z)$ (cf. Sec. 3.1). The probability of an integrator step $p(\psi_{\hat{f}}(x_n)|z, x_n)$ is formally obtained by integrating over all possible GP dynamics $\hat{f}|z$,

$$p(x_{n+1}|z, x_n) = p(\psi_{\hat{f}}(x_n)|z, x_n) = \int_{\hat{f}} p(\psi_{\hat{f}}(x_n)|\hat{f}) p(\hat{f}|z) d\hat{f}. \tag{7}$$

Next, we show how to sample from the distribution $p(\psi_{\hat{f}}(x_n)|z, x_n)$ in Eq. (7). Performing an RK integrator step $\psi_{\hat{f}}(x_n)$ requires the computation of RK stages $g^\star = (g_1^\star, \ldots, g_s^\star)$ (6)

$$g^\star = \arg\min_g \|g - \hat{f}(x_n + hAg)\|^2, \tag{8}$$

with $A = (a_{jl})_{j=1,\ldots s, l=1\ldots j}$ determined by the RK scheme (6). In the explicit case $A$ is a sub-diagonal matrix so $g_j$ can be calculated iteratively. In the implicit case, a minimization problem has to be solved numerically, which requires the iterative evaluation of multiple intermediate function evaluations. In both, the explicit and implicit case, inputs to the dynamics $\hat{f}$ depend on the output of previous dynamics function evaluations. Thus, all evaluations can not directly be drawn from their joint posterior GP distribution. In order to ensure that iterative evaluations of $\hat{f}$ indeed correspond to the identical GP posterior sample, conditioning on prior evaluations is necessary.

In prior work on state-space models, iterative samples at different time-steps were drawn independently, ignoring these correlations [7]. It was shown in [19] that this introduces a non-negligible error in the forward propagation of the probability distribution. In our case, the sampling scheme in [7] would lead to an even larger error since consistency would not be ensured along a single integration step. In the implicit case, this would result in a minimization problem that changes while it is solved numerically. However, naive GP conditioning on prior function evaluations is computationally intractable.

We address the problem by leveraging decoupled sampling [34]. This sampling approach allows to compute a consistent GP dynamics function $\hat{f}|z$ via Eq. (5) before applying the RK scheme. Technically, sampling the dynamics function $\hat{f}$ is

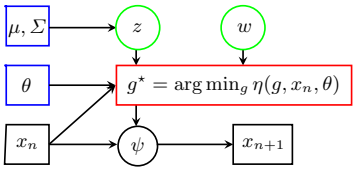



Fig. 1: The evaluation of an integrator step. First, weights $w \sim \mathcal{N}(0,1)$ and inducing targets $z \sim \mathcal{N}(\mu, \Sigma)$ (green) are sampled. This yields tractable solutions to the minimization problem for the RK stages $g^\star$ (red), which yields RK steps $x_{n+1} = \psi(y_n)$. The trainable parameters are marked in blue.

achieved by sampling inducing targets $z$ from $\mathcal{N}(\mu, \Sigma)$ and weights $w \sim \mathcal{N}(0,1)$ (cf. Sec. 3.1). We are now able to evaluate the dynamics $\hat{f}$ at arbitrary locations. This allows to define and solve the system of equations (8) with respect to the fixed, but sampled, GP dynamics $\hat{f}$. Combining Eq. (8) and Eq. (5) with $u = u(g) = x_n + hAg$ yields

$$g^\star = \arg\min_g \left\| g - \sum_{i=1}^{S} w_i \phi_i(u) - \sum_{j=1}^{M} v_j k(u, \xi_j) \right\|^2. \tag{9}$$

Next, we give an example. Consider the IA-Radau method [14] $x_{n+1} = x_n + h\left(\frac{1}{4}g_1 + \frac{3}{4}g_2\right)$, with

$$g_1 = \hat{f}\left(x_n + \frac{h}{4}(g_1 - g_2)\right),\ g_2 = \hat{f}\left(x_n + h\left(\frac{1}{4}g_1 + \frac{5}{12}g_2\right)\right). \tag{10}$$

