
Learning Partially Known Stochastic Dynamics with Empirical PAC Bayes

Manuel Haußmann^{1*}

Sebastian Gerwinn^{2*}

Andreas Look²

Barbara Rakitsch²

Melih Kandemir²

¹Heidelberg University, Germany

manuel.haussmann@iwr.uni-heidelberg.de

²Bosch Center for Artificial Intelligence, Renningen, Germany

firstname.lastname@de.bosch.com

Abstract

Neural Stochastic Differential Equations (NSDEs) model dynamical environments with neural nets assigned to their drift and diffusion terms. The high expressive power of their nonlinearity comes at the expense of instability in the identification of the large set of free parameters. This paper presents a recipe to improve the prediction accuracy of such models in three steps: i) accounting for epistemic uncertainty by assuming probabilistic weights, ii) incorporation of partial knowledge on the state dynamics, and iii) training the resultant hybrid model by an objective derived from a PAC Bayesian generalization bound. We observe in our experiments that this recipe effectively translates partial and noisy prior knowledge into an improved model fit.¹

1 Introduction

In many engineering applications, it is often easy to model dominant characteristics of a dynamical environment by a system of differential equations with a small set of state variables. In contrast, black-box machine learning methods are often highly accurate but less interpretable. Pushing the model towards high fidelity by capturing intricate properties of the environment, however, often requires highly flexible, e.g. over-parameterized models. Fitting these models to data can, in turn, result in over-fitting and hence poor generalization ability due to their high capacity.

Our work combines the benefits of both types of models by hybrid modeling: We set up the learning task as a non-linear system identification problem with partially known system characteristics. In the following, we assume to have access to a differential equation system that describes the dynamics of the target environment with low fidelity, e.g. by describing the vector field on a reduced dimensionality, by ignoring detailed models of some system components, or by avoiding certain dependencies for computational feasibility. We incorporate the system provided by the domain expert into a non-linear system identification engine, which we choose to be a *Bayesian Neural Stochastic Differential Equation* (BNSDE) to cover a large scope of dynamical systems, resulting in a *hybrid model*.

We propose a new algorithm for stable and effective training of such a hybrid BNSDE that combines the strengths of two statistical approaches: i) Bayesian model selection ([Williams and Rasmussen](#),

*Equal contribution

¹See <https://arxiv.org/abs/2006.09914> for the most current version.

2006), and ii) Probably Approximately Correct (PAC) Bayesian training (McAllester, 1999; Seeger, 2002). We improve the theoretical links between these two approaches (Germain et al., 2016b) by demonstrating how they can co-operate *during* training. To this end, we propose a novel training objective that suits to SDE inference, and derive a PAC-Bayesian generalization bound. Further, we provide a proof that this bound is upper bounded by the marginal likelihood of the BSDE hyperparameters and a complexity penalizer. Gradients of this upper bound are *tied* to the actual PAC bound, hence tightening the upper bound also tightens the PAC bound. Consequently, optimizing this bound amounts to Empirical Bayes stabilized by a regularizer developed from first principles. We refer to using this objective for training as *Empirical PAC Bayes*.

We demonstrate that our method can translate coarse descriptions of the true underlying dynamics into a consistent increase of forecasting accuracy. We first show in an ablation study the necessity of each of the multiple steps that comprise our method. Finally, we demonstrate in a real-world motion capture modeling task, that our method outperforms black-box system identification approaches (Chen et al., 2018; Hegde et al., 2019; Look and Kandemir, 2019) and alternative hybridization schemes that incorporate second-order Newtonian mechanics (Yildiz et al., 2019).

2 Background

Our contribution combines approaches from stochastic differential equations, PAC learning, and Empirical Bayes. Hence, we first introduce each of these concepts.

Stochastic Differential Equations. Stochastic differential equations (SDEs) are an extension of ordinary differential equations (ODEs) to include stochastic fluctuations in the dynamics (Oksendal, 1992). If we let $\mathbf{h}_t \in \mathbb{R}^P$ denote the P -dimensional state, the dynamics can be written in the following form:

$$d\mathbf{h}_t = f(\mathbf{h}_t, t)dt + G(\mathbf{h}_t, t)dW_t, \quad (1)$$

where the drift term is given by an arbitrary non-linear function $f(\cdot, \cdot) : \mathbb{R}^P \times \mathbb{R}_+ \rightarrow \mathbb{R}^P$ and the matrix valued function $G(\cdot, \cdot) : \mathbb{R}^P \times \mathbb{R}_+ \rightarrow \mathbb{R}^{P \times P}$ governs the diffusion dynamics. Finally, W_t denotes a P -dimensional Wiener Process determining the stochastic fluctuations. The solution to the SDE is a stochastic process \mathbf{h}_t . By setting the diffusion term G to zero, the SDE reverts to an ODE.

As analytical solutions of SDEs, i.e. descriptions of \mathbf{h}_t are not available except for specific choices of f and G , one has to resort to numerical approximation methods. Analogous to the practice for ODEs, a common approach is to use the Euler-Maruyama (EM) method (Särkkä and Solin, 2019), which discretizes the SDE in time steps t_1, \dots, t_K , resulting in the following sample based approximation to the joint distribution:

$$\begin{aligned} \mathbf{h}_{t_{k+1}} &= \mathbf{h}_{t_k} + f(\mathbf{h}_{t_k}, t_k)\Delta t_k + G(\mathbf{h}_{t_k}, t_k)\Delta W_k, \\ \Delta W_k &\sim \mathcal{N}(0, \Delta t_k \mathbf{1}_P), \quad \Delta t_k := t_{k+1} - t_k, \end{aligned} \quad (2)$$

where $\mathbf{1}_P$ is a P dimensional identity matrix. Using this sampling scheme, we obtain an approximation to the joint distribution $p(\mathbf{h}_{t_1}, \dots, \mathbf{h}_{t_K})$ for the given (fixed) drift and diffusion functions. In the following, we rely on this approximation scheme.

PAC Learning. Probably Approximately Correct (PAC) bounds quantify the generalization capabilities of a model from a training set to the true data distribution (McAllester, 1999; 2003). To this end, a risk $R(h) = \mathbb{E}_x[l(x, h(x))]$ of a hypothesis h is defined via a loss function $l(x, h(x))$ that measures the loss of the hypothesis evaluated at a data point x . Particularly, we build upon the Bayesian PAC formulation, in which the generalization performance of a Bayesian posterior, i.e. a distribution Q over hypotheses, is characterized by the following bound:

$$\forall Q : \mathbb{E}_{h \sim Q}[R(h)] \leq \mathbb{E}_{h \sim Q}[R_{\mathcal{D}}(h)] + \mathcal{C}(P, Q, \delta, N).$$

In the inequality above, $\mathbb{E}_Q[R(h)]$ is the expected risk across all hypotheses under the true data distribution, which is not accessible in practice, and $\mathbb{E}_Q[R_{\mathcal{D}}(h)] = \mathbb{E}_Q\left[\frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} l(x, h(x))\right]$ is its empirical counterpart in which the risk is averaged across the observed data \mathcal{D} . A distribution P over the hypotheses, referred to as the prior, determines the complexity term $\mathcal{C}(P, Q, \delta, N)$. This term additionally depends on the number of observed data points N and a confidence variable δ specifying the probability with which the bound holds (McAllester, 1999; Maurer, 2004).

Empirical Bayes. Bayesian models define a prior distribution $p_\phi(\theta)$ over parameters θ with hyperparameter ϕ , which together with the likelihood $p(\mathcal{D}|\theta)$ defines the full model. The standard approach consists of learning a posterior over these parameters $p(\theta|\mathcal{D})$ while keeping the hyperparameters ϕ fixed and marginalizing over θ in a second step to get the posterior predictive. An alternative, known as Empirical Bayes or Type-II maximum likelihood (see e.g. Bishop, 2006), directly marginalizes over the prior, and optimizes the resulting marginal likelihood/prior predictive with respect to the hyperparameters ϕ , i.e.

$$\arg \max_{\phi} \int p(\mathcal{D}|\theta)p_\phi(\theta)d\theta. \quad (3)$$

3 The Proposed Method

In this section, we describe how to combine the introduced tools into a coherent method for effective BNSDE inference. We first construct a BNSDE and equip it with domain-specific prior knowledge. We then derive a PAC-Bayesian objective to fit it to data, and conclude with results on the convergence of the proposed approach.

3.1 A Hybrid BNSDE

Application of deep learning to differential equation modeling paves the way to high-capacity predictors for capturing complex dynamics (Chen et al., 2018; Rackauckas et al., 2020). Neural Stochastic Differential Equations (NSDEs) (Look and Kandemir, 2019; Tzen and Raginsky, 2019) are SDEs as defined in (1) where the drift function, and potentially also the diffusion function are modeled as neural nets. As an initial step towards effective training, we introduce a prior distribution $p_\phi(\theta_f)$, parameterized by ϕ on the weights θ_f of a NSDE drift network, and arrive at

$$d\mathbf{h}_t = f_{\theta_f}(\mathbf{h}_t, t)dt + G(\mathbf{h}_t, t)dW_t, \quad \theta_f \sim p_\phi(\theta_f), \quad (4)$$

which we refer to as a *Bayesian Neural Stochastic Differential Equation (BSNDE)*. The epistemic uncertainty introduced on the network weights allows the model to quantify the model uncertainty, i.e. the knowledge of which synaptic map fits best to data, in addition to the aleatoric uncertainty that is modeled by the Wiener Process. For technical reasons to be clarified below, we assume $f_{\theta_f}(\cdot, \cdot)$ and $G(\cdot, \cdot)$ to be L -Lipschitz-continuous, and $G(\cdot, \cdot)$ not to have any learnable parameters.

In real-world applications, a coarse description of the environment dynamics is sometimes available as an incomplete set of differential equations. For instance, the dynamics of a three-dimensional volume might be modeled as a flow through a single point, such as the center of mass. Alternatively, a model on a subset of the system components might be provided. We assume this prior knowledge to be available as an ODE

$$d\mathbf{h}_t = r_\xi(\mathbf{h}_t, t)dt, \quad (5)$$

where $r_\xi(\cdot, \cdot) : \mathbb{R}^P \times \mathbb{R}_+ \rightarrow \mathbb{R}^P$ is an arbitrary non-linear function parameterized by a fixed set of parameters ξ . We can incorporate these known dynamics into the BNSDE by adding them to the drift

$$d\mathbf{h}_t = (f_{\theta_f}(\mathbf{h}_t, t) + \gamma \circ r_\xi(\mathbf{h}_t, t))dt + G(\mathbf{h}_t, t)dW_t, \quad (6)$$

which can be viewed as a hybrid SDE with the free parameter vector $\gamma \in [0, 1]^P$ governing the relative importance of prior knowledge on the learning problem and \circ referring to element-wise multiplication. Although we specified (5) within the same dimensional state space as (6), γ allows us to provide only partial information. When prior knowledge is available only for a subset of the state space dimensions, the remaining dimensions d can be filled in by simply setting $\gamma_d = 0$.

We define a prior stochastic process representing solely the prior knowledge of the dynamics as

$$d\mathbf{h}_t = (\gamma \circ r_\xi(\mathbf{h}_t, t))dt + G(\mathbf{h}_t, t)dW_t. \quad (7)$$

This prior SDE will be used as a reference distribution for complexity penalization as part of the final PAC training objective of our hybrid SDE. Note that we have used the same diffusion term as in (6) for specifying the prior SDE, which makes the complexity term within the PAC-formulation tractable, as we will show later. Also note that γ is a free parameter of the prior.

