PnP-PIML: Physics-informed Learning of Outlier Dynamics using Uncertainty Quantified Port-Hamiltonian Models

Kaiyuan Tan[1], Peilun Li[1], Jun Wang[2], Thomas Beckers[1]

***Abstract*— The ability to predict trajectories of surrounding agents and obstacles is a crucial component in many robotic applications. Data-driven approaches are commonly adopted for state prediction in scenarios where the underlying physical functions are unknown. However, the performance, reliability, and uncertainty of data-driven predictors become compromised when encountering outlier or out-of-distribution dynamics relative to the training data. In this paper, we introduce a Plug-and-Play Physics-Informed Machine Learning (PnP-PIML) framework to address this challenge. Our method employs conformal prediction to identify outlier dynamics and, in that case, switches from a nominal predictor to a physics-consistent model, namely distributed Port-Hamiltonian systems (dPHS). We leverage Gaussian processes to model the energy function of the dPHS, enabling not only the learning of system dynamics but also the quantification of predictive uncertainty through its Bayesian nature. In this way, the proposed framework does not require exact knowledge of the underlying physics, as it leverages PHS as a generalized structure of any real-world system.**

Code + Dataset: PnP-PIML LINK

I. Introduction

Recent advances in robotics have enabled a wide range of applications, such as goods delivery, environmental surveying, and firefighting. However, some robots, such as drones, are inherently vulnerable to damage from both rigid and deformable obstacles, e.g., electric wires in the air. Consequently, understanding and modeling the dynamics of these obstacles is crucial to improving the safety and reliability of fragile robotic systems. Precise modeling of obstacle dynamics enables robots to perform safe path planning, thereby minimizing collision risks and ensuring operational safety. However, the various physical properties of different objects can lead to hard-to-predict obstacle behaviors, making it difficult to establish a unified modeling framework that can accurately predict and respond to these dynamics.

Classical approaches to modeling unknown objects create (linear) autoregressive models that can capture the main behaviors of the system (Lydia et al., 2016). They often face challenges when dealing with highly nonlinear effects, leading to solutions that may not accurately predict the trajectory of the obstacles. However, obstacles with nonlinear dynamics are quite common in real-world environments, e.g., cables on a construction site, but have not been extensively studied. In particular, the challenge of learning the dynamics of flexible obstacles is non-trivial due to the inherent complexity and nonlinearity of their behavior (Wei et al., 2017). To address the limitations of traditional first principle models in capturing complex systems, data-driven methods have been proposed for state forecasting tasks, e.g. (Long et al., 2019; Stephany and Earls, 2022), leveraging their universal approximation properties. Especially, deep neural networks can learn complex, nonlinear patterns, enabling them to model intricate system dynamics that linear models cannot easily handle. However, purely data-driven approaches raise concerns about the efficiency, reliability, and physical correctness of the learned model (Hou and Wang, 2013). The absence of physical priors in these models often leads to limited trustworthiness and other disadvantages such as vulnerability against adversarial attacks, see (Zhang et al., 2022; Tan et al., 2023). As a result, there is growing interest in developing learning methods that are both trustworthy and capable of effectively modeling complex systems.

Some recent work has been proposed based on Physics-informed Neural Networks (PINNs) (Raissi et al., 2019; Si and Yan, 2024; Serrano et al., 2024). Compared to standard neural networks, PINNs are a supervised learning approach that considers physics constraints in the loss function. By explicitly adding physics conditions, PINNs ensure that the learning outcomes follow the encoded physics equations. However, there are also two concerns regarding this approach: First, we do not always have access to the exact underlying physics functions, making it challenging to properly train the PINNs. Secondly, even with the boost of physics priors, we are still eager to know the accuracy of the model for reliable decision-making. Hence, there is a demand for uncertainty-quantified.

To address the challenge of limited knowledge of the exact underlying physical dynamics, the Port-Hamiltonian system (PHS) framework (Nageshrao et al., 2015; Duong and Atanasov, 2021) has gained prominence due to its ability to systematically represent a generalized physics-consistent structure. Recently, PHSs have been combined with Bayesian learning for both ordinary differential equation (ODE) (Beckers et al., 2022) and partial differential equation (PDE) (Tan et al., 2024), enabling the learning of the Hamiltonian for physically consistent predictions without further knowledge of the underlying equations. By providing confidence intervals, regions of potential risk, particularly those arising from nonlinear obstacle movements, can be effectively modeled.