After sampling $z$ and $w$, the RK scheme (10) is transformed into a minimization problem (9). With $u_1 = x_n + \frac{h}{4}(g_1 - g_2)$, and $u_2 = x_n + h(\frac{1}{4}g_1 + \frac{5}{12}g_2)$ it holds that $\begin{pmatrix} g_1^\star \\ g_2^\star \end{pmatrix} = \arg\min_g F(g)$, with

$$F(g) = \left\| \begin{matrix} g_1 - \sum_{i=1}^{S} w_i \phi_i(u_1) - \sum_{j=1}^{M} v_j k(u_1, \xi_j) \\ g_2 - \sum_{i=1}^{S} w_i \phi_i(u_2) - \sum_{j=1}^{M} v_j k(u_2, \xi_j) \end{matrix} \right\|^2. \tag{11}$$

### 4.2 Application to model learning via variational inference

Next, we construct a variational-inference model learning scheme that is based on the previously introduced numerical integrators. Here, we exemplary develop the integrators for an inference scheme similar to [7] and make the method precise. It is also possible to extend the arguments to other inference schemes such as [19]. In contrast to [7] we sample functions $\hat{f}$ instead of independent draws from the GP dynamics. This allows to produce trajectories that are generated by a fixed but probabilistic vector field. Unlike typical state-space models as [7, 19] we do

not consider transition noise. Thus, the proposed variational posterior is suitable [19]. Structure preservation is in general not possible when adding transition noise to each time step [1, §5].

Factorizing the joint distribution of noisy observations, noise-free states, inducing targets and GP posterior yields

$$p(\hat{x}_{1:N}, x_{1:N}, z, \hat{f}) = \prod_{n=0}^{N-1} p(\hat{x}_{n+1}|x_{n+1}) p(x_{n+1}|x_n, \hat{f}) p(\hat{f}|z) p(z). \tag{12}$$

The posterior distribution $p(x_{1:N}, z, \hat{f}|\hat{x}_{1:N})$ is factorized and approximated by a variational distribution $q(x_{1:N}, z, \hat{f})$. Here, the variational distribution $q$ is chosen

$$q(x_{1:N}, z, \hat{f}) = \prod_{n=0}^{N-1} p(x_{n+1}|x_n, \hat{f}) p(\hat{f}|z) q(z), \tag{13}$$

with the variational distribution $q(z)$ of the inducing targets from Sec. 3.1. The model is adapted by maximizing the Evidence Lower Bound (ELBO)

$$\begin{aligned} \log p(\hat{x}_{1:N}) &\geq \mathbb{E}_{q(x_{1:N}, z, \hat{f})} \left[ \log \frac{p(\hat{x}_{1:N}, x_{1:N}, z, \hat{f})}{q(x_{1:N}, z, \hat{f})} \right] \\ &= \sum_{n=1}^{N} \mathbb{E}_{q(x_{1:N}, z, \hat{f})} \left[ \log p(\hat{x}_n|x_n) \right] - \mathrm{KL}(p(z)||q(z)) =: \mathcal{L}. \end{aligned} \tag{14}$$

Now, the model can be trained by maximizing the ELBO $\mathcal{L}$ (14) with a sampling-based stochastic gradient descent method that optimizes the sparse inputs and hyperparameters. The expectation $\mathbb{E}_{q(x_{1:N}, z, \hat{f})} \left[ \log \frac{p(\hat{x}_{1:N}, x_{1:N}, z, \hat{f})}{q(x_{1:N}, z, \hat{f})} \right]$ is approximated by drawing samples from the variational distribution $q(x_{1:N}, z, \hat{f})$ and evaluating $p(\hat{x}_n|x_n)$ at these samples. Samples from $q$ are drawn by first sampling pseudo targets $z$ and a dynamics function $\hat{f}$ from the GP posterior (5). Trajectories are produced by succesively computing consistent integrator steps $x_{n+1} = \psi_{\hat{f}}(x_n)$ as described in Sec. 4.1. This yields a recurrent learning scheme, by iterating over multiple integration steps. We are able to use our model for predictions by sampling functions from the trained posterior.