3.2 Learning via Empirical Bayes

Solving the SDE in (6) even for fixed parameters θ_f over an interval $[0, T]$ is analytically intractable for basically all practically interesting use cases. While our method is applicable to any discretization scheme, we demonstrate its use with the straightforward EM for simplicity, which gives us the discrete-time version of the hybrid BSDE below

$$\begin{aligned}\theta_f &\sim p_\phi(\theta_f), \quad \mathbf{h}_0 \sim p(\mathbf{h}_0), \\ \mathbf{h}_{k+1} | \mathbf{h}_k, \theta_f &\sim \mathcal{N}(\mathbf{h}_{k+1} | \mathbf{h}_k + d(\mathbf{h}_k, t_k) \Delta t_k, \Sigma_k), \\ d(\mathbf{h}_k, t_k) &= f_{\theta_f}(\mathbf{h}_k, t_k) + \gamma \circ r_\xi(\mathbf{h}_k, t_k)\end{aligned}$$

with $\Sigma_k := \mathbf{J}_k \Delta t_k$, $\mathbf{J}_k := G(\mathbf{h}_k, t_k)G(\mathbf{h}_k, t_k)^\top$, and $\Delta t_k := t_{k+1} - t_k$. The distribution $p(\mathbf{h}_0)$ is defined on the initial state.

Analogously to latent state space models, we assume that the observations of the dynamics described in (4), (6), and (7) are linked via a likelihood $p(\mathbf{y}_k | \mathbf{h}_k)$. Specifically, we observe these dynamics as time series $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_K\}$ consisting of K observations $\mathbf{y}_k \in \mathbb{R}^D$, collected at irregular time points $\mathbf{t} = \{t_1, \dots, t_K\}$.

Given an observed set of N such time series trajectories $\mathcal{D} = \{\mathbf{Y}_1, \dots, \mathbf{Y}_N\}$, the classical approach (MacKay, 2003; Gelman et al., 2013) would now require as a first step the inference of the posterior over both the global variables θ_f as well as the local variables $\mathbf{H}_n = \{\mathbf{h}_1^n, \dots, \mathbf{h}_K^n\}$, i.e. of $p(\theta_f, \mathbf{H}_1, \dots, \mathbf{H}_N | \mathcal{D})$, and as a second step a marginalization over this posterior to get the posterior predictive. As an analytical solution is intractable, approximate solutions such as Markov Chain Monte Carlo (MCMC) methods or Variational Inference (VI) are required. Application of either of these approaches to BNSDEs is prohibitive, the former computationally, the latter in terms of expressiveness since existing work makes strong independence and structural assumptions on the approximate posterior.

We propose in the following to apply model selection as an alternative path to BNSDE inference. Instead of performing the posterior inference on the latent variables, we marginalize them out and learn those hyperparameters ϕ from data that provide the highest log marginal likelihood (Williams and Rasmussen, 2006). That is our BNSDE learns the optimal ϕ^* via

$$\arg \max_{\phi} \int p(\mathcal{D} | \mathbf{H}) p(\mathbf{H} | \theta_f) p_\phi(\theta_f) d(\mathbf{H}, \theta_f). \quad (8)$$

An advantage of this construction is that the marginal likelihood has the identical functional form to the predictive distribution, which is the quantity of interest in a typical prediction task. Marginal likelihood learning has also been applied before in the context of Neural Networks (Sensoy et al., 2018; Malinin and Gales, 2018; Garnelo et al., 2018). Fitting the hyperparameters of an SDE to data via marginal likelihood maximization can also be viewed as an instance of the simulated likelihood method (Särkkä and Solin, 2019).

Marginalizing over θ_f in (8) is intractable for most practical use cases. However, it can be approximated by Monte Carlo integration without constructing chains on the global parameters. Sampling directly from the prior, we get for a single observation n and $s = 1, \dots, S$

$$\begin{aligned}\theta_f^s &\sim p_\phi(\theta_f), \quad \mathbf{H}^s \sim p(\mathbf{H} | \theta_f^s), \\ \phi^* &:= \arg \max_{\phi} \log \left(\frac{1}{S} \sum_{s=1}^S p(\mathcal{D} | \mathbf{H}^s) \right).\end{aligned} \quad (9)$$

In order to maximize this objective, we require an efficient computation of gradients w.r.t. the hyperparameter ϕ . Access to ϕ is only given via samples from the distribution it is parameterizing. In our experiments, we assume this distribution $p_\phi(\theta_f)$ to be normal, allowing us to make use of the standard reparameterization. We separate the sampling process into a parameter-free source of randomness and a parametric transformation, i.e. we have $\varepsilon \sim p(\varepsilon)$, $\theta_f = g_\phi(\varepsilon)$, for a suitable function $g_\phi(\cdot)$. In order to further reduce the variance noise introduced to the gradients due to this sampling step, we also use the *local reparameterization trick* (Kingma et al., 2015) in the drift, i.e. we sample the layer outputs during the forward propagation instead of individual layer weights.

The objective (9) is agnostic to the specific SDE employed. Therefore, we refer to the discretized black-box SDE in (4) governing $p(\mathbf{H} | \theta_f)$ and trained w.r.t. ϕ via this objective as *E-Bayes* throughout. Analogously, we refer to training a hybrid SDE as in (6) with the same method as *E-Bayes-Hybrid*.

3.3 A Trainable PAC Bound

A major downside of the objective in (9) when applied to BNSDEs is that it optimizes a large set of hyperparameters, i.e. means and variances of drift network weights, without a proper regularization aside from the implicit regularization inherent in the chosen architecture and the marginalization itself. While the hybrid approach already allows us to incorporate prior expert knowledge, it remains only a guiding signal without an explicit model capacity regularizer. We address this problem next by developing a training objective derived from a PAC-Bayesian bound objective that combines the benefits from the results we arrived at so far with a proper regularization scheme.

The proposed approach is still agnostic to the chosen discretization scheme. Consequently, we refer for any time horizon $T > 0$ to all local latent variables by $\mathbf{h}_{0 \rightarrow T}$. To distinguish the density given by the hybrid SDE in (6) from the prior SDE in (7), we denote the densities induced by them respectively as $p_{\text{hyb}}(\mathbf{h}_{0 \rightarrow T} | \theta_f)$ and $p_{\text{pri}}(\mathbf{h}_{0 \rightarrow T})$. Consequently, we define two distributions Q and P over $(\mathbf{h}_{0 \rightarrow T}, \theta_f)$. For the former, we have the joint distribution of the hybrid process developed in (6)

$$Q_{0 \rightarrow T}(\mathbf{h}_{0 \rightarrow T}, \theta_f) = p_{\text{hyb}}(\mathbf{h}_{0 \rightarrow T} | \theta_f) p_\phi(\theta_f), \quad (10)$$

while the latter stands for the joint distribution of the prior process developed in (7)

$$P_{0 \rightarrow T}(\mathbf{h}_{0 \rightarrow T}, \theta_f) = p_{\text{pri}}(\mathbf{h}_{0 \rightarrow T}) p_{\text{pri}}(\theta_f). \quad (11)$$

Although the prior process is independent of the drift parameters θ_f , we specify a fixed prior distribution $p_{\text{pri}}(\theta_f)$ within the prior P , which we choose to be a standard normal density within our experiments. In order to be compliant with the notational practice in the PAC Bayesian literature, we denote the prior distribution as P and the posterior distribution that is fit to data as Q . In the PAC-Bayesian framework, P and Q do not have to be linked to each other via application of the Bayes rule on an explicitly defined likelihood.

As both Q and P share the same diffusion term, the Kullback-Leibler (KL) divergence between these processes can be calculated by extending the proof of Archambeau et al. (2008). The following Lemma holds for any choice of diffusion term $G(\cdot, \cdot)$. The proofs for the Lemma and the following Theorems can be found in the appendix.

Lemma 1. *For the process distributions $Q_{0 \rightarrow T}$ and $P_{0 \rightarrow T}$, it holds that*

$$\begin{aligned} D_{KL}(Q_{0 \rightarrow T} || P_{0 \rightarrow T}) = & \\ & \frac{1}{2} \int_0^T \mathbb{E}_{Q_{0 \rightarrow T}} \left[f_{\theta_f}(\mathbf{h}_t, t)^\top \mathbf{J}_t^{-1} f_{\theta_f}(\mathbf{h}_t, t) \right] dt \\ & + D_{KL}(p_\phi(\theta_f) || p_{\text{pri}}(\theta_f)), \end{aligned}$$

for some $T > 0$, where $\mathbf{J}_t = G(\mathbf{h}_t, t)G(\mathbf{h}_t, t)^\top$.

This lemma provides one of the main ingredients for deriving a PAC-Bayesian bound on the generalization performance of a learned distribution $Q_{0 \rightarrow T}$. To derive such a bound, we additionally specify the risk via a loss function measuring the model mismatch. We assume the likelihood function $p(\mathbf{y}_t | \mathbf{h}_t)$ to be uniformly bounded everywhere.² We then define the true risk of a draw from $Q_{0 \rightarrow T}$ on an i.i.d. sampled trajectory $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_K\}$ at discrete and potentially irregular time points t_1, \dots, t_K drawn from an unknown ground-truth stochastic process $\mathfrak{G}(t)$ as the expected model misfit on the sample. Specifically, we define the risk over hypotheses $H = (\mathbf{h}_{0 \rightarrow T}, \theta_f)$ as follows:

$$R(H) = -\mathbb{E}_{\mathbf{Y} \sim \mathfrak{G}(t)} \left[\prod_{k=1}^K p(\mathbf{y}_k | \mathbf{h}_k) \right], \quad (12)$$

for time horizon $T > 0$ and the corresponding empirical risk on a data set $\mathcal{D} = \{\mathbf{Y}_1, \dots, \mathbf{Y}_N\}$ as

$$R_{\mathcal{D}}(H) = -\frac{1}{N} \sum_{n=1}^N \left[\prod_{k=1}^K p(\mathbf{y}_k^n | \mathbf{h}_k^n) \right]. \quad (13)$$

Next, we develop a PAC-Bayesian generalization bound building on these risk definitions. Furthermore, we upper bound it with a trainable objective.

²In our experiments, we ensure this condition by choosing the likelihood to be a normal density with bounded variance, i.e. bounded mass on the mode.

Theorem 1. *The expected true risk is bounded above with probability $\mathbb{P} \geq 1 - \delta$ by:*

$$\mathbb{E}_{H \sim Q_{0 \rightarrow T}} [R(H)] \leq \mathbb{E}_{H \sim Q_{0 \rightarrow T}} [R_{\mathcal{D}}(H)] + \mathcal{C}_{\delta}(Q_{0 \rightarrow T}, P_{0 \rightarrow T}) \quad (14)$$

$$\begin{aligned} &\leq -\frac{1}{N} \sum_{n=1}^N \log \left(\frac{1}{S} \sum_{s=1}^S \prod_{k=1}^K p(\mathbf{y}_k^n | \mathbf{h}_k^{n,s}) \right) \\ &\quad + \underbrace{\mathcal{C}_{\delta/2}(Q_{0 \rightarrow T}, P_{0 \rightarrow T}) + \sqrt{\frac{\log(2N/\delta)}{2S}} + K \log \bar{B}}_{=: C}, \end{aligned} \quad (15)$$

$$\leq -\frac{1}{SN} \sum_{n=1}^N \sum_{s=1}^S \sum_{k=1}^K \log \left(p(\mathbf{y}_k^n | \mathbf{h}_k^{n,s}) \right) + C \quad (16)$$

where $\bar{B} := \max_{\mathbf{y}_k, \mathbf{h}_k} p(\mathbf{y}_k | \mathbf{h}_k)$ is the uniform bound, S is the sample count taken independently for each observed sequence, and the complexity functional is given for some $\delta > 0$ as

$$\mathcal{C}_{\delta}(H_{0 \rightarrow T}, P_{0 \rightarrow T}) = \sqrt{\frac{D_{KL}(Q_{0 \rightarrow T} || P_{0 \rightarrow T}) + \log(2\sqrt{N}) - \log(\delta/2)}{2N}}$$

with $D_{KL}(Q_{0 \rightarrow T} || P_{0 \rightarrow T})$ as in Lemma 1.

Theorem 1 can be used to learn a posterior distribution $Q_{0 \rightarrow T}$ from data by adjusting ϕ . Additionally, we can also learn the importance of the prior by fitting the γ parameter to data. While directly learning γ by optimizing the PAC-bound violates the generalization guarantee, we can define a collection of prior distributions $P_{0 \rightarrow T}$ for a set Γ of discretized values of γ and employ the same union bound as Reeb et al. (2018). The resulting PAC-bound differs by a constant accounting for the number of distinct γ values within the collection. Therefore, we can use the same gradient based optimization to learn γ and quantize the value to the closest point within Γ to evaluate the PAC bound.