However, these generalized physics models are computa-

[1]The authors are with the Department of Computer Science, Vanderbilt University, Nashville, TN 37212, USA. {kaiyuan.tan, peilun.li, thomas.beckers@vanderbilt.edu}

[2]The author is with the Department of Electrical and Systems Engineering, Washington University in St. Louis, Saint Louis, MO 63108, USA. {junw@wustl.edu}

tionally demanding, making them slow to make predictions. Hence, we only want to use them when necessary, and, otherwise, use predictors such as neural networks that support quick inference. Thus, it is necessary to identify whether the output of an existing, inference-efficient predictor would be reliable, and, if not, to switch over to the slower but physically consistent model. We propose a solution to provide physically reliable predictions in the case of out-of-distribution events, which can be directly integrated on top of any existing data-driven predictor.

Contribution: In this paper, we propose a novel plug-and-play physics-informed machine learning framework **PnP-PIML** to predict the motion of flexible objects. Our method aims to infuse the model with a deeper understanding of the physical laws governing the system. Unlike conventional pure-learning approaches, which may overlook conservation laws, our method utilizes Gaussian Process distributed Port-Hamiltonian systems (GP-dPHS) as a framework to incorporate these physical priors as inductive bias directly into the learning process. By mentioning plug-and-play, we are saying that our proposed learning method can be added to any existing predictor. We leverage conformal prediction (CP) to detect unreliable predictions and encode data through the dPHS framework into the Bayesian learning part. In addition, the Bayesian nature of GP-dPHS also enables uncertainty quantification to justify the reliability of the given predictions. We evaluate the proposed method on the task of predicting the motion of a flexible object, namely, an oscillating string.

II. Preliminary

In this section, we provide a brief overview of conformal prediction and distributed Port-Hamiltonian Systems.

A. Conformal Prediction

Conformal prediction (CP) is a statistical technique used to quantify the uncertainty of predictions in machine learning models. It provides a prediction set that contains the true output with a user-specified probability. Unlike traditional models that give point estimates, conformal prediction gives a range or set of possible outcomes, ensuring that the correct outcome is included within that set a certain percentage of the time. We use CP to derive a non-conformity score to decide whether new test data lies within the regime of calibration data, and thus results in trustworthy predictions, see (Angelopoulos and Bates, 2021; Zhao et al., 2024).

Consider[1] $R^{(0)}, R^{(1)}, \ldots, R^{(K)} \sim \mathscr{R}_0$, where these are $K+1$ independent and identically distributed (i.i.d.) random variables following a distribution $\mathscr{R}_0$, where the exchangeability of $R^{(0)}, \ldots, R^{(K)}$ is sufficient. The nonconformity score is denoted by the variable $R^{(i)}$ and its computation in our case can be described as follows: Let f be a neural network predictor that aims to predict trajectory $Z^{(i)}$ from the input array $X^{(i)}$. The score $R^{(i)}$ represents the prediction error, defined as $R^{(i)} = \left| Z^{(i)} - f(X^{(i)}) \right|$.

Our objective is to compute an upper bound for $R^{(0)}$, which corresponds to the score for the test data. We utilize the values $R^{(1)}, \ldots, R^{(K)}$, obtained from the calibration process to achieve this. Given a failure probability $\delta \in (0,1)$, our goal is to determine a constant C such that

$$\mathbb{P}(R^{(0)} \leq C) \geq 1 - \delta.$$

The probability $\mathbb{P}(\cdot)$ is defined with respect to the joint distribution of $R^{(0)}, \ldots, R^{(K)}$. Then, we can estimate C as the $(1-\delta)$-th quantile of the empirical distribution of $R^{(1)}, \ldots, R^{(K)}$, adjusted by a factor of $1/K$. Hence, with a probability less than $(1-\delta)$, the new test data score would be less than the constant C. Formally, we set

$$C := \text{Quantile}_{(1+1/K)(1-\delta)}(R^{(1)}, \ldots, R^{(K)}), \tag{1}$$

which implies a lower bound on the number of calibration data points K. We sort $R^{(1)}, \ldots, R^{(K)}$ in ascending order, then $C = R^{(p)}$ corresponds to the p-th smallest nonconformity score in the calibration set. In other words, as long as the score $R^{(0)}$ that we acquired based on the new test data is greater than the constant C, we can determine it as out-of-distribution with probability $(1-\delta)$.

B. Distributed Port-Hamiltonian System

The composition of Hamiltonian systems through input/output ports leads to the development of Port-Hamiltonian systems (PHS), a class of dynamical systems characterized by ports that define the interactions among their components. The classical finite-dimensional Port-Hamiltonian formulation is extended to accommodate distributed parameter systems and multivariable cases. Unlike finite-dimensional Port-Hamiltonian systems, where the interconnection, damping, and input/output matrices are typically constant, in the infinite-dimensional case, these matrices are replaced by matrix differential operators. These operators are assumed to be constant, meaning they do not explicitly depend on the state (or energy) variables. Similar to the finite-dimensional scenario, the system model can be easily derived once the Hamiltonian function is defined, given the underlying Stokes–Dirac structure. This generalized class of infinite-dimensional systems in Port-Hamiltonian form is highly versatile, allowing for the modeling of various classical PDEs and the description of several physical phenomena, such as heat conduction, piezoelectricity, and elasticity. In the following sections, we recall the definition of dPHS as provided in (Macchelli et al., 2004).