### 4.3 Gradients

The ELBO (14) is minimized by applying stochastic gradient descent to the hyperparameters. When conditioning on the sparse GP (4), the hyperparameters include $\theta = (\mu_{1:d}, \Sigma_{1:d}, \theta^{GP}_{1:d})$ with variational sparse GP parameters $\mu_{1:d}$, $\Sigma_{1:d}$ and GP hyperparameters $\theta^{GP}_{1:d}$. The gradient $\frac{dx_{n+1}}{d\theta}$ depends on $\frac{dg^\star}{d\theta}$ and $\frac{dx_n}{d\theta}$ via the integrator (6). It holds that

$$\frac{dg^\star}{d\theta} = \frac{\partial g^\star}{\partial \theta} + \frac{dg^\star}{dx_n} \frac{dx_n}{d\theta}. \tag{15}$$

By the dependence of $x_{n+1}$ on $g^\star$ and of $g^\star$ on $x_n$ (15), the gradient is backpropagated through time. For an explicit integrator, the gradient $\frac{dg^\star}{d\theta}$ can be computed explicitly, since $g_j^\star$ depends on $g_i^\star$ with $i < j$. For implicit solvers, the implicit functions theorem [21] is applied. It holds that $g^\star = \arg\min_g \eta(g, x_n, \theta)$ with the minimization problem $\eta$ derived in Eq. (9). For the gradients of $g^\star$ with respect to $x_n$, respectively $\theta$, it holds with the implicit function theorem [21]

$$\frac{dg^\star}{dx_n} = \left(\frac{\partial^2 \eta}{\partial g^{\star 2}}\right)^{-1} \left(\frac{\partial^2 \eta}{\partial x_n \partial g^\star}\right). \tag{16}$$

## 5 Application to symplectic integrators

In summary, we have first derived how to evaluate the implicitly defined RK distributions. Afterward, we have embedded this technique into a recurrent learning scheme and finally, shown how it is trained. Next, we make the method precise for symplectic integrators and Hamiltonian systems.

### 5.1 Hamiltonian systems and symplectic integrators

An autonomous Hamiltonian system is given by

$$x(t) = \begin{pmatrix} p(t) \\ q(t) \end{pmatrix} \text{ with } \dot{x}(t) = \begin{pmatrix} \dot{p}(t) \\ \dot{q}(t) \end{pmatrix} = \begin{pmatrix} -H_q(p,q) \\ H_p(p,q) \end{pmatrix} \tag{17}$$

and $p, q \in \mathbb{R}^d$. Here, $H_p$ and $H_q$ denote the partial derivatives of $H$ with resepect to $p$ and $q$. In many applications, $q$ corresponds to the state and $p$ to the velocity. The Hamiltonian $H$ often resembles the total energy and is constant along trajectories. The flow of Hamiltonian systems $\psi_t$ is volume preserving in the sense of $\mathrm{vol}(\psi_t(\Omega)) = \mathrm{vol}(\Omega)$ for each bounded open set $\Omega$. The flow $\psi_t$ describes the solution at time point $t$ for the initial values $x_0 \in \Omega$.

Symplectic integrators are volume preserving for Hamiltonian systems (17) [12]. Thus, $\mathrm{vol}(\Omega) = \mathrm{vol}(\psi_f(\Omega))$ for each bounded $\Omega$. Further, they provide a more accurate approximation of the total energy than standard integrators [12]. When designing the GP, it is critical to respect the Hamiltonian structure (17). Additionally, the symplectic integrator ensures that the volume is preserved.

### 5.2 Explicit symplectic integrators

A broad class of real world systems can be modeled by separable Hamiltonians $H(p,q) = T(p) + V(q)$. For example, ideal pendulums and the two body problem. Then, for the dynamical system it holds that

$$\dot{p}(t) = -V'(q),\ \dot{q}(t) = T'(p), \tag{18}$$

with $V : \mathbb{R}^d \to \mathbb{R}^d$ and $T : \mathbb{R}^d \to \mathbb{R}^d$. For this class of problems, explicit symplectic integrators can be constructed. In order to ensure Hamiltonian structure, $V_1'(q), \dots, V_d'(q)$ and $T_1'(p), \dots, T_d'(p)$ are modeled with independent sparse GPs. Symplecticity is enforced via discretizing with a symplectic integrator.