3.4 The Training Algorithm

The first term in (14) does not correspond to the Empirical Bayes objective as it averages over likelihoods, and not log-likelihoods (Germain et al., 2016a). However, the first term in (15) provides a sampling based approximation to the empirical Bayes objective. By defining the risk in such a way and employing the PAC-Bayesian framework, we obtain a regularized version of empirical Bayes. Although placing the $\log(\cdot)$ function into its summands loosens the bound on the true risk, it improves numerical robustness and optimizing (16) still tightens the original PAC-Bayesian bound, i.e. (14), as stated in the following corollary.

Corollary 1. *For Lipschitz-continuous risk and likelihood, a gradient step that reduces (16) also tightens the PAC bound in (14).*

Minimizing (16) hence closes the loop as the Empirical Bayes objective derived in (9) reappears in (15) but is now combined in a principled way with the regularization term \mathcal{C}_{δ} . We can ignore the terms that do not depend on ϕ and adopt the remaining expression bound as our final objective and learn ϕ^* via

$$\phi^* := \arg \max_{\phi} \frac{1}{SN} \sum_{n=1}^N \sum_{s=1}^S \sum_{k=1}^K \log \left(p(\mathbf{y}_k^n | \mathbf{h}_k^{n,s}) \right) + \sqrt{D_{KL}(H_{0 \rightarrow T} || P_{0 \rightarrow T}) + \log(4\sqrt{N}/\delta)/2N}.$$

In this training procedure, we only train w.r.t. ϕ which determine the drift term. To also learn the diffusion, one could represent G also by a BNN. However, the corresponding training procedure would invalidate the PAC statement. Nevertheless, the diffusion term could be learnt on a held-out data set and then incorporated as fixed to the bound (16). As Theorem 1 applies to any diffusion term, we keep the genericness of its statement. However, in the experiments, we stick to a constant diffusion term for practical reasons.

Although we require i.i.d. observations of time series in the theory, we can in practice use mini-batches of trajectories provided that the batches are sufficiently far apart so that they become essentially independent. The objective (15) differs from the Empirical Bayes one in (9) only by the complexity

term. The only complicated calculation step in this term is the integral through the process, which can be made more implementation friendly using Fubini’s theorem:

$$\int_0^T \mathbb{E}_{Q_{0 \rightarrow T}} \left[f_{\theta_f}(\mathbf{h}_t, t)^\top \mathbf{J}_t^{-1} f_{\theta_f}(\mathbf{h}_t, t) \right] dt = \mathbb{E}_{Q_{0 \rightarrow T}} \left[\int_0^T f_{\theta_f}(\mathbf{h}_t, t)^\top \mathbf{J}_t^{-1} f_{\theta_f}(\mathbf{h}_t, t) dt \right].$$

An algorithmic description of the resultant procedure is given in Algorithm 1. Our sampling-based method naturally couples with the EM approximation and inherits its convergence properties. Below we show strong convergence to the true solution with shrinking step size by extending the plain EM proof (Kloeden and Platen, 2011).

Theorem 2 (strong convergence). *Let \mathbf{h}_t^θ be an Itô process as in (4) with drift and diffusion parameters θ and \mathbf{h}_t^θ its Euler-Maruyama approximation for some regular step size $\Delta t > 0$. For some coefficient $R > 0$ and any $T > 0$, the below inequality holds*

$$\mathbb{E} \left[\sup_{0 \leq t \leq T} \left| \mathbb{E}_\theta[\mathbf{h}_t^\theta] - \frac{1}{S} \sum_{s=1}^S \tilde{\mathbf{h}}_t^{\theta^{(s)}} \right| \right] \leq R \Delta t^{1/2},$$

as $S \rightarrow \infty$ where $\{\theta^{(s)} \sim p_\phi(\theta) | s = 1, \dots, S\}$ are i.i.d. draws from a prior $p_\phi(\theta)$.

4 Experiments

We evaluate four variants of our method:

- (i) *E-Bayes.* Empirical Bayes without prior knowledge, i.e. training (9) with $p(\mathbf{h}_{0 \rightarrow T})$ given by (4).
- (ii) *E-PAC-Bayes.* Empirical PAC Bayes on the BNDSE using the objective in (16) with an uninformative prior drift, i.e. $r_\xi(\mathbf{h}_t, t) = 0$.
- (iii) *E-Bayes-Hybrid.* Same training objective as (i), however with the hybrid model as proposed in (6).
- (iv) *E-PAC-Bayes-Hybrid.* The hybrid model (6) with the same loss as *E-PAC-Bayes*, which is the combination we propose.

We extend the Empirical Bayes objective in (9) by PAC-Bayes to tune many hyperparameters without overfitting and incorporate prior domain knowledge in a principled way. We evaluate the first motivation as *E-PAC-Bayes*, i.e. objective (16) but without a prior SDE, and the complete model including a prior SDE as *E-PAC-Bayes-Hybrid*. See appendix for detailed discussion of the computational cost of each of these methods as well as further experiments.

Lorenz Attractor. This chaotic non-linear system has the the following inherently unsolvable dynamics

$$\begin{aligned} dx_t &= \zeta(y_t - x_t)dt + dW_t, \\ dy_t &= (x_t(\kappa - z_t) - y_t)dt + dW_t, \\ dz_t &= (x_t y_t - \rho z_t)dt + dW_t, \end{aligned}$$

where $\zeta = 10, \kappa = 28, \rho = 2.67$, and W_t is a random variable following a Wiener process with unit diffusion. We generate 1920 observations from the above dynamics initiating the system at $x_0, y_0, z_0 = (1, 1, 28)$, use the first half for training and the rest for testing. We split both the training and the test data into 20 sequences of length 24, which can be interpreted as i.i.d. samples of the system with different initial states. Table 1 presents the 24-step ahead forecasting error in MSE on

Table 1: Ablation study on the Lorenz attractor to evaluate the contributions of the prior knowledge on the predictive performance measured in Mean Squared Error (MSE) with standard deviation over ten repetitions. The hybrid models ((iii), (iv)) consistently improve on the black box models ((i),(ii)). The last row (v) shows the performance for the case the model has full access to the true dynamics with noisy parameters in (5).

Prior Knowledge	Model	Test MSE
None	(i)	29.67 ± 1.09
	(ii)	30.51 ± 1.53
$\gamma = [1, 0, 0], \zeta \sim \mathcal{N}(10, 1)$	(iii)	28.17 ± 1.05
	(iv)	26.98 ± 1.18
$\gamma = [0, 1, 0], \kappa \sim \mathcal{N}(2.67, 1)$	(iii)	15.93 ± 3.42
	(iv)	15.73 ± 3.22
$\gamma = [0, 0, 1], \rho \sim \mathcal{N}(28, 1)$	(iii)	28.72 ± 2.35
	(iv)	27.59 ± 1.15
$\gamma = [1, 1, 1], (\zeta, \kappa, \rho)^\top \sim \mathcal{N}((10, 2.67, 28)^\top, \mathbb{I}_3)$	(v)	16.42 ± 17.06

Table 2: Benchmarking of our method on the CMU Motion Capture Data Set. Mean Squared Error (MSE) and Negative Log-Likelihood (NLL) on 300 future frames is averaged over ten repetitions (\pm standard deviation).

Method	Reference	Bayesian	Hybrid	+KL	Test MSE	Test NLL
DTSBN-S	(Gan et al., 2015)	No	No	No	34.86 ± 0.02	Not Applicable
npODE	(Heinonen et al., 2018)	No	No	No	22.96	Not Applicable
Neural-ODE	(Chen et al., 2018)	No	No	No	22.49 ± 0.88	Not Applicable
ODE ² VAE	(Yildiz et al., 2019)	Yes	Yes	No	10.06 ± 1.40	Not Reported
ODE ² VAE-KL	(Yildiz et al., 2019)	Yes	Yes	Yes	8.09 ± 1.95	Not Reported
D-BNN (SGLD)	(Look and Kandemir, 2019)	Yes	No	No	13.89 ± 2.56	747.92 ± 58.49
D-BNN (VI)	(Hegde et al., 2019)	Yes	No	Yes	9.05 ± 2.05	452.47 ± 102.59
E-Bayes	Baseline	Yes	No	No	8.68 ± 1.56	433.76 ± 77.78
E-PAC-Bayes	Ablation	Yes	No	Yes	9.17 ± 1.20	489.82 ± 67.06
E-Bayes-Hybrid	Ablation	Yes	Yes	No	9.25 ± 1.99	462.82 ± 99.61
E-PAC-Bayes-Hybrid	Proposed	Yes	Yes	Yes	7.84 ± 1.41	415.38 ± 80.37

the test set for our model variants. In each experiment repetition, E-Bayes-Hybrid and E-PAC-Bayes-Hybrid are provided one equation after distorting the corresponding parameter by normal distributed noise. The other equations are held out by being hard assigned to zero. To set up the corresponding prior and model, we used a constant diffusion with $G = \mathbb{1}$. Despite the imprecision of the provided prior knowledge, the largest performance leap comes from the hybrid models. The complexity term on the PAC-Bayesian bound restricts the model capacity for black-box system identification, while it improves the hybrid setup.

Figure 1 visualizes the predicted trajectories on the test sequence for prior knowledge on dz_t . Even with weak prior knowledge, the proposed model is stable longer than the baseline as well as showing a proper increase in the variance as the predicted trajectory increases, while the baseline diverges sooner without a proper increase in uncertainty.

CMU Walking Data Set. We benchmark against the state of the art on this motion capture data set following the setup of Yildiz et al. (2019). We train an *E-PAC-Bayes* model on the MOCAP-1 data set that consists of 43 motion capture sequences measured from 43 different subjects. The drift net of the learned BNSDE is then treated as weak and broad prior knowledge on human walking dynamics. We use MOCAP-2 with 23 walking sequences from Subject 35 to represent a high-fidelity subject-specific modeling task. As reported in the Table 2 of Yildiz et al. (2019), the state of the art of subject-independent mocap dynamic modeling has twice as high prediction error as subject-specific dynamics (MSE of 15.99 versus 8.09). Analogously to the Lorenz attractor experiment, we fixed the prior diffusion term for the PAC-variants to be constant. We report the test MSE and negative log-likelihoods in Table 2. Our method delivers best prediction accuracy and model fit when all its components are active.

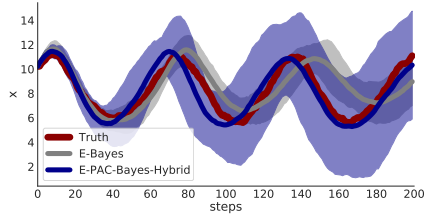


Figure 1: Predicted trajectory for 200 time steps starting at $T = 10$ of the Lorenz data set mapped to one dimension. The error bars indicate ± 2 standard deviations over 21 trajectories.

5 Conclusion

We have shown that our method incorporates vague prior knowledge into a flexible Bayesian black-box modelling approach for learning SDEs resulting in a robust learning scheme guided by generalization performance via a PAC-Bayesian bound. The method is easily adaptable to other solvers. For example, the training loss derived in (9) can also be optimized using a closed-form normal assumed density scheme applied over a stochastic Runge-Kutta variant (Li et al., 2019). Independent from the sampling scheme and model used, our tied gradient update procedure allows training on the loose, yet numerically stable, bound while providing an improvement w.r.t. the generalization guarantees

on its tighter counterpart. Our stochastic approximation of the data log-likelihood currently relies on samples obtained from the prior, yet could be improved by incorporating a more sophisticated sampling scheme, e.g. using particle filtering (Kantas et al., 2015). Finally, the bound in (16) has the potential to be vacuous for certain drift nets, incorporating a Hoeffding assumption (Alquier et al., 2016) could further tighten it.