Denote by $\mathscr{Z}$ a compact subset of $\mathbb{R}^n$ representing the spatial domain of the distributed parameter system. Moreover, denote by J a skew-adjoint constant differential operator, and by G_d a constant differential operator. Let $\mathscr{H}\colon \mathscr{Z} \to \mathbb{R}$ be

[1] Vectors a and vector-valued functions $f(\cdot)$ are denoted with bold characters. Matrices are described with capital letters. I_n is the n-dimensional identity matrix and 0_n the zero matrix. The expression $A_{:,i}$ denotes the i-th column of A. For a positive semidefinite matrix Λ, $\|x-y\|_\Lambda^2 = (x-y)^\top \Lambda (x-y)$. $\mathbb{R}_{>0}$ denotes the set of positive real number whereas $\mathbb{R}_{\geq 0}$ is the set of non-negative real numbers. $\mathscr{C}^i$ denotes the class of i-th times differentiable functions. The operator ∇_x with $x \in \mathbb{R}^n$ denotes $[\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}]^\top$

the Hamiltonian functional such that

$$\mathscr{H}(x) = \int_{\mathscr{Z}} H(z,x)dV, \tag{2}$$

where $H\colon \mathscr{Z} \times \mathscr{X} \to \mathbb{R}$ is the energy density. Denote by $\mathscr{W}$ the space of vector-valued smooth functions on $\partial\mathscr{Z}$ representing the boundary terms $\mathscr{W} \coloneqq \{w | w = B_{\mathscr{Z}}(\delta_x\mathscr{H}, u)\}$ defined by the boundary operator $B_{\mathscr{Z}}$. Then, the general formulation of a multivariable dPHS Σ is fully described by

$$\Sigma(J,R,\mathscr{H},G) = \begin{cases} \frac{\partial x}{\partial t} = (J-R)\delta_x\mathscr{H} + G_d u \\ y = G_d^*\delta_x\mathscr{H} \\ w = B_{\mathscr{Z}}(\delta_x\mathscr{H}, u), \end{cases} \tag{3}$$

where R is a constant differential operator taking into account energy dissipation. Furthermore, $x(t,z) \in \mathbb{R}^n$ denotes the state (also called energy variable) at time $t \in \mathbb{R}_{\geq 0}$ and location $z \in \mathscr{Z}$ and $u, y \in \mathbb{R}^m$ the I/O ports. Generally, the J matrix defines the interconnection of the elements in the dPHS, whereas the Hamiltonian H characterizes their dynamical behavior. The constitution of the J matrix predominantly involves partial differential operators. The port variables u and y are conjugate variables in the sense that their duality product defines the energy flows exchanged with the environment of the system, for instance, currents and voltages in electrical circuits or forces and velocities in mechanical systems, see (Van der Schaft, 2000) for more information.

C. Gaussian process

A Gaussian process is fully characterized by its mean function, $m(\mathbf{x})$, and covariance function, $k(\mathbf{x},\mathbf{x}')$, which together define the process as $\mathscr{GP}(m(\mathbf{x}), k(\mathbf{x},\mathbf{x}'))$. The covariance function, denoted as kernel $k(\mathbf{x},\mathbf{x}')$, represents the covariance between any two input observations. Due to its properties, Gaussian processes become a widely used tool in non-parametric Bayesian regression. Given an input dataset $X = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}]$ with corresponding outputs $Y = [y^{(1)}, \ldots, y^{(N)}]$, the training data $\mathscr{D}$ is modeled as being sampled from an unknown function $y = f(\mathbf{x})$. By placing a GP prior on f, Bayesian inference can be applied to compute the posterior distribution for a new input $\mathbf{x}^*$. This inference updates both the mean and covariance of the GP by:

$$\mu(y^*|\mathbf{x}^*) = k(\mathbf{x}^*,X)K(X,X)^{-1}Y, \tag{4}$$

$$\Sigma(y^*|\mathbf{x}^*) = k(\mathbf{x}^*,\mathbf{x}^*) - k(\mathbf{x}^*,X)K(X,X)^{-1}k(\mathbf{x}^*,X), \tag{5}$$

where $k(\mathbf{x}^*,X)$ denotes the covariance between the new input and the training set, and $k(\mathbf{x}^*,\mathbf{x}^*)$ represents the prior variance at the new input. Consequently, the mean prediction and uncertainty for the new input are derived from the posterior predictive distribution $p(\mathbf{y}^*|\mathbf{x}^*,X,Y)$. For further details on Gaussian processes, see (Beckers, 2021).