Consider for example the explicit symplectic Euler method

$$p_{n+1} = p_n - hV'(q_n),\ q_{n+1} = q_n + hT'(p_{n+1}). \tag{19}$$

The symplectic Euler method (19) is a partitioned RK method, meaning that different schemes are applied to different dimensions. Here, the explicit Euler method is applied to $p_n$ and the implicit Euler method to $q_n$. The integrator (19) is embedded into the sampling scheme (cf. Sec. 4.1) by sampling from $V'$ and $T'$ and the scheme can readily be embedded into the inference scheme (cf. Sec. 4.2).

### 5.3 General symplectic integrators

The general Hamiltonian system (17) requires the application of an implicit symplectic integrator. An example for a symplectic integrator is the midpoint rule applied to (17)

$$x_{n+1} = x_n + hJ^{-1}\nabla H\left(\frac{x_n + x_{n+1}}{2}\right),\ \text{with } J^{-1} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{20}$$

Again, it is critical to embed the Hamiltonian structure into the dynamics model by modeling $H$ with a sparse GP. Sampling from (20) requires evaluating the gradient $\nabla H$, which is again a GP [25].

## 6 Experiments

In this section, we validate our method numerically. In particular, we i) demonstrate volume-preserving predictions for Hamiltonian systems and the satisfaction of a quadratic geometric constraint for a mechanical system; ii) show that we achieve higher or equal accuracy as state-of-the-art methods; and iii) illustrate that our method can easily deal with different choices of RK integrators.

### 6.1 Methods

We construct our structure-preserving GP model (SGPD) by tailoring the RK integrator to the underlying problem. We compare with the following methods:
**Hamiltonian neural network (HNN)** [11]: Deep learning approach that is tailored to respect Hamiltonian structure.
**Consistent PR-SSM model (Euler)** [7]: Standard variational GP model that corresponds to explicit Euler discretizations. Therefore, we refer to it in the following as Euler. In general all common GP state-space models correspond to the Euler discretization. Here, we use a model similar to [7], but in contrast to [7] we compute consistent predictions via decoupled sampling as discussed in Sec. 4. The general framework is more flexible and can also cope with lower-dimensional state observations. Here, we consider the special case, where we assume noisy state measurements.

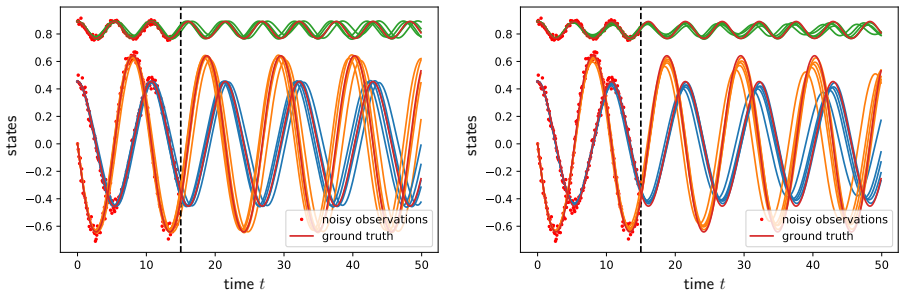



Fig. 2: State trajectory samples for the rigid body dynamics with SGPD (left) and Euler (right). Shown are 5 rollouts from the GP posterior. Both systems are illustrated as a function over time. The training horizon is marked with dotted lines. SGPD visibly enforces structure in contrast to Euler.

### 6.2 Learning task and comparison

In the following, we describe the common setup of all experiments. For each Hamiltonian system, we consider at least one period of a trajectory as training data and all methods are provided with identical training data. For all experiments, we choose the ARD kernel [25]. We apply the training procedure described in Sec. 4.2 on subtrajectories and perform predictions via sampling of trajectories. Since we observed too much influence of the KL-divergence on the ELBO, we include a scaling factor inspired by [2]. In order to draw a fair comparison, we choose similar hyperparameters and number of inducing inputs for our SGPD method and the standard Euler discretization. Details are moved to the appendix. In contrast to our method, the HNN requires additional derivative information, either analytical or as finite differences. Here, we assume that analytical derivative information is not available and thus compute finite differences.