References

- P. Alquier, J. Ridgway, and N. Chopin. On the properties of variational approximations of gibbs posteriors. *The Journal of Machine Learning Research*, 17(1):8374–8414, 2016.
- C. Archambeau, M. Opper, Y. Shen, D. Cornford, and J.S. Shawe-Taylor. Variational Inference for Diffusion Processes. In *NIPS*. 2008.
- C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.
- S. L Brunton, J.L. Proctor, and J.N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.
- O. Catoni. PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning. *IMS Lecture Notes Monograph Series*, 56, 2007.
- R. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural Ordinary Differential Equations. In *NeurIPS*, 2018.
- A. Doerr, C. Daniel, M. Schiegg, D. Nguyen-Tuong, S. Schaal, M. Toussaint, and S. Trimpe. Probabilistic Recurrent State-Space Models. In *ICML*, 2018.
- D. Durstewitz. A state space approach for piecewise-linear recurrent neural networks for reconstructing nonlinear dynamics from neural measurements. *arXiv preprint arXiv:1612.07846*, 2016.
- G. Dziugaite and D.M. Roy. Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data. In *UAI*, 2017.
- Z. Gan, C. Li, R. Henao, D.E. Carlson, and L. Carin. Deep Temporal Sigmoid Belief Networks for Sequence Modeling. In *NIPS*, 2015.
- M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y.W. Teh, D. Rezende, and S. M.A. Eslami. Conditional Neural Processes. *ICML*, 2018.
- Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.
- P. Germain, F. Bach, A. Lacoste, and S. Lacoste-Julien. PAC-Bayesian Theory Meets Bayesian Inference. In *NIPS*. 2016a.
- P. Germain, F. Bach, A. Lacoste, and S. Lacoste-Julien. PAC-Bayesian Theory Meets Bayesian Inference. In *NIPS*, 2016b.
- P. Hegde, M. Heinonen, H. Lähdesmäki, and S. Kaski. Deep Learning with Differential Gaussian Process Flows. In *AISTATS*, 2019.
- M. Heinonen, C. Yildiz, H. Mannerström, J. Intosalmi, and H. Lähdesmäki. Learning Unknown ODE Models with Gaussian Processes. In *ICML*, 2018.
- N. Kantas, A. Doucet, S.S. Singh, J. Maciejowski, N. Chopin, et al. On particle methods for parameter estimation in state-space models. *Statistical science*, 30(3):328–351, 2015.
- D.P. Kingma, T. Salimans, and M. Welling. Variational Dropout and The Local Reparameterization Trick. In *NIPS*, 2015.
- P.E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, 2011.

- J. Knoblauch, J. Jewson, and T. Damoulas. Generalized variational inference. *arXiv preprint arXiv:1904.02063*, 2019.
- X. Li, Y. Wu, L. Mackey, and M.A. Erdogdu. Stochastic Runge-Kutta accelerates Langevin Monte Carlo and beyond. In *NeurIPS*. 2019.
- A. Look and M. Kandemir. Differential Bayesian Neural Nets. In *4th NeurIPS Workshop on Bayesian Deep Learning*, 2019.
- D. JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.
- A. Malinin and M. Gales. Predictive uncertainty estimation via prior networks. In *NeurIPS*, 2018.
- A. Maurer. A note on the PAC Bayesian theorem. *arXiv preprint cs/0411099*, 2004.
- D. McAllester. PAC-Bayesian Model Averaging. In *COLT*, 1999.
- D. McAllester. PAC-Bayesian Stochastic Model Selection. *Machine Learning*, 51:5–21, 2003.
- B. Oksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer-Verlag, 1992.
- C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, and A. Ramadhan. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.
- D. Reeb, A. Doerr, S. Gerwinn, and B. Rakitsch. Learning Gaussian Processes by Minimizing PAC-Bayesian Generalization Bounds. In *NeurIPS*. 2018.
- S. Särkkä and A. Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.
- M. Seeger. PAC-Bayesian Generalisation Error Bounds for Gaussian Process Classification. *Journal of Machine Learning Research*, 3:233–269, 2002.
- M. Sensoy, L. Kaplan, and M. Kandemir. Evidential Deep Learning to Quantify Classification Uncertainty. In *NeurIPS*, 2018.
- B. Tzen and M. Raginsky. Neural Stochastic Differential Equations: Deep Latent Gaussian Models in the Diffusion Limit. *ArXiv*, abs/1905.09883, 2019.
- C. KI Williams and C. E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.
- C. Yildiz, M. Heinonen, and H. Lahdesmaki. ODE2VAE: Deep Generative Second Order ODEs with Bayesian Neural Networks. In *NeurIPS*. 2019.

APPENDIX

A Related Work

Empirical Bayes as PAC Learning. Germain et al. (2016b) propose a learnable PAC-Bayesian bound that provides generalization guarantees as a function of a marginal log-likelihood. Our method differs from this work in two main lines. First, Germain et al. (2016b) define risk as $-\log p(\mathbf{Y}|\mathbf{H}) \in (-\infty, +\infty)$ and compensate for the unboundedness by either truncating the support of the likelihood function or introducing assumptions on the data distribution, such as sub-Gaussian or sub-Gamma. Contrarily, our risk defined in (12) assumes uniform boundedness, yet can be incorporated into a PAC-Bayesian bound without further restrictions on the data. Second, Germain et al. (2016b)’s bound is an unparameterized rescaling of the marginal log-likelihood. Hence, it is not linked to a capacity penalizer, which can be used at *training time* for regularization. Applying this method to hybrid sequence modeling boils down to performing plain Empirical Bayes, i.e. *E-Bayes* in our experiments.

Differential GPs. Hegde et al. (2019) model the dynamics of the activation maps of a *feed-forward* learner by the predictive distribution of a GP. This method allocates the mean of a GP as the drift and covariance as the diffusion. It infers the resultant model using variational inference. While direct application of this method to time series modeling is not straightforward, we represent it in our experiments by sticking to our generic non-linear BNSDE design in (4), and inferring it by maximizing the ELBO: $\mathcal{L}(\phi) = \mathbb{E}_{\mathbf{H},\theta} [\log p(\mathbf{Y}|\mathbf{H})] - D_{KL}(p_\phi(\theta)||p(\theta))$, applying the local reparameterization trick on θ . Although variational inference can be seen from a PAC-perspective by choosing the log-likelihood as the loss (Knoblauch et al., 2019), the ELBO does not account for the deviation of variational posterior over latent dynamics from the prior latent dynamics. We refer to this baseline in the experiments as *D-BNN (VI)*. The approximate posterior design here closely follows the PR-SSM approach (Doerr et al., 2018), which represents the state of the art in state-space modeling.

Differential BNNs with SGLD. The learning algorithm of Look and Kandemir (2019) shares our BNSDE modeling assumptions, however, it uses Stochastic Gradient Langevin Dynamics (SGLD) to infer θ . The algorithm is equivalent to performing MAP estimation of the model parameters in (4) while distorting the gradient updates with decaying normal noise that also determines the learning rate.

Black-box identification of dynamic systems. There are various approaches to identify a dynamical system that differ in the model class used for fitting the right-hand side of the differential equation and may also allow for transitional noise (e.g. Brunton et al., 2016; Durstewitz, 2016). These approaches could be incorporated into ours, using their transition likelihood and prior over parameters. In fact, our black-box neural SDE can be seen as one instance of such a black-box identification of dynamical systems (*E-Bayes*, see below). As we are mainly interested in incorporating prior knowledge into such black-box models, we chose one such competitor (Hegde et al., 2019), which allows for the most flexible right-hand side with reported results on the CMU Motion capture data set (Tab. 2).

B Continuous Time SDEs

Solving the SDE system in (1) for a time interval $[0, T]$ and fixed θ_f requires computing integrals of the form

$$\int_0^T d\mathbf{h}_t = \int_0^T f_{\theta_f}(\mathbf{h}_t, t)dt + \int_0^T G(\mathbf{h}_t, t)dW_t.$$

This operation is intractable for almost any practical choice of $f_{\theta_f}(\cdot, \cdot)$ and $G(\cdot, \cdot)$ for two reasons. First, the integral around the drift term $f_{\theta_f}(\cdot, \cdot)$ does not have an analytical solution, due both to potential nonlinearities of the drift and to the fact that $\mathbf{h}_t \sim p(\mathbf{h}_t, t)$ is a stochastic variable following an implicitly defined distribution. Second, the diffusion term involves the Itô integral (Oksendal, 1992) about W_t which multiplies the non-linear function $G(\cdot, \cdot)$.

For each of the SDEs in (6) and (7), we could alternatively to the Euler-Maruyama integration theme use the Fokker-Planck-Kolmogorov equation to derive a partial differential equation (PDE) system

$$\begin{aligned}\partial p_{\text{hyb}}(\mathbf{h}_t, t|\theta_f)/\partial t &= -\nabla \cdot [(f_{\theta_f}(\mathbf{h}_t, t) + \gamma \circ r_\xi(\mathbf{h}_t, t))p_{\text{hyb}}(\mathbf{h}_t, t|\theta_f)] \\ &\quad + \nabla \cdot (\mathbf{1}\nabla \cdot G(\mathbf{h}_t, t)p_{\text{hyb}}(\mathbf{h}_t, t|\theta_f)), \\ \partial p_{\text{pri}}(\mathbf{h}_t, t)/\partial t &= -\nabla \cdot [(\gamma \circ r_\xi(\mathbf{h}_t, t))p_{\text{pri}}(\mathbf{h}_t, t)] + \nabla \cdot (\mathbf{1}\nabla \cdot G(\mathbf{h}_t, t)p_{\text{pri}}(\mathbf{h}_t, t)),\end{aligned}$$

where $\nabla \cdot$ is the divergence operator and $\mathbf{1} = (1, \dots, 1)^\top$. Theoretically, these distributions can be obtained by solving the Fokker-Planck PDE. As this requires solving a PDE which is not analytically tractable, we instead resort to the discrete time Euler-Maruyama integration.

C Proofs

This section gives a more detailed derivation of the individual results stated in the main paper.

Lemma 1. *For the process distributions³ $Q_{0 \rightarrow T}$ and $P_{0 \rightarrow T}$ the following property holds*

$$D_{KL}(Q_{0 \rightarrow T}||P_{0 \rightarrow T}) = \frac{1}{2} \int_0^T \mathbb{E}_{Q_{0 \rightarrow T}} [f_{\theta_f}(\mathbf{h}_t, t)^\top \mathbf{J}_t^{-1} f_{\theta_f}(\mathbf{h}_t, t)] dt + D_{KL}(p_\phi(\theta_f)||p_{\text{pri}}(\theta_f))$$

for some $T > 0$, where $\mathbf{J}_t = G(\mathbf{h}_t, t)G(\mathbf{h}_t, t)^\top$.