III. Proposed PnP-PIML

In this section, we will discuss the assumptions and problem formulation, followed by a detailed discussion on how the proposed physics-informed machine learning framework seamlessly integrates with existing data-driven models to enhance predictive performance.

A. Assumptions and Settings

We aim to handle situations where the observation data is outside of the training data regime, which wildly exists in robotics applications. With the aroused attention regarding safety assurance, reliability, and performance, physics-informed approaches with uncertainty quantification start to gain visibility. We apply conformal prediction under calibration data distribution $\mathscr{D}_0$. Recalling the calculation of nonconformity score $R^{(i)}$ in II-A, we are able to get the constant C corresponds to the p-th smallest nonconformity score in the calibration set. When the score function is greater than this constant, we detect the outlier under $(1-\delta)$ confidence. Whenever the outlier in dataset $\mathscr{D}_1 = \{s(t_1,z_1), \ldots, s(t_1,z_{N_z}), \ldots, s(t_{N_t},z_1), \ldots, s(t_{N_t},z_{N_z})\}$ is detected, the Bayesian physics-informed learning branch is awakened (plug) and starts the learning steps (play). For many PDE physics systems, their dynamics can be encoded in distributed Port-Hamiltonian form; see 3. By encoding the inputs using dPHS form, we ensure that the evolution of the dynamics follows the rules that govern the real world.

In our case, the evolution of the states of a PDE system $x(t,z) \in \mathbb{R}^n$ over time $t \in \mathbb{R}_{\geq 0}$ and spatial domain $z \in \mathscr{Z}$ is defined by the equation with initial state $x(0,z)$ as described in II-B.

The Hamiltonian functional $\mathscr{H} \in \mathscr{C}^\infty$ is assumed to be *completely unknown* due to the unknown dynamics of the observed obstacle. Therefore, we aim to learn a dPHS model

$$\frac{\partial x}{\partial t} = (J-R)\delta_x\hat{\mathscr{H}} + G_d u \tag{6}$$

with an estimated Hamiltonian functional $\hat{\mathscr{H}}$ based on observations of the system (3). To effectively address the problem under consideration, the following assumptions are introduced:

Assumption 1: We can draw K exchangeable samples from the calibration dataset of the normal learning branch, which follows the distribution $\mathscr{D}_0$. The test time observations might be drawn from a different distribution $\mathscr{D}_1$, where $\mathscr{D}_0 \neq \mathscr{D}_1$.

Assumption 2: The structure of the system matrices J, R, and G is assumed to be known, except for a finite set of parameters $\Theta \subset \mathbb{R}^{n_\Theta}$, where n_Θ is the total number of unknown parameters in these matrices.

Assumption 3: We can observe the state of the PDE system (3) at certain temporal points t_i and spatial points z_j, resulting in a set of observations $\{x(t_1,z_1), \ldots, x(t_1,z_{N_z}), \ldots, x(t_{N_t},z_1), \ldots, x(t_{N_t},z_{N_z})\}$.

Assumption 4: There exists a unique and smooth solution $x(t,z)$ for the PDE system under the given boundary conditions $B_{\mathscr{Z}}$ (3).

Assumption 1 establishes a prerequisite for the PnP setting, ensuring the presence of outlier dynamics. Additionally, it supports the validity of conformal prediction. Assumption 2 is only mildly restrictive since J and G are typically matrix differential operators derived from prior knowledge of the observed PDE system. Although we assume knowledge

of the dissipation matrix R, we allow for unknown parameters, requiring only that the general structure be known, such as the friction model, but not the specific parameters. Assumption 3 ensures that data can be collected from the PDE system, implying the state is observable. If this is not the case, an observer must be implemented. Finally, Assumption 4 guarantees the existence of a solution, ensuring that deriving a model is meaningful.

B. PnP PIML Framework

1) Data Preparation: As described in the previous section, we define the observations as follows

$$\mathscr{D}_1 = \{t_i, z_j, x(t_i, z_j), u(t_i)\}_{i=1,j=1}^{i=N_t, j=N_z}, \tag{7}$$

where $s(t_i, z_j)$ denotes the observed position at time t_i and spatial point z_j, corresponding to N_t time steps and N_z spatial points. These observations align with the input sequence $\{u(t_1), u(t_2), \ldots, u(t_{N_t})\}$. As outlined in Assumption 1, a deep neural network, represented by the function f, is trained on the dataset $\mathscr{D}_0$, and calibrates its conformal prediction under the same data distribution. The prediction P of the next w future frames can be formulated as:

$$P_{N_t+1:N_t+w} = f\left(\{t_i, z_j, s(t_i, z_j), u(t_i)\}_{i=N_t-h, j=1}^{i=N_t, j=N_z}\right), \tag{8}$$

based on the h historical frames of the observations. Based on the calibration data, the conformal prediction quantile C is computed using the formulation in (1). If the prediction error R^i at future point i exceeds the corresponding quantile C, it can be inferred that an outlier dynamic exists. Then, we plug in our PnP-PIML framework to perform the outlier learning (Play). The pipeline is shown in figure 1. This plug-and-play (PnP) physics-informed learning framework is applicable in scenarios where failure in the standard data-driven approach is detected. Given that data for failure cases is often scarce and time-sensitive, our Bayesian physics-informed learning method offers greater data efficiency, accuracy, and reliability compared to retraining a DNN, particularly in cases with unknown underlying physics and limited observations.