We consider a twofold goal: invariant preservation and accurate predictions in the $L^2$ sense. Predictions are performed by unrolling the first training point over multiple periods of the system trajectory. The $L^2$-error is computed via averaging 5 independent samples from the GP posterior $\hat{X}_i = \frac{1}{5}\sum_{j=1}^{5} \hat{X}_i^j$ and computing $\sqrt{\frac{1}{N}\sum_{i=1}^{N} \|X_i - \hat{X}_i\|^2}$ with ground truth $X_{1:N}$. Integrators are volume-preserving if and only if they are divergence free, which requires $\det(\psi') = 1$ [12]. Thus, we evaluate $\det(\psi')$ for the rollouts, which is intractable for the Hamiltonian neural networks. We observed that we could achieve similar results by propagating the GP mean in terms of constraint satisfaction and $L^2$-error. Here, we focus on trajectory samples in order to highlight that our approach allows to sample structure-preserving rollouts from a structure-preserving GP distribution.

Code will be published upon request.

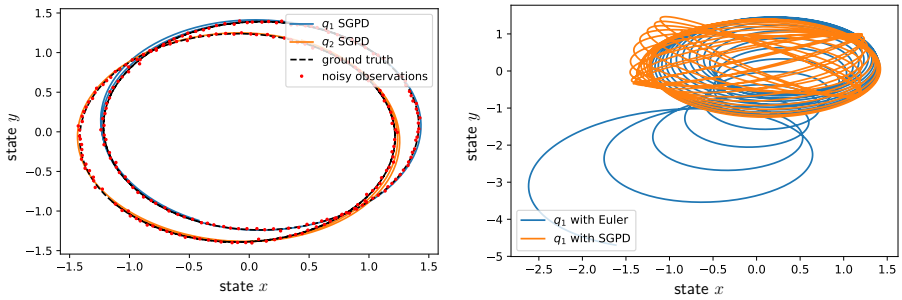



Fig. 3: State trajectory samples for the two-body problem with SGPD (left) and long-term behavior of SGPD and Euler (right). The two-body problem is represented as a phase plot in the two-dimensional space. Shown are single rollouts. The divergent behavior of Euler is clearly visible (right).

Table 1: Shown are the total $L^2$-errors in 1a and an analysis of the total energy for the non-separable system 1b.

(a) $L^2$-err. (mean (std) over 5 indep. runs)

| task | SGPD | Euler | HNN |
|---|---|---|---|
| (i) | **0.421** (0.1) | 0.459 (0.12) | 4.69 (0.02) |
| (ii) | **0.056** (0.01) | **0.057** (0.009) | 0.12 (0.009) |
| (iii) | **0.033** (0.01) | 0.062 (0.04) | **0.035** (0.007) |
| (iv) | **0.046** (0.014) | 0.073 (0.02) | - |

(b) Energy for system iii)

| method | energy err. | std. dev. |
|---|---|---|
| SGPD | $9 \cdot 10^{-4}$ | $2 \cdot 10^{-3}$ |
| Euler | $2 \cdot 10^{-3}$ | $4 \cdot 10^{-3}$ |
| HNN | $9 \cdot 10^{-3}$ | $7 \cdot 10^{-5}$ |

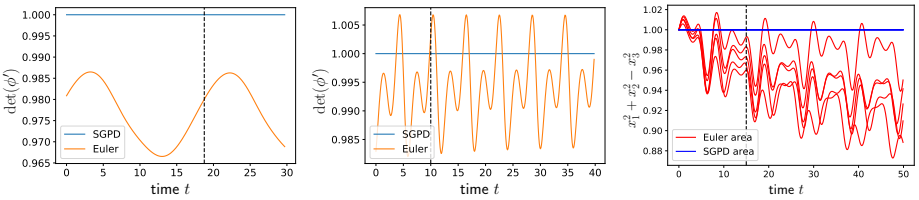
### 6.3 Systems and integrators

We consider four different systems here i) ideal pendulum; ii) two-body problem; iii) non-separable Hamiltonian; and iv) rigid body dynamics.

**Separable Hamiltonians:** the systems i) and ii) are both separable Hamiltonians that are also considered as baseline problems in [11]. Due to their structure, we can apply the symplectic Euler method (cf. Sec. 5.2) to both problems.