Proof. Assume Euler-Maruyama discretization for the process $Q_{0 \rightarrow T}$ on arbitrarily chosen K time points within the interval $[0, T]$. Then we have $D_{KL}(Q||P)$ denoting the Kullback-Leibler divergence between processes $Q_{0 \rightarrow T}$ and $P_{0 \rightarrow T}$ up to discretization into T time points as:

$$\begin{aligned}D_{KL}(Q||P) &= \iint \log \frac{\prod_{t=0}^{K-1} (\mathcal{N}(\mathbf{h}_{t+1} | (f_{\theta_f}(\mathbf{h}_t, t) + \gamma \circ r_\xi(\mathbf{h}_t, t)) \Delta t, \mathbf{J}_t \Delta t))}{\prod_{t=0}^{K-1} (\mathcal{N}(\mathbf{h}_{t+1} | \gamma \circ r_\xi(\mathbf{h}_t, t) \Delta t, \mathbf{J}_t \Delta t))} \\ &\quad \cdot \frac{p(\mathbf{h}_0)p_\phi(\theta_f)}{p(\mathbf{h}_0)p_{\text{pri}}(\theta_f)} Q_{0 \rightarrow T} d\mathbf{H}d\theta_f \\ &= \sum_{t=0}^{K-1} \iint \log \mathcal{N}(\mathbf{h}_{t+1} | (f_{\theta_f}(\mathbf{h}_t, t) + \gamma \circ r_\xi(\mathbf{h}_t, t)) \Delta t, \mathbf{J}_t \Delta t) \\ &\quad - \log \mathcal{N}(\mathbf{h}_{t+1} | \gamma \circ r_\xi(\mathbf{h}_t, t) \Delta t, \mathbf{J}_t \Delta t) Q_{0 \rightarrow T} d\mathbf{H}d\theta_f \\ &\quad + D_{KL}(p_\phi(\theta_f)||p_{\text{pri}}(\theta_f)).\end{aligned}$$

For simplicity, let us modify notation and adopt $\mathbf{f}_t := f_{\theta_f}(\mathbf{h}_t, t) + \gamma \circ r_\xi(\mathbf{h}_t, t)$, $\mathbf{g}_t := -\gamma \circ r_\xi(\mathbf{h}_t, t)$, and $\Delta \mathbf{h}_{t+1} := \mathbf{h}_{t+1} - \mathbf{h}_t$. Now writing down the $\log(\cdot)$ terms explicitly, we get

$$\begin{aligned}D_{KL}(Q||P) &= \frac{1}{2} \sum_{t=0}^{K-1} \iiint \left[-(\Delta \mathbf{h}_{t+1} - \mathbf{f}_t \Delta t)^\top (\mathbf{J}_t \Delta t)^{-1} (\Delta \mathbf{h}_{t+1} - \mathbf{f}_t \Delta t) \right. \\ &\quad \left. + (\Delta \mathbf{h}_{t+1} - \mathbf{g}_t \Delta t)^\top (\mathbf{J}_t \Delta t)^{-1} (\Delta \mathbf{h}_{t+1} - \mathbf{g}_t \Delta t) \right] \\ &\quad \cdot p_{\text{hyb}}(\mathbf{h}_{0 \rightarrow T}|\theta_f)p_\phi(\theta_f) d\mathbf{H}d\theta_f \\ &\quad + D_{KL}(p_\phi(\theta_f)||p_{\text{pri}}(\theta_f)).\end{aligned}$$

Expanding the products, removing the terms that cancel out, and rearranging the rest, we get

$$\begin{aligned}D_{KL}(Q||P) &= \frac{1}{2} \sum_{t=0}^{K-1} \iiint \left[-\mathbf{f}_t^\top \mathbf{J}_t^{-1} \mathbf{f}_t \Delta t + 2\Delta \mathbf{h}_{t+1} \mathbf{J}_t^{-1} \mathbf{f}_t + \mathbf{g}_t^\top \mathbf{J}_t^{-1} \mathbf{g}_t \Delta t - 2\Delta \mathbf{h}_{t+1} \mathbf{J}_t^{-1} \mathbf{g}_t \right] \\ &\quad \cdot p_{\text{hyb}}(\mathbf{h}_{0 \rightarrow T}|\theta_f)p_\phi(\theta_f) d\mathbf{H}d\theta_f \\ &\quad + D_{KL}(p_\phi(\theta_f)||p_{\text{pri}}(\theta_f)).\end{aligned}$$

³See the main paper for their definitions.

Note that from the definition of the process it follows that

$$\int \Delta \mathbf{h}_{t+1} p_{\text{hyb}}(\mathbf{h}_{0 \rightarrow T} | \theta_f) d\Delta \mathbf{h}_{t+1} = \mathbf{f}_t \Delta t.$$

Plugging this fact into the KL term, we have

$$\begin{aligned} D_{KL}(Q||P) &= \frac{1}{2} \sum_{t=0}^{K-1} \int \left[\mathbf{f}_t^\top \mathbf{J}_t^{-1} \mathbf{f}_t \Delta t + \mathbf{g}_t^\top \mathbf{J}_t^{-1} \mathbf{g}_t \Delta t - 2\mathbf{f}_t^\top \mathbf{J}_t^{-1} \mathbf{g}_t \Delta t \right] p_{\phi_f}(\theta_f) d\theta_f \\ &\quad + D_{KL}(p_\phi(\theta_f) || p_{\text{pri}}(\theta_f)). \end{aligned}$$

For any pair of vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^P$ and symmetric matrix $\mathbf{C} \in \mathbb{R}^{P \times P}$, the following identity holds:

$$\mathbf{a}^\top \mathbf{C} \mathbf{a} - \mathbf{b}^\top \mathbf{C} \mathbf{b} = (\mathbf{a} - \mathbf{b})^\top \mathbf{C} (\mathbf{a} - \mathbf{b}) + 2\mathbf{a}^\top \mathbf{C} \mathbf{b}.$$

Applying this identity to the above, we attain

$$D_{KL}(Q||P) = \frac{1}{2} \sum_{t=0}^{K-1} \int \left[(\mathbf{f}_t - \mathbf{g}_t)^\top \mathbf{J}_t^{-1} (\mathbf{f}_t - \mathbf{g}_t) \Delta t \right] p_{\phi_f}(\theta_f) d\theta_f + D_{KL}(q(\theta_f) || p(\theta_f)).$$

Plugging back the original terms and setting K to the limit, we arrive at the desired outcome

$$\begin{aligned} &\lim_{K \rightarrow +\infty} \left\{ \frac{1}{2} \sum_{t=0}^{K-1} \int \left[(f_{\theta_f}(\mathbf{h}_t, t))^\top \mathbf{J}_t^{-1} f_{\theta_f}(\mathbf{h}_t, t) \Delta t \right] p_{\phi_f}(\theta_f) d\theta_f + D_{KL}(p_\phi(\theta_f) || p_{\text{pri}}(\theta_f)) \right\} \\ &= \frac{1}{2} \int \left[\int f_{\theta_f}(\mathbf{h}_t, t)^\top \mathbf{J}_t^{-1} f_{\theta_f}(\mathbf{h}_t, t) p_{\phi_f}(\theta_f) d\theta_f \right] dt + D_{KL}(p_\phi(\theta_f) || p_{\text{pri}}(\theta_f)) \\ &= \frac{1}{2} \int_0^T \mathbb{E}_{Q_{0 \rightarrow T}} \left[f_{\theta_f}(\mathbf{h}_t, t)^\top \mathbf{J}_t^{-1} f_{\theta_f}(\mathbf{h}_t, t) \right] dt + D_{KL}(p_\phi(\theta_f) || p_{\text{pri}}(\theta_f)). \end{aligned}$$

□

Theorem 1. Let $p(\mathbf{y}_t | \mathbf{h}_t)$ be uniformly bounded likelihood function with density $p(\mathbf{y}_t | \mathbf{h}_t)$ everywhere and $Q_{0 \rightarrow T}$ and $P_{0 \rightarrow T}$ be the joints stochastic processes defined on the hypothesis class of the learning task, respectively. Define the true risk of a draw from $Q_{0 \rightarrow T}$ on an i.i.d. sample $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_K\}$ at discrete and potentially irregular time points t_1, \dots, t_K drawn from an unknown ground-truth stochastic process $\mathfrak{G}(t)$ as the expected model misfit as on the sample as defined via the following risk over hypotheses $H = (\mathbf{h}_{0 \rightarrow T}, \theta_f)$

$$R(H) = -\mathbb{E}_{\mathbf{Y} \sim \mathfrak{G}(t)} \left[\prod_{k=1}^K p(\mathbf{y}_k | \mathbf{h}_k) \right], \quad (17)$$

for time horizon $T > 0$ and the corresponding empirical risk on a data set $\mathcal{D} = \{\mathbf{Y}_1, \dots, \mathbf{Y}_N\}$ as

$$R_{\mathcal{D}}(H) = -\frac{1}{N} \sum_{n=1}^N \left[\prod_{k=1}^K p(\mathbf{y}_k^n | \mathbf{h}_k) \right]. \quad (18)$$

Then the expected true risk is bounded above by the marginal negative log-likelihood of the predictor and a complexity functional as

$$\mathbb{E}_{H \sim Q_{0 \rightarrow T}} [R(H)] \leq \mathbb{E}_{H \sim Q_{0 \rightarrow T}} [R_{\mathcal{D}}(H)] + \mathcal{C}_\delta(Q_{0 \rightarrow T}, P_{0 \rightarrow T}), \quad (19)$$

$$\leq -\frac{1}{N} \sum_{n=1}^N \log \left(\frac{1}{S} \sum_{s=1}^S \prod_{k=1}^K p(\mathbf{y}_k^n | \mathbf{h}_k^{n,s}) \right) + \mathcal{C}_{\delta/2}(Q_{0 \rightarrow T}, P_{0 \rightarrow T}) + \sqrt{\frac{\log(2N/\delta)}{2S}} + K \log \bar{B} \quad (20)$$

$$\leq -\frac{1}{SN} \sum_{n=1}^N \sum_{s=1}^S \sum_{k=1}^K \log \left(p(\mathbf{y}_k^n | \mathbf{h}_k^{s,n}) \right) + \mathcal{C}_{\delta/2}(Q_{0 \rightarrow T}, P_{0 \rightarrow T}) + \sqrt{\frac{\log(2N/\delta)}{2S}} + K \log \bar{B}, \quad (21)$$

where $\bar{B} := \max_{\mathbf{y}_k, \mathbf{h}_k} p(\mathbf{y}_k | \mathbf{h}_k)$ is the uniform bound, S is the sample count taken independently for each observed sequence, and the complexity functional is given as

$$\mathcal{C}_\delta(Q_{0 \rightarrow T}, P_{0 \rightarrow T}) := \sqrt{\frac{D_{KL}(Q_{0 \rightarrow T} || P_{0 \rightarrow T}) + \log(2\sqrt{N}) - \log(\delta/2)}{2N}}$$

with $D_{KL}(Q_{0 \rightarrow T} || P_{0 \rightarrow T})$ as in Lemma 1 for some $\delta > 0$.

Proof. To be able to apply known PAC bounds, we first define the hypothesis class $H \in \mathcal{H}_K$ that contain latent states \mathbf{h}_k, θ_f that explain the observations \mathbf{y}_k . Then, we define the true risk as

$$R(H) = \mathbb{E}_{\mathbf{Y}_k \sim \mathfrak{G}(t)} \left[1 - \frac{1}{\bar{B}_K} \prod_{k=1}^K p(\mathbf{y}_k | \mathbf{h}_k) \right]$$

and the empirical risk as

$$R_{\mathcal{D}}(H) = \frac{1}{N} \sum_{n=1}^N \left\{ 1 - \frac{1}{\bar{B}_K} \prod_{k=1}^K p(\mathbf{y}_k^n | \mathbf{h}_k^n) \right\},$$

where we defined

$$\bar{B}_K := \max_{\mathbf{y}, \mathbf{h}_k} \prod_{k=1}^K p(\mathbf{y}_k | \mathbf{h}_k) \leq \left(\max_{\mathbf{y}, \mathbf{h}_k} p(\mathbf{y}_k | \mathbf{h}_k) \right)^K.$$

The data set $\mathcal{D} = \{\mathbf{Y}_k^n\}_{k,n}$ was generated by an unknown stochastic process $\mathfrak{G}(t)$. Note that we normalize the risks $R_{\mathcal{D}}(H)$ and $R(H)$ by the maximum of the likelihood and thereby obtaining a possible range of these risk of $[0, 1]$. The likelihood can be bounded, as the term $p(\mathbf{y}_k | \mathbf{h}_k)$ can be bounded from above, as we model this by a Gaussian. Therefore, it is bounded, if we assume a minimal allowed variance.

To obtain a tractable bound, it is common practice is to upper bound its analytically intractable inverse (Germain et al., 2016b) using Pinsker's inequality (Catoni, 2007; Dziugaite and Roy, 2017). Here, we rely on the following theorem.