In the practical context of robotics, the observations $\mathscr{D}_1$ are typically acquired via computer vision methods, which are prone to noise. Consequently, the dataset typically undergoes a data denoising process, as detailed in Section III-C, resulting in a clean dataset $\hat{\mathscr{D}}_1$.

We denote the temporal and spatial state variables (p, q) as x, where $p = \frac{\partial x}{\partial t}$ and $q = \frac{\partial x}{\partial z}$. Recalling the dPHS formulation in 3, we require the time and spatial derivatives of the state variable x. To achieve this, we apply a smooth GP function and derive the desired derivative over this function. Hence, we can acquire the desired derivative $\frac{\partial x}{\partial t}$ for the corresponding states x.

Algorithm 1 PnP-PIML: Plug-and-Play Physics-Informed Machine Learning with Uncertainty Quantification

Input: {DNN training data $\mathbf{D_0}$, DNN f, test data $\mathbf{D_1}$, Distributed Port-Hamiltonian system branch **dPHS**, Gaussian process **GP**}
Output: {Reliable Prediction $\tilde{\mathbf{P}}$}
Calibrate the non-conformality score R^i of f based on data in the same distribution of $\mathbf{D_0}$, get the threshold C
Calculate the score R^0 for $f(\mathbf{D_1})$ based on the observations

if $(R^0 \leq C)$ **then**
  Make prediction use existing predictor $\tilde{\mathbf{P}} = f(\mathbf{D_1})$
else
  Wake up **dPHS** branch and encode the limited observation data marked as $\mathbf{D_1'}$
  Energy representation learning $E = GP(\mathbf{D_1'})$
  Propagate the learned representations through a numerical solver and get the prediction: $\tilde{\mathbf{P}} = Solver(E)$
end if
Output: $\tilde{\mathbf{P}}$

Consequently, we construct a new dataset $\mathscr{E}$, which comprises the states $X = [\tilde{x}(t_1), \ldots, \tilde{x}(t_{N_t})]$ and the state derivatives $\dot{X} = \left[\frac{\partial \tilde{x}(t_1)}{\partial t}, \ldots, \frac{\partial \tilde{x}(t_{N_t})}{\partial t}\right]$, where $\tilde{x}(t_i) = [x(t_i, z_1)^\top, \ldots, x(t_i, z_{N_e})^\top]^\top$ represents the spatially stacked state at time t_i. Thus, the dataset is defined as $\mathscr{E} = [\dot{X}, X]$.

2) Encoding through dPHS model: With the underlying dynamics unknown, we encode the dataset $\mathscr{E}$ using a distributed Port-Hamiltonian model to ensure physical correctness. This approach leverages a Gaussian process to approximate the unknown Hamiltonian function while treating the parametric uncertainties in the matrices J, R, and G as hyperparameters. In simpler terms, learning the Hamiltonian derivatives is equivalent to learning the energy representations of the PDE system. This is made possible because GPs are invariant under linear transformations (Jidling et al., 2017), allowing us to incorporate the Hamiltonian functional derivatives into the GP framework. This integration of PHS dynamics into the GP is mathematically represented as

$$\frac{\partial x}{\partial t} \sim \mathscr{GP}(\hat{G}_\Theta u, k_{dphs}(x, x')), \tag{9}$$

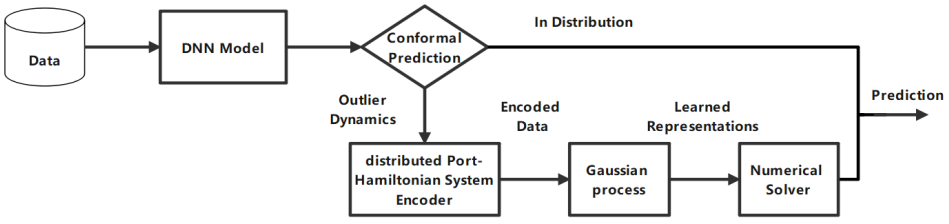

Fig. 1. Graphical illustration of proposed PnP-PIML pipeline, we leverage CP to detect the outlier dynamics and apply a data-efficient Bayesian physics-informed learning method to deal with the prediction failure

where the system dynamics are encapsulated within the probabilistic model. Consequently, we can rewrite a new physics-informed kernel function under the dPHS rule, k_{dphs}, which is formally described as