The Hamiltonian of a pendulum is given by $H(p,q) = (1 - 6\cos(p)) + \frac{p^2}{2}$. Training data is generated from a 10-second ground truth trajectory with discretization $dt = 0.1$ and disturbed with observation noise with variance $\sigma^2 = 0.1$. Predictions are performed on a 40-second interval.

The two body problem models the interaction of two unit-mass particles $(p_1, q_1)$ and $(p_2, q_2)$, where $p_1, p_2, q_1, q_2 \in \mathbb{R}^2$ and $H(p,q) = \frac{1}{2} + \|p_1\|^2 + \|p_2\|^2 + \frac{1}{\|q_1 - q_2\|^2}$. Noisy training data is generated on an interval of 18.75 seconds, discretization level $dt = 0.15$, and variance $\sigma^2 = 1 \cdot 10^{-3}$. Predictions are performed on an interval of 30 seconds. The orbits of the two bodies $q_1$ and $q_2$, predicted by our SGPD method, are shown in Fig. 3 (left) for a single sample from posterior.




(a) Two-body problem (b) Non-separable system (c) Rigid body dynamics

Fig. 4: The proposed SGPD (blue) preserves structure: The analytic volume is preserved via SGPD for the Hamiltonian systems in 4a and 4b, while simulations show that the explicit Euler (orange) does not preserve volume. Further, the SGPD rollout preserves the quadratic constraint over time for the rigid body system in 4c, while the standard GP with explicit Euler does not.

**Non-separable Hamiltonian:** As an example for a non-separable Hamiltonian system we consider Eq. (17) with $H(p,q) = \frac{1}{2}\left[(q^2+1)(p^2+1)\right]$ [30]. The implicit midpoint rule (20) is applied as the numerical integrator (cf. Sec. 5.3). The training trajectory is generated on a 10-seconds interval with discretization $dt = 0.1$ and disturbed with noise with variance $\sigma^2 = 5 \cdot 10^{-4}$. Rollouts are performed on an interval of 40 seconds.

**Rigid body dynamics:** Consider the rigid body dynamics [12]

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} 0 & \frac{3}{2}x_3 & -x_2 \\ -\frac{3}{2}x_3 & 0 & \frac{x_1}{2} \\ x_2 & -\frac{x_1}{2} & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} =: f(x) \tag{21}$$

that describe the angular momentum of a body rotating around an axis. The equations of motion can be derived via a constrained Hamiltonian system. We apply the implicit midpoint method. Since the HNN is designed for non-constrained Hamiltonians it requires pairs of $p$ and $q$ and is, thus, not applicable. Training data is generated on a 15-seconds interval with discretization $dt = 0.1$. Due to different scales, $x_1$ and $x_2$ are disturbed with noise with variance $\sigma^2 = 1 \cdot 10^{-3}$, and $x_3$ is disturbed with noise with variance $\sigma^2 = 1 \cdot 10^{-4}$. Predictions are performed on an interval of 50 seconds (see Fig. 2 (middle) for SGPD and Fig. 2 (right) for Euler). The rigid body dynamics preserve the invariant $x_1^2+x_2^2+x_3^2 = 1$, which refers to the ellipsoid determined by the axis of the rotating body. We include this property as prior knowledge in our SGPD model via $x^T \hat{f}(x) = 1$ [12]. The dynamics $\hat{f}$ is again trained with independent sparse GPs, where the third dimension is obtained by solving $\hat{f}(x) = 1 - \frac{\hat{f}_1 x_1 + \hat{f}_2 x_2}{x_3}$.

### 6.4 Results

For systems (i),(ii), and (iii), we demonstrate volume preservation. Fig. 4 depicts that volume is preserved for the symplectic integrator-based SGPD in contrast

to the standard explicit Euler method. Shown are the results for samples from the GP posterior in order to highlight the properties of the structure-preserving distribution. However, volume preservation applies to mean predictions as well. For the rigid body dynamics, we consider the invariant $x_1^2 + x_2^2 + x_3^2 = 1$. Fig. 4c demonstrates that the implicit midpoint is able to approximately preserve the invariant along 5 samples from the GP posterior. In contrast, the explicit Euler fails on all samples. The results demonstrate that even though the explicit Euler method achieves comparable results in terms of accuracy, it is not able to preserve structure. In summary, our method either shows the smallest $L^2$-error (see Table 1a) or achieves state-of-the-art accuracy. For the two-body problem, we demonstrate that the Euler long-term predictions are less stable than long-term predictions with SGPD. To this end we compute rollouts with 2500 points in Fig. 3 with SGPD (left) and Euler (right).