PAC-theorem (Maurer, 2004) For any $[0, 1]$ -valued loss function giving rise to empirical and true risk $R_{\mathcal{D}}(H), R(H)$, for any distribution Δ , for any $N \in \mathbb{N}$, for any distribution $P_{0 \rightarrow T}$ on a hypothesis set \mathcal{Q}_K , and for any $\delta \in (0, 1]$, the following holds with probability at least $1 - \delta$ over the training set $\mathcal{D} \sim \Delta^N$:

$$\forall Q_{0 \rightarrow T} : \mathbb{E}_{H \sim Q_{0 \rightarrow T}} [R(H)] \leq \mathbb{E}_{H \sim Q_{0 \rightarrow T}} [R_{\mathcal{D}}(H)] + \sqrt{\frac{\text{KL}(Q_{0 \rightarrow T} \parallel P_{0 \rightarrow T}) + \log\left(\frac{2\sqrt{N}}{\delta}\right)}{2N}}$$

Here, $\text{KL}(Q_{0 \rightarrow T} \parallel P_{0 \rightarrow T})$ acts as a complexity measure that measures, how much the posterior predictive governing the SDE $Q_{0 \rightarrow T}$ needed to be adapted to the data when compared to an a priori chosen SDE that could alternatively have generated data $P_{0 \rightarrow T}$. In our situation, $Q_{0 \rightarrow T}$ is obtained by our approximation scheme, resulting in a bounded likelihood of observations \mathbf{y}_k which factorizes over different observations n . The $P_{0 \rightarrow T}$ can be arbitrarily chosen as long as it does not depend on the observations. As mentioned in the main paper, we chose an SDE with the same diffusion term which also factorizes over observations. Using this setting, we can analytically compute the KL-distance (as shown in Lemma 1).

On the right hand side of this PAC-bound, we need to evaluate $\mathbb{E}_{H \sim Q_{0 \rightarrow T}} [R_{\mathcal{D}}(H)]$. To this end, we note

$$\begin{aligned} \mathbb{E}_{H \sim Q_{0 \rightarrow T}} [R_{\mathcal{D}}(H)] &= \frac{1}{N} \sum_{n=1}^N \mathbb{E}_{H \sim Q_{0 \rightarrow T}} \left[1 - \frac{1}{\bar{B}_K} \left(\prod_{k=1}^K p(\mathbf{y}_k^n | \mathbf{h}_k^n) \right) \right] \\ &= 1 - \frac{1}{N} \sum_{n=1}^N \mathbb{E}_{H \sim Q_{0 \rightarrow T}} \left[\frac{1}{\bar{B}_K} \prod_{k=1}^K p(\mathbf{y}_k^n | \mathbf{h}_k^n) \right] \\ &\stackrel{\text{Hoeffding}}{\leq} 1 - \frac{1}{SN} \sum_{n=1}^N \sum_{s=1}^S \left[\frac{1}{\bar{B}_K} \prod_{k=1}^K p(\mathbf{y}_k^n | \mathbf{h}_k^{n,s}) \right] + \sqrt{\frac{\log(2N/\delta)}{2S}} \\ &= \frac{1}{N} \sum_{n=1}^N \left\{ 1 - \frac{1}{S} \sum_{s=1}^S \left[\frac{1}{\bar{B}_K} \prod_{k=1}^K p(\mathbf{y}_k^n | \mathbf{h}_k^{n,s}) \right] \right\} + \sqrt{\frac{\log(2N/\delta)}{2S}} \\ &\stackrel{-\log(z) \geq 1-z}{\leq} -\frac{1}{N} \sum_{n=1}^N \log \left(\frac{1}{S} \sum_{s=1}^S \prod_{k=1}^K p(\mathbf{y}_k^n | \mathbf{h}_k^{n,s}) \right) + \log \bar{B}_K + \sqrt{\frac{\log(2N/\delta)}{2S}} \end{aligned}$$

$$\stackrel{\text{Jensen's ineq.}}{\leq} -\frac{1}{SN} \sum_{n=1}^N \sum_{s=1}^S \sum_{k=1}^K [\log p(\mathbf{y}_k^n | \mathbf{h}_k^{n,s})] + \log \bar{B}_K + \sqrt{\frac{\log(2N/\delta)}{2S}},$$

where we have used Hoeffding's inequality for estimating the true expectation over hypotheses with a K samples trace $\mathbf{h}_k^{n,s}$, $k = 1, \dots, K$, $s = 1, \dots, S$ for each observation n . As we approximate the integral for each time-series n separately via sampling, we require Hoeffding to hold simultaneously for all n . Using a union bound, we have to scale δ for each n by N . Splitting confidences between the PAC-bound and the sampling based approximation results an additional factor of 2. With $\delta/(2N)$, the corresponding inequality holds with a probability of $\mathbb{P} > \delta/2$. Also using $\delta/2$ in PAC-theorem, we obtain that with $\mathbb{P} \geq 1 - \delta$ we have for all $Q_{0 \rightarrow T}$ that

$$\begin{aligned} \mathbb{E}_{H \sim Q_{0 \rightarrow T}} [R(H)] &\leq \mathbb{E}_{H \sim Q_{0 \rightarrow T}} [R_{\mathcal{D}}(H)] + \sqrt{\frac{\text{KL}(Q_{0 \rightarrow T} \| P_{0 \rightarrow T}) + \log\left(\frac{2\sqrt{N}}{\delta/2}\right)}{2N}} \\ &\leq -\frac{1}{N} \sum_{n=1}^N \log\left(\frac{1}{S} \sum_{s=1}^S \prod_{k=1}^K p(\mathbf{y}_k^n | \mathbf{h}_k^{n,s})\right) + \sqrt{\frac{\text{KL}(Q_{0 \rightarrow T} \| P_{0 \rightarrow T}) + \log\left(\frac{2\sqrt{N}}{\delta/2}\right)}{2N}} \\ &\quad + \log \bar{B}_K + \sqrt{\frac{\log(2N/\delta)}{2S}} \\ &\leq -\frac{1}{SN} \sum_{n=1}^N \sum_{s=1}^S \sum_{k=1}^K [\log p(\mathbf{y}_k^n | \mathbf{h}_k^{n,s})] + \sqrt{\frac{\text{KL}(Q_{0 \rightarrow T} \| P_{0 \rightarrow T}) + \log\left(\frac{2\sqrt{N}}{\delta/2}\right)}{2N}} \\ &\quad + \log \bar{B}_K + \sqrt{\frac{\log(2N/\delta)}{2S}} \end{aligned}$$

□

Corollary 1. *Given a L -Lipschitz continuous function set*

$$\left\{ f_{\theta}^n(x) : \mathbb{R} \rightarrow [0, 1] \mid n = 1, \dots, N \right\} \cup \left\{ g_{\theta}(x) : \mathbb{R} \rightarrow [0, +\infty] \right\},$$

for the two losses:

$$l_1(\theta) = -\sum_{n=1}^N f_{\theta}^n(x) + g_{\theta}(x) \quad \text{and} \quad l_2(\theta) = -\sum_{n=1}^N \log f_{\theta}^n(x) + g_{\theta}(x),$$

the sequential updates ($\theta^0 := \theta$)

$$\begin{aligned} \theta^{(n)} &\leftarrow \theta^{(n-1)} + \alpha_n \nabla(\log f_{\theta^{(n-1)}}^n(x)), \quad n = 1, \dots, N, \\ \theta^{(N+1)} &\leftarrow \theta^{(N)} - \alpha_{N+1} \nabla g_{\theta^{(N)}}(x), \end{aligned}$$

where $\alpha_n \in (0, f_{\theta^{(n-1)}}^n(x)/L) \forall n$ and $\alpha_{N+1} \in (0, 1/L)$, satisfy both $l_1(\theta^{(N+1)}) \leq l_1(\theta)$ and $l_2(\theta^{(N+1)}) \leq l_2(\theta)$.

Proof. As we only consider updates in θ for constant x , we simplify the notation for this proof to $f^n(\theta) := f_{\theta}^n(x)$, $g(\theta) = g_{\theta}(x)$. I.e. we have as the two loss terms

$$l_1(\theta) = -\sum_{n=1}^N f^n(\theta) + g(\theta) \quad \text{and} \quad l_2(\theta) = -\sum_{n=1}^N \log f^n(\theta) + g(\theta).$$

In general we have with $\log f(\theta) < f(\theta)$ that $l_1(\theta) < l_2(\theta)$. Similarly we have

$$\nabla l_2(\theta) = -\sum_n \underbrace{\frac{1}{f^n(\theta)}}_{\geq 1} \nabla f^n(\theta) + \nabla g(\theta) \leq -\sum_n \nabla f^n(\theta) + \nabla g(\theta) = \nabla l_1(\theta).$$

Due to the sequential updates we can consider each term separately. For an L -Lipschitz function $f^n(\theta)$, we have that for arbitrary x, y

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2} \|y - x\|_2^2.$$

Choosing $y = \theta^{(n-1)}$ and $x = \theta^{(n)} = \theta^{(n-1)} + \alpha_n \nabla \log f^n$ this gives us

$$\begin{aligned} f(\theta^{(n-1)}) &\leq f(\theta^{(n)}) - \frac{\alpha_n}{f^n(\theta^{(n)})} \|\nabla f^n(\theta^{(n)})\|_2^2 + \frac{L\alpha_n^2}{2f^n(\theta^{(n)})^2} \|\nabla f^n(\theta^{(n)})\|_2^2 \\ &= f^n(\theta^{(n)}) - \underbrace{\frac{\alpha_n}{f^n(\theta^{(n)})}}_{\geq 0} \underbrace{\left(1 - \frac{L\alpha_n}{2f^n(\theta^{(n)})}\right)}_{> 0} \|\nabla f^n(\theta^{(n)})\|_2^2 \leq f^n(\theta^{(n)}), \end{aligned}$$

and hence chaining the update steps gives the desired result. \square

That is, updating the terms in $l_2(\theta)$ sequentially, one can ensure concurrent optimization of $l_1(\theta)$. Note that $l_1(\theta)$ and $l_2(\theta)$ are not necessarily dual objectives, hence may have different extrema. Nevertheless, a gradient step that decreases one loss also decreases the other with potentially a different magnitude. In practice, we observe this behavior to also hold empirically for joint gradient update steps with shared learning rates. Applying Lemma 2 to the setup in Theorem 2, we establish a useful link between Empirical Bayes and PAC learning.

Theorem 2 (strong convergence). *Let \mathbf{h}_t^θ be an Itô process as in (4) with drift parameters θ and its Euler-Maruyama approximation $\tilde{\mathbf{h}}_t^\theta$ for some regular step size $\Delta t > 0$. For some coefficient $R > 0$ and any $T > 0$, the following inequality holds*

$$\mathbb{E} \left[\sup_{0 \leq t \leq T} \left| \mathbb{E}_\theta[\mathbf{h}_t^\theta] - \frac{1}{S} \sum_{s=1}^S \tilde{\mathbf{h}}_t^{\theta^{(s)}} \right| \right] \leq R\Delta t^{1/2},$$

as $S \rightarrow \infty$, where $\{\theta^{(s)} \sim p_\phi(\theta_f) | s = 1, \dots, S\}$ are i.i.d. draws from a prior $p_\phi(\theta_f)$.