$$k_{dphs}(x,x') = \sigma_f^2 (\hat{J}R_\Theta)\delta_x \exp\left(-\frac{\|x-x'\|^2}{2\varphi_l^2}\right)\delta_{x'}^\top (\hat{J}R_\Theta)^\top, \tag{10}$$

see (Tan et al., 2024) for more details, Where $\hat{J}R_\Theta = \hat{J}_\Theta - \hat{R}_\Theta$. This kernel is based on the squared exponential function to represent the smooth Hamiltonian functional. The model is trained as a standard GP using the dataset $\mathscr{E}$. The matrices J, R, and G of the dPHS system are inferred through their estimates $\hat{J}_\Theta$, $\hat{R}_\Theta$, and $\hat{G}_\Theta$, where the unknown parameters, denoted by Θ, are treated as hyperparameters. The hyperparameters Θ, φ_l, and σ_f are optimized by minimizing the negative log marginal likelihood. This completes the training of the proposed GP-dPHS model.

Utilizing the invariance of GPs under linear transformation, we can rewrite (9) as:

$$\frac{\partial}{\partial t}x(t,z) = (\hat{J}_\Theta - \hat{R}_\Theta)\delta_x \hat{\mathscr{H}} + \hat{G}_\Theta u, \tag{11}$$

where the Hamiltonian $\hat{\mathscr{H}}$ follows a GP. By putting the GP priors based on the physics rule, we are able to make a comprehensive, informed prediction using the posterior from the GP, as shown in the following part.

C. Data Acquisition and Processing

1) Prediction: The PnP-PIML framework aims to learn the energy representations of the system rather than directly modeling the position.

To reduce computational complexity, we sample a deterministic Hamiltonian $\hat{\mathscr{H}}$ from the GP distribution and use a numerical solver to compute a solution for the equation in (11). Since a sample from the GP represents a deterministic function, and the dPHS structure adheres to the form in 3, the system is fully compliant with the Stokes-Dirac structure. This procedure ensures that the solution of the learned dynamics is physically consistent with the energy evolution, as it must satisfy the dPHS formulation.

The predicted sample is then propagated through a PDE solver with given initial conditions and time spans. Given that PDE learning is based on Bayesian physics priors, this multifaceted approach captures a powerful model of the dynamical interplay across various applications. Since its plug-and-play properties, it can be widely applied to existing learning method to deal with the outlier dynamics.

IV. Experimental evaluation

A. Setup

As an abstraction of deformable obstacles that robots may encounter in their operational environments, we aim to predict the motion of a spring (length of 45 cm and diameter of 1.6 cm) with fixed endpoints, exhibiting transversal oscillations. This setup mimics the dynamic behavior of flexible obstacles, such as a swinging electrical wire. The goal is to predict the spring's trajectory based on test data. Initially, a pre-trained DNN is employed for prediction; however, when outlier dynamics are detected using conformal prediction, the plug-in branch is activated. We then compare the performance of the standard retraining data-driven approach with our Bayesian physics representation learning framework under failure scenarios, highlighting the advantages of the proposed method.

B. Data collection

We utilized a high-speed camera to record the motion of the spring at a frame rate of 240 frames per second (FPS). To facilitate the segmentation of the spring from the background, the spring was colored red. The initial image segmentation was performed using RGB thresholding, calculated as follows

$$\sqrt{(R-R_0)^2 + (G-G_0)^2 + (B-B_0)^2} \geq T, \tag{12}$$

where T is the threshold value, and R, G, and B represent the color values captured from the video footage. The parameters R_0, G_0, and B_0 correspond to the background color values. For this segmentation task, the RGB color range was set with a lower bound of $[150,0,0]$ and an upper bound of $[255,80,80]$. Based on the formulation 12, the body movement of the flexible obstacle is segmented. We applied this mask to segment the spring's body. In pursuit of computational efficiency, the shape of soft obstacles is simplified using skeletonization. By applying the mask from equation 12, the pixel value can be described as a binary value. This step is visualized in Figure 2. Considering potential video noise in real-world scenarios, we account for this by using a Kalman filter to denoise the skeletonized data. The denoised dataset, $\hat{\mathscr{D}}_1$, is then used to train a Gaussian Process model, with the the spatial and temporal variables (t_i, z_j), and the state (transversal deflection) $s(t_i, z_j)$ of the spring. By employing a smooth kernel function k, such as the squared exponential kernel, the GP model is constructed. The derivatives of the GP function provide estimates for $\frac{\partial s}{\partial t}$ and $\frac{\partial s}{\partial z}$, allowing the dataset to be augmented with N_e additional spatial points, where $N_e \gg N_z$.