The midpoint method-based SGPD furthermore shows accurate approximation of the constant total energy for the systems (iii) and (iv). For system (iii), the total energy corresponds to the Hamiltonian $H$. Inspired by [11], we average the approximated energy along 5 independent trajectories $H_n = \sum_{i=1}^{5} \frac{H_n^i}{5}$ and compute the average total energy $\hat{H} = \frac{1}{n}\sum_n H_n$. Afterwards, we evaluate the error $\|H - \hat{H}\|$ and the deviation $\sqrt{\sum_n \frac{|H_n - H|^2}{n-1}}$ (see Table 1b). Our SGPD method yields the best approximation to the ground truth energy. In the appendix, we provide additional information for multiple runs of the experiment. For the rigid body dynamics our method yields accurate approximation of the energy as well. Details and an evaluation of higher-order methods is moved to the appendix.

## 7 Conclusion and future work

In this paper we combine numerical integrators with GP regression. Thus, resulting in an inference scheme that preserves physical properties and yields high accuracy. On a technical level, we derive a method that samples from implicitly defined distributions. By the means of empiricial comparison, we show the advantages over Euler-based state-of-the-art methods that are not able to preserve physical structure. An important extension that we want to address in the future are partial observations and control input.

**Acknowledgements** The authors thank Barbara Rakitsch, Alexander von Rohr and Mona Buisson-Fenet for helpful discussions.

## References

1. Abdulle, A., Garegnani, G.: Random time step probabilistic methods for uncertainty quantification in chaotic and geometric numerical integration. Statistics and Computing **30**(4), 907–932 (Feb 2020)

2. Alemi, A., Poole, B., Fischer, I., Dillon, J., Saurous, R.A., Murphy, K.: Fixing a broken ELBO. In: Proceedings of the 35th International Conference on Machine Learning, vol. 80, pp. 159–168, PMLR (2018)
3. Bai, S., Kolter, J.Z., Koltun, V.: Deep equilibrium models. In: Advances in Neural Information Processing Systems 32, pp. 690–701 (2019)
4. Brüdigam, J., Schuck, M., Capone, A., Sosnowski, S., Hirche, S.: Structure-preserving learning using Gaussian processes and variational integrators. In: Proceedings of the 4th Conference on Learning for Dynamics and Control, PMLR (2022)
5. Buisson-Fenet, M., Solowjow, F., Trimpe, S.: Actively learning Gaussian process dynamics. In: Proceedings of the 2nd Conference on Learning for Dynamics and Control, PMLR (2020)
6. Chen, Z., Zhang, J., Arjovsky, M., Bottou, L.: Symplectic recurrent neural networks. In: 8th International Conference on Learning Representations, ICLR 2020 (2020)
7. Doerr, A., Daniel, C., Schiegg, M., Nguyen-Tuong, D., Schaal, S., Toussaint, M., Trimpe, S.: Probabilistic recurrent state-space models. In: Proceedings of the International Conference on Machine Learning (ICML) (2018)
8. Frigola, R., Chen, Y., Rasmussen, C.: Variational Gaussian process state-space models. Advances in Neural Information Processing Systems 27 (NIPS 2014) pp. 3680–3688 (2014)
9. Geist, A., Trimpe, S.: Learning constrained dynamics with Gauss' principle adhering Gaussian processes. In: Proceedings of the 2nd Conference on Learning for Dynamics and Control, pp. 225–234, PMLR (2020)
10. Gould, S., Hartley, R., Campbell, D.: Deep declarative networks: A new hope. arXiv:1909.04866 (2019)
11. Greydanus, S., Dzamba, M., Yosinski, J.: Hamiltonian neural networks. In: Advances in Neural Information Processing Systems 32, pp. 15379–15389 (2019)
12. Hairer, E., Lubich, C., Wanner, G.: Geometric numerical integration: structure-preserving algorithms for ordinary differential equations. Springer (2006)
13. Hairer, E., Nørsett, S., Wanner, G.: Solving Ordinary Differential Equations I – Nonstiff Problems. Springer (1987)
14. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II – Stiff and Differential-Algebraic Problems. Springer (1996)
15. Hegde, P., Çağatay Yıldız, Lähdesmäki, H., Kaski, S., Heinonen, M.: Variational multiple shooting for Bayesian ODEs with Gaussian processes. In: Proceedings of the 38th Uncertainty in Artificial Intelligence Conference, PMLR (2022)
16. Heinonen, M., Yildiz, C., Mannerström, H., Intosalmi, J., Lähdesmäki, H.: Learning unknown ODE models with Gaussian processes. In: Proceedings of the 35th International Conference on Machine Learning (2018)
17. Hensman, J., Fusi, N., Lawrence, N.: Gaussian processes for big data. Uncertainty in Artificial Intelligence - Proceedings of the 29th Conference, UAI 2013 (2013)