Proof: The Euler-Maruyama (EM) approximation converges strongly as

$$\mathbb{E} \left[|\mathbf{h}_T^\theta - \tilde{\mathbf{h}}_T^\theta| \right] \leq R\Delta t^{1/2},$$

for a positive constant R and a suitably small step size Δt as discussed e.g. by [Kloeden and Platen \(2011\)](#). To simplify the mathematical notation we follow their approach of comparing the absolute error of the end of the trajectory throughout the proof. As our sampling scheme is unbiased it is a consistent estimator and we have that asymptotically for $S \rightarrow \infty$

$$\frac{1}{S} \sum_{s=1}^S \tilde{\mathbf{h}}_T^{\theta^{(s)}} = \mathbb{E}_\theta[\tilde{\mathbf{h}}_T^\theta].$$

We then have for the marginal $\mathbf{h}_T, \tilde{\mathbf{h}}_T$ that

$$\begin{aligned} \mathbb{E} \left[|\mathbf{h}_T - \tilde{\mathbf{h}}_T| \right] &= \mathbb{E} \left[|\mathbb{E}_\theta \mathbf{h}_T^\theta - \mathbb{E}_\theta \tilde{\mathbf{h}}_T^\theta| \right] = \mathbb{E} \left[|\mathbb{E}_\theta [\mathbf{h}_T^\theta - \tilde{\mathbf{h}}_T^\theta]| \right] \\ &\leq \mathbb{E} \left[\mathbb{E}_\theta \left[|\mathbf{h}_T^\theta - \tilde{\mathbf{h}}_T^\theta| \right] \right] \leq \mathbb{E}_\theta \left[R\Delta t^{1/2} \right] = R\Delta t^{1/2}, \end{aligned}$$

where the first inequality is due to Jensen and the second due to the strong convergence result for a fixed set of parameters. \square

D Computational Cost

We present the runtimes of the different approaches in Table 3. D-BNN samples the weights of the neural network directly leading to the runtime term $\mathcal{O}(MTF)$. All other approaches do not sample the weights but the linear activations of the each data points leading to $\mathcal{O}(2MTF)$. When we apply empirical Bayes, we do not use any regularization term on the weights, while all other

approaches contain a penalty term with cost $\mathcal{O}(W)$. Using the PAC-framework, we employ a second regularization term that leads to an additional runtime cost of $\mathcal{O}(TMD^3)$. However, the cubic cost in D is invoked by inverting the diffusion matrix $G(h_t, t)$ and can be further reduced by choosing a simpler form for $G(h_t, t)$ (e.g. diagonal). In case that prior knowledge is available in ODE form, we need to compute the corresponding drift term for each time point and each MC sample leading to the term $\mathcal{O}(MTP)$.

Table 3: Computational cost analysis in FLOPs for time series of length \mathbf{T} . \mathbf{M} : Number of Monte Carlo Samples. \mathbf{W} : Number of weights in the neural net. \mathbf{F} : Forward pass cost of a neural net. \mathbf{L} : Cost for computing the likelihood term. \mathbf{D} : Number of dimensions. \mathbf{P} : Cost of a prior SDE integration.

Model	Training per Iteration
D-BNN (SGLD)	$\mathcal{O}(MTF + MTDL + W)$
Variational Bayes	$\mathcal{O}(2MTF + MTDL + W)$
E-Bayes	$\mathcal{O}(2MTF + MTDL)$
E-PAC-Bayes	$\mathcal{O}(2MTF + MTDL + W + TMD^3)$
E-Bayes-Hybrid	$\mathcal{O}(2MTF + MTDL + MTP)$
E-PAC-Bayes-Hybrid	$\mathcal{O}(2MTF + MTDL + W + TMD^3 + MTP)$

E The Algorithm and a Visualization of the Pipeline

A pseudo-code algorithm of the proposed pipeline is given in Algorithm 1. Figure 2 summarizes the proposed method in an illustration.

F Further Details on The Experiments

Here we provide the details of the experiment setup we used in obtaining our results reported in the main paper. We observed our results to be robust against most of the design choices.

F.1 Lorenz Attractor

We took 200000 Euler-Maruyama steps ahead with a time step size of 10^{-4} and downsampling by factor 0.01, which gives a sequence of 2000 observations with frequency 0.01. We split the first half of this data set into 20 sequences of length 50 and use them for training, and the second half to 10 sequences of length 100 and use for test. For all model variants, we used an Adam optimizer learning rate 0.001, minibatch size of two, a drift net with two hidden layers of 100 neurons and softplus activation function. We trained all models for 100 epochs and observed this training period to be sufficient for convergence.

F.2 CMU Motion Capture

In this experiment, we tightly follow the design choices reported by Yildiz et al. (2019) to maintain commensurateness. This setup assumes the stochastic dynamics are determined in a six-dimensional latent space. Yildiz et al. (2019) use an auto-encoder to map this latent space to the 50-dimensional observation space back and forth. We adopt their exact encoder-decoder architecture and incorporate it into our BNSDE, arriving at the data generating process

$$\begin{aligned}
 \theta_f &\sim p_{\phi_f}(\theta_f), \\
 d\mathbf{h}_t | \theta_f &\sim f_{\theta_f}(b_\lambda(\mathbf{h}_t), t)dt + G(b_\lambda(\mathbf{h}_t), t)d\beta_t, \\
 \mathbf{z}_t | \mathbf{h}_t &\sim \mathcal{N}(\mathbf{z}_t | a_\psi(\mathbf{h}_t), 0.5 \cdot 10^{-6} \mathbf{1}), \\
 \mathbf{y}_t | \mathbf{z}_t &\sim \mathcal{N}(\mathbf{y}_t | \mathbf{z}_t, 0.5 \cdot 10^{-6} \mathbf{1}), \quad \forall t \in \mathbf{t}.
 \end{aligned}$$

Above, $b_\lambda(\cdot, \cdot)$ is the encoder which takes the observations of the last three time points as input, passes them through two dense layers with 30 neurons and softplus activation function, and then

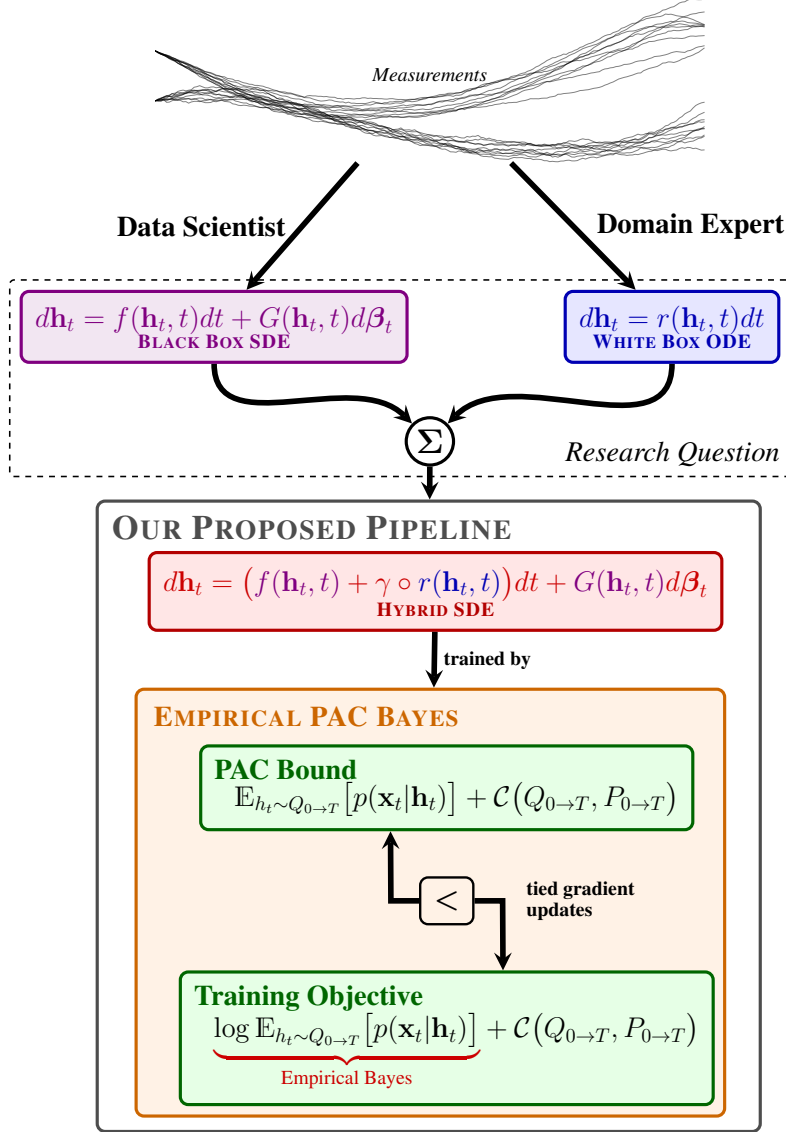


Figure 2: Illustration of the research question we pose (above) and our proposed solution (below).

linearly projects them to a six-dimensional latent space, where the dynamics are modeled. The decoder $a_{\psi}(\mathbf{h}_t)$ follows the same chain of mapping operations in reverse order. The only difference is that the output layer of the decoder emits only one observation point, as opposed to the encoder admitting three points at once.

The drift function $f_{\theta_f}(\cdot, \cdot)$ is governed by another separate Bayesian neural net, again with one hidden layer of 30 neurons and softplus activation function on the hidden layer. The diffusion function is fixed to be a constant.

We train all models except SGLD with the Adam optimizer for 3000 epochs on seven randomly chosen snippets at a time with a learning rate of 10^{-3} . We use snippet length 30 for the first 1000 epochs, 50 until epoch 2500, and 100 afterwards. SGLD demonstrates significant training instability for this learning rate, hence for it we drop its learning rate to the largest possible stable value 10^{-5} and increase the epoch count to 5000.

Algorithm 1: E-PAC-Bayes-Hybrid Loss

Input: set of N trajectories \mathcal{D} , prior drift $r_\xi(\cdot, \cdot)$, time points \mathbf{t} , drift $f_{\theta_f}(\cdot, \cdot)$, diffusion $G(\cdot, \cdot)$, weight distribution $p_\phi(\theta_f)$, number of samples S , prior parameter γ

Output: training objective loss

```
// init. marginal log-likelihood (mll) and kl
mll  $\leftarrow$  0; kl  $\leftarrow$  0
for  $n \in \{1, \dots, N\}$  do // for each trajectory
  for  $s \in \{1, \dots, S\}$  do // and each sample
    // sample initial state and weights
     $\mathbf{h}_0^{n,s} \sim p(\mathbf{h}_0)$ 
     $\theta_f^{n,s} \sim p_\phi(\theta_f)$ 
    // for each of the  $K$  steps
    for  $k \in \{1, \dots, K\}$  do
      // get drift, prior, diffusion output
       $f_k^{n,s} \leftarrow f_{\theta_f}^{n,s}(\mathbf{h}_{k-1}^{n,s}, t_{k-1})$ 
       $r_k^{n,s} \leftarrow r_\xi(\mathbf{h}_{k-1}^{n,s}, t_{k-1})$ 
       $G_k^{n,s} \leftarrow G(\mathbf{h}_{k-1}^{n,s}, t_{k-1})$ 
      // sample stochasticity
       $\Delta t_k \leftarrow t_k - t_{k-1}$ 
       $W_k^{n,s} \sim \mathcal{N}(0, \Delta t_k \mathbf{1})$ 
      // update state
       $\mathbf{h}_k^{n,s} \leftarrow \mathbf{h}_{k-1}^{n,s} + (f_k^{n,s} + \gamma r_k^{n,s}) \Delta t_k + G_k^{n,s} W_k^{n,s}$ 
      // and update mll and kl
      mll  $\leftarrow$  mll +  $\frac{1}{SN} \log p(\mathbf{y}_k^n | \mathbf{h}_k^{n,s})$ 
      kl  $\leftarrow$  kl +  $\frac{1}{2S} f_k^{n,s \top} (G_k^{n,s} G_k^{n,s \top})^{-1} f_k^{n,s} \Delta t_k$ 
    end
  end
end
// add penalty for modified drift distribution
kl  $\leftarrow$  kl +  $D_{KL}(p_\phi(\theta_f) || p_{\text{pri}}(\theta_f))$ 
// and assign final loss
loss  $\leftarrow$  -mll +  $\sqrt{(\text{kl} + \log(4\sqrt{N}/\delta)) / (2N)}$ 
// to be returned and optimized
return loss
```

G Further Experiments

G.1 Lotka Volterra

We demonstrate the benefits of incorporating prior knowledge although it is a coarse approximation to the true system. We consider the Lotka-Volterra system specified as:

$$\begin{aligned} dx_t &= (\theta_1 x_t - \theta_2 x_t y_t) dt + 0.2 d\beta_t, \\ dy_t &= (-\theta_3 y_t + \theta_4 x_t y_t) dt + 0.3 d\beta_t. \end{aligned}$$

with $\theta = (2.0, 1.0, 4.0, 1.0)$. Assuming that the trajectory is observed on the interval $t = [0, 1]$ with a resolution of $dt = 0.01$, we compare the following three methods: *i*) the black-box BNSDE without prior knowledge, *ii*) the white-box SDE in (7) representing partial prior knowledge (parameters are sampled from a normal distribution centered on the true values with a standard deviation of 0.5), and finally *iii*) combining them in our proposed hybrid method. The outcome is summarized in Figure 3. While the plain black-box model delivers a poor fit to data, our hybrid BNSDE brings significant improvement from relevant but inaccurate prior knowledge.