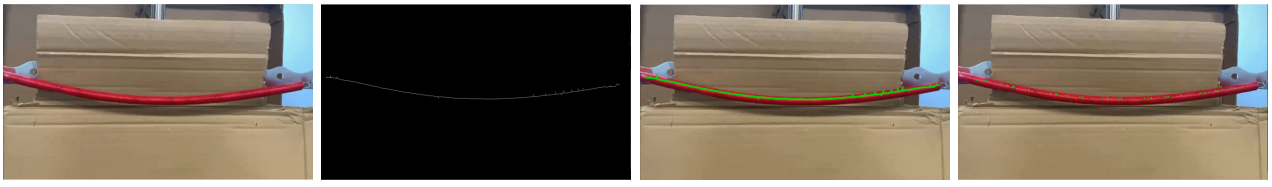

Fig. 2. From left to right, the figure illustrates the process starting with the original video, exact skeletonization, visualization of the centerline points through skeletonization, and finally data denoising.

C. Predictor structures

a) Neural Network (nominal predictor): Recalling the pipeline depicted in Figure 1, we assume the existence of a pre-trained DNN model serving as the nominal predictor. In this study, we construct a 100-layer neural network trained on a dataset representing the oscillatory motion of a rigid body governed by near-linear dynamics. To enhance the realism of the simulated data, Gaussian white noise is introduced to the angular velocity. The model is trained on 100,000 observation samples, each consisting of 100 discrete points along the rigid body. Following training, the neural network demonstrates stable performance in predicting the dynamics of the rigid body system.

b) Gp-dPHS: We recast the oscillating spring system into a general dPHS framework, where, however, the Hamiltonian $\mathscr{H}$ is unknown, leading to

$$\frac{\partial}{\partial t}\begin{bmatrix} p(t,z) \\ q(t,z) \end{bmatrix} = \underbrace{\begin{bmatrix} -c & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial z} & 0 \end{bmatrix}}_{\mathscr{J}-\mathscr{R}} \delta_x \mathscr{H}, \tag{13}$$

where $p = \frac{\partial x}{\partial t}$ and $q = \frac{\partial x}{\partial z}$, see Tan et al. (2024).

D. Results

In the presence of out-of-distribution scenarios, the quality of the nominal predictor (the neural network) drops, which is noticed by the conformal prediction, see table I. Recall the score calculation in II-A. A higher score R^0 represents higher prediction errors. When this score is greater than the threshold quantified by C, then we can say under $\delta = 10\%$ failure probability, this new test input is an outlier. After detecting that the nominal predictor is out-of-distribution, we switch the GP-dPHS model for prediction. As evaluation of its prediction quality, we compare it against a retrained deep neural network and a vanilla Gaussian Process. We train the Deep Neural Network, vanilla GP, and the proposed GP-dPHS model using the observed outlier data. The dataset consists of 234 time frames, each containing 50 spatial points. We use the first 80% frames of the test points as the training set, and the remaining 20% as the test set to evaluate model performance. For the GP-dPHS, we optimized the unknown damping factor by an Adam optimizer to $c = 0.03$. The results of the prediction compared to the ground truth are presented in Fig. 3 and 4. It can be seen, that the non-physics informed approaches (DNN and vanilla GP Fig. 3) fail to accurately predict the motion of the spring whereas our GP-dPHS captures the underlying physics and makes more accurate predictions.

Data	**Conformal prediction score** (Normalized)
In distribution	$C = 0.0016$, for $\delta = 0.1$
Outlier Data	$R^0 = 0.128$

TABLE I

CP DETECTION OF OUTLIER

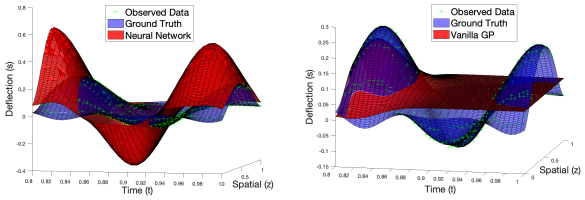

Fig. 3. Non-physics informed predictions. Left: The retrained DNN's (red) cannot accurately predict the before unseen test data (blue surface). Right: The vanilla GP prediction (red) remains nearly flat, suggesting inadequate learning of the system's dynamics.

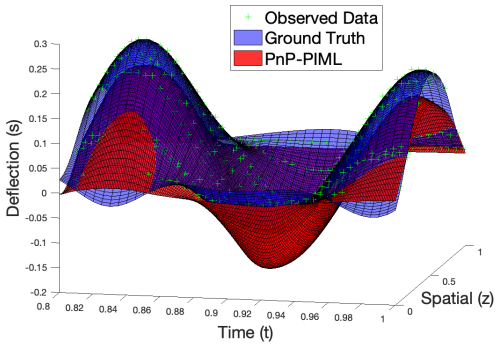

Fig. 4. Proposed PnP-PIML approach: The physics-consistent GP-dPHS generalizes well on unseen test data, leading to improved accuracy.