18. Howarth, R.J.: Mining geostatistics. London & New york (academic press), 1978. Mineralogical Magazine **43**, 1–4 (1979)
19. Ialongo, A.D., Van Der Wilk, M., Hensman, J., Rasmussen, C.E.: Overcoming mean-field approximations in recurrent Gaussian process models. In: Proceedings of the 36th International Conference on Machine Learning (ICML) (2019)
20. Jin, P., Zhang, Z., Zhu, A., Tang, Y., Karniadakis, G.E.: Sympnets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems. Neural Networks **132**(C) (12 2020)
21. Krantz, S., Parks, H.: The implicit function theorem. History, theory, and applications. Reprint of the 2003 hardback edition (01 2013)
22. Ljung, L.: System identification. Wiley encyclopedia of electrical and electronics engineering pp. 1–19 (1999)
23. Look, A., Doneva, S., Kandemir, M., Gemulla, R., Peters, J.: Differentiable implicit layers. In: Workshop on machine learning for engineering modeling, simulation and design at NeurIPS 2020 (2020)
24. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. In: Advances in Neural Information Processing Systems, vol. 20 (2008)
25. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press (2005)
26. Rath, K., Albert, C.G., Bischl, B., von Toussaint, U.: Symplectic Gaussian process regression of maps in Hamiltonian systems. Chaos **31 5** (2021)
27. Saemundsson, S., Terenin, A., Hofmann, K., Deisenroth, M.P.: Variational integrator networks for physically structured embeddings. In: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS), vol. 108 (2020)
28. Sakurai, J.J.: Modern quantum mechanics; rev. ed. Addison-Wesley (1994)
29. Salmon, R.: Hamiltonian fluid mechanics. Annual Review of Fluid Mechanics **20**, 225–256 (2003)
30. Tao, M.: Explicit symplectic approximation of nonseparable Hamiltonians: Algorithm and long time performance. Physical Review E **94**(4) (Oct 2016)
31. Titsias, M.: Variational learning of inducing variables in sparse Gaussian processes. Journal of Machine Learning Research - Proceedings Track pp. 567–574 (2009)
32. Turner, R., Deisenroth, M., Rasmussen, C.: State-space inference and learning with Gaussian processes. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, vol. 9, pp. 868–875, PMLR (2010)
33. Wang, J., Fleet, D., Hertzmann, A.: Gaussian process dynamical models for human motion. IEEE transactions on pattern analysis and machine intelligence **30**, 283–98 (2008)
34. Wilson, J., Borovitskiy, V., Terenin, A., Mostowsky, P., Deisenroth, M.: Efficiently sampling functions from Gaussian process posteriors. In: Proceedings of the 37th International Conference on Machine Learning, vol. 119, pp. 10292–10302 (2020)

35. Zhong, Y.D., Dey, B., Chakraborty, A.: Symplectic ODE-net: Learning Hamiltonian dynamics with control. In: 8th International Conference on Learning Representations, ICLR 2020 (2020)