Furthermore, by employing the dPHS framework for data encoding, we are able to capture the energy representations of the systems, as depicted in Figure 5. As discussed in Section III-C.1, it is possible to sample from the dPHS kernel and integrate these samples using a numerical solver to model the system's motion with uncertainty quantification, see Figure 5.

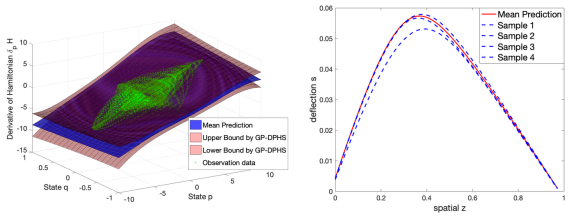

Fig. 5. Left: Energy representations (derivate of Hamiltonian) learned by GP with 95% confidence intervals. Right: Example of drawn samples at a fixed time point to determine the uncertainty of the prediction.

V. CONCLUSION

In this paper, we propose the PnP-PIML framework to increase the accuracy of predictions in the case that the test data is out-of-distribution. We use CP to identify outliers and, if detected, switch from the nominal predictor to a physics-informed model. Our approach ensures the physical fidelity of the predictions, demonstrates robust generalization with limited observations, and incorporates uncertainty quantification. It can be seamlessly integrated into any existing data-driven framework to enhance performance in scenarios involving complex outlier dynamics.

References

M. Lydia, S. S. Kumar, A. I. Selvakumar, and G. E. P. Kumar, "Linear and non-linear autoregressive models for short-term wind speed forecasting," *Energy conversion and management*, vol. 112, pp. 115–124, 2016.

Z. Wei, W. Chen, H. Wang, and J. Wang, "Manipulator motion planning using flexible obstacle avoidance based on model learning," *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, p. 1729881417703930, 2017.

Z. Long, Y. Lu, and B. Dong, "Pde-net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network," *Journal of Computational Physics*, vol. 399, p. 108925, 2019.

R. Stephany and C. Earls, "PDE-read: Human-readable partial differential equation discovery using deep learning," *Neural Networks*, vol. 154, pp. 360–382, 2022.

Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.

Q. Zhang, S. Hu, J. Sun, Q. A. Chen, and Z. M. Mao, "On adversarial robustness of trajectory prediction for autonomous vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 159–15 168.

K. Tan, J. Wang, and Y. Kantaros, "Targeted adversarial attacks against neural network trajectory predictors," in *Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 431–444.

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*, vol. 378, pp. 686–707, 2019.

C. Si and M. Yan, "Initialization-enhanced physics-informed neural network with domain decomposition (idpinn)," *arXiv preprint arXiv:2406.03172*, 2024.

G. Serrano, M. Jacinto, J. Ribeiro-Gomes, J. Pinto, B. J. Guerreiro, A. Bernardino, and R. Cunha, "Physics-informed neural network for multirotor slung load systems modeling," *arXiv preprint arXiv:2405.09428*, 2024.

S. P. Nageshrao, G. A. Lopes, D. Jeltsema, and R. Babuška, "Port-Hamiltonian systems in adaptive and learning control: A survey," *IEEE Transactions on Automatic Control*, vol. 61, no. 5, pp. 1223–1238, 2015.

T. Duong and N. Atanasov, "Hamiltonian-based neural ode networks on the se (3) manifold for dynamics learning and control," *arXiv preprint arXiv:2106.12782*, 2021.

T. Beckers, J. Seidman, P. Perdikaris, and G. J. Pappas, "Gaussian process port-hamiltonian systems: Bayesian learning with physics prior," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 1447–1453.

K. Tan, P. Li, and T. Beckers, "Physics-constrained learning for pde systems with uncertainty quantified port-hamiltonian models," *arXiv preprint arXiv:2406.11809*, 2024.

A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," *arXiv preprint arXiv:2107.07511*, 2021.

Y. Zhao, B. Hoxha, G. Fainekos, J. V. Deshmukh, and L. Lindemann, "Robust conformal prediction for stl runtime verification under distribution shift," in *2024 ACM/IEEE 15th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2024, pp. 169–179.

A. Macchelli, A. J. Van Der Schaft, and C. Melchiorri, "Port Hamiltonian formulation of infinite dimensional systems i. modeling," in *2004 43rd IEEE Conference on Decision and Control (CDC)*, vol. 4. IEEE, 2004, pp. 3762–3767.

A. Van der Schaft, *L2-gain and passivity techniques in nonlinear control*. Springer, 2000.

T. Beckers, "An introduction to gaussian process models," *arXiv preprint arXiv:2102.05497*, 2021.

C. Jidling, N. Wahlström, A. Wills, and T. B. Schön, "Linearly constrained Gaussian processes," *Advances in Neural Information Processing Systems*, vol. 30, 2017.