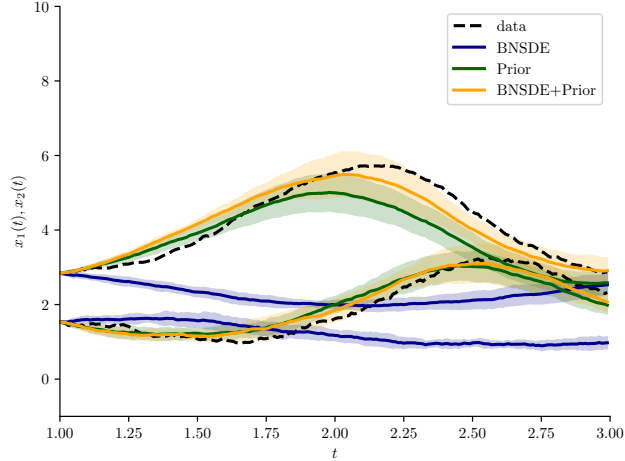


Figure 3: Lotka-Volterra visualization. Error bars indicate three standard deviations over 10 trajectories starting from the true value at $t = 1$. The predictions over 200 time steps ($dt = 0.01$) are for: *i*) a BNSDE trained without prior knowledge, *ii*) an SDE with known prior parameters, *iii*) the joint hybrid BNSDE. The dashed lines are the observed trajectories for x_t and y_t .

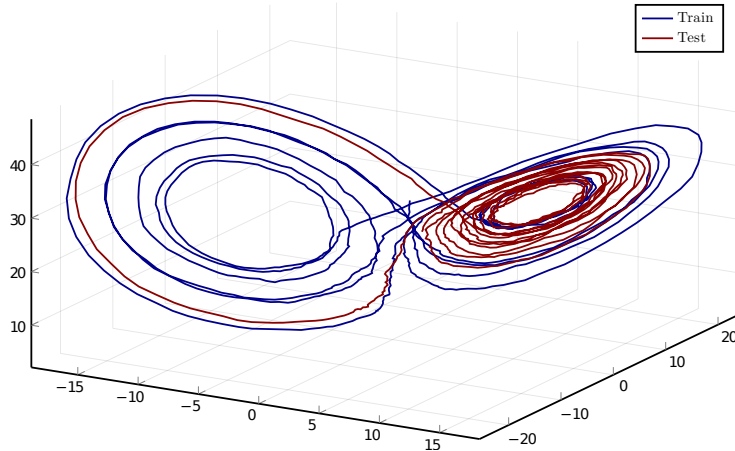


Figure 4: Visualization of the stochastic Lorenz attractor. Of the 2000 observations, the first 1000 constitute the training data (marked in blue), while the second 1000 are the test observations (marked in red). Note the qualitative difference of the two sets.

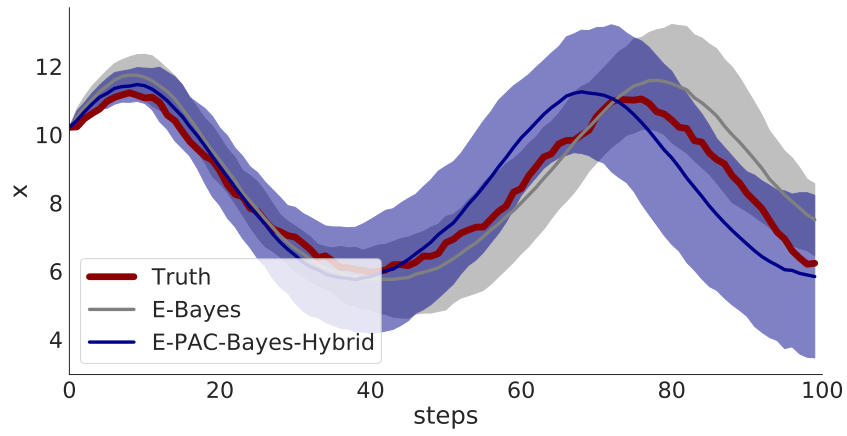
G.1.1 Experimental details

We took 10^5 Euler-Maruyama steps on the interval $[0, 10]$ with a time step size of 10^{-4} , downsampling them by a factor of 100 giving us 1000 observations with a frequency of 0.01. We take the first 500 observations on the interval $[0, 5]$ to be the training data and the observations in $(5, 10]$ to be the test data. Each sequence is split into ten sequences of length 50. Assuming the diffusion parameters to be known and fixed, both BNSDEs (i.e. with and without prior knowledge) get a 4 layer net as the drift function with 50 neurons per layer and ReLU activation functions. The BNSDE with prior knowledge as well as the raw SDE estimate each get an initial sample of $\tilde{\theta}$ parameters as the prior information by sampling from a normal distribution centered around the true parameters ($\tilde{\theta} \sim \mathcal{N}(\tilde{\theta}, \sigma^2 \mathbb{1}_4)$). The models are each trained for 50 epochs with the Adam optimizer and a learning rate of $1e-3$. Since both the latent and observed spaces are only two dimensional, we did not need an observation model in this experiment. We directly linked the BNSDE to the likelihood.

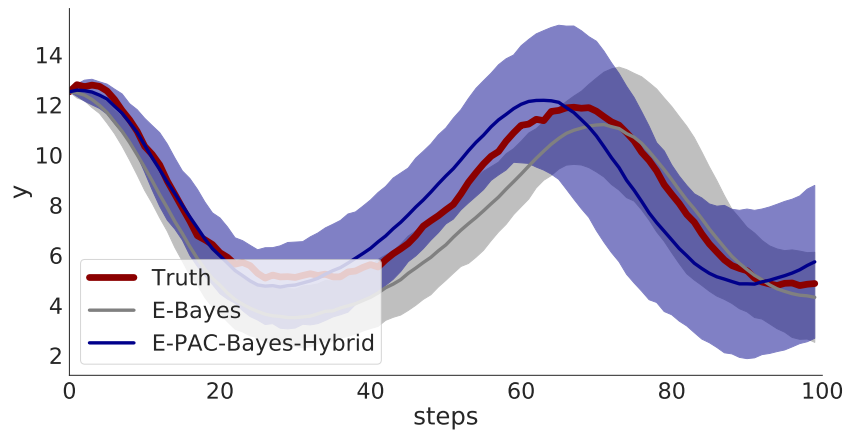
G.2 Lorenz Attractor

As discussed in the main paper, the model is trained solely on the first 1000 observations of a trajectory consisting of 2000 observations, leaving the second half for the test evaluation. Figure 4 visualizes the qualitative difference between the two. Note also the single loop the trajectory performs which we will see again in the 1d projections below. To visualize explore the qualitative difference of our proposed model with weak prior knowledge compared to one lacking this knowledge we consider the situation where we have structural prior knowledge only about the third SDE (i.e. the penultimate case in Table 1 with $\rho = [0, 0, 1]$).

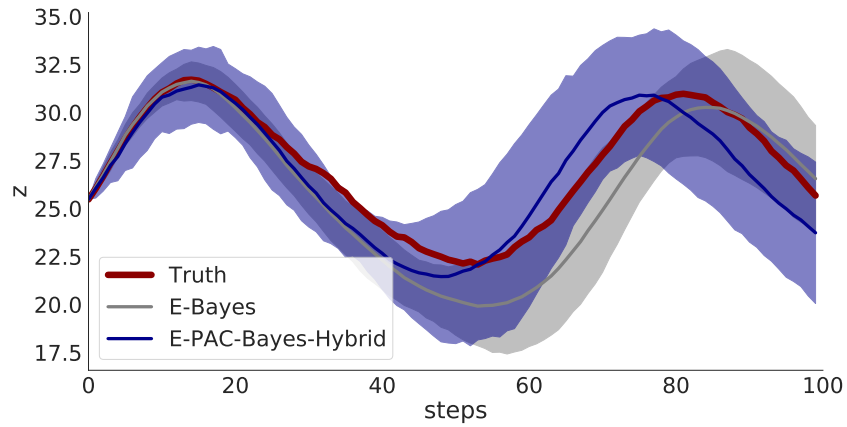
In order to properly visualize it we switch from the 3d plot to 1d plots showing always one of the three dimensions vs the time component. We always start at $T = 10$, forecasting either 100 steps (as in the numerical evaluation), 200 or 1000 steps. All the following figures show the mean trajectory averaged over 21 trajectories, as well as an envelope of ± 2 standard deviations. Figure 5 visualizes that at that time scale the qualitative behavior is similar without clear differences. Doubling the predicted time interval as shown in Figure 6 the baseline starts to diverge from the true test sequence, while our proposed model still tracks it closely but at an increased variance. Finally predicting for 1000 time steps (Figure 7) the chaotic behavior of the Lorenz attractor becomes visible as the mean in both setups no longer tracks the true trajectory. Note however that the baseline keeps a rather small variance and a strong tendency in its predictions that do not replicate the qualitative behavior of the Lorenz attractor. While the proposed model also shows an unreliable average, the large variance, which nearly always includes the true trajectory shows that the qualitative behavior is still replicated properly by individual trajectories of the model. See Figure 8 for seven individual trajectories of each of the two models. All trajectories of *E-PAC-Bayes-Hybrid* show the qualitatively correct behavior, including even the characteristic loop.



(a) x coordinate over time

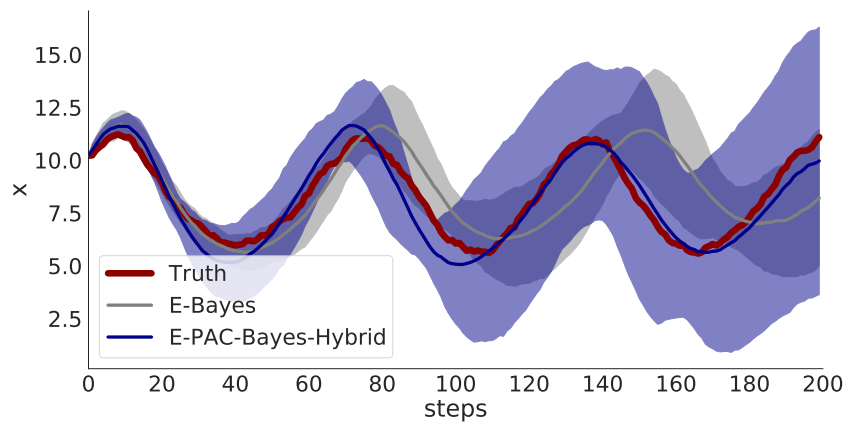


(b) y coordinate over time

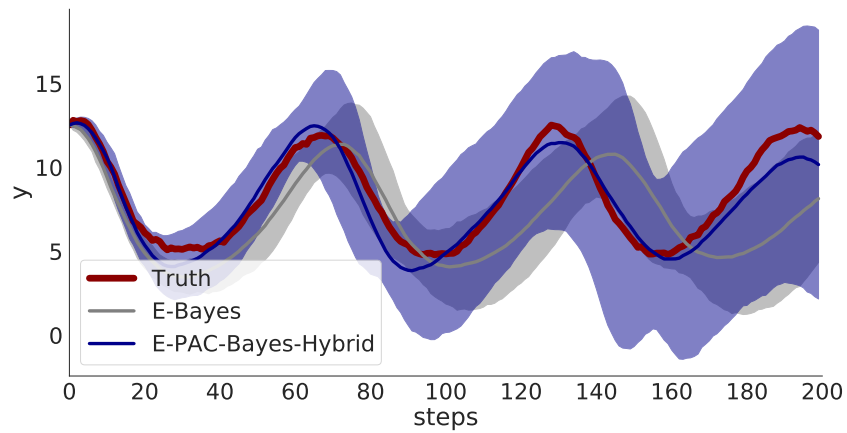


(c) z coordinate over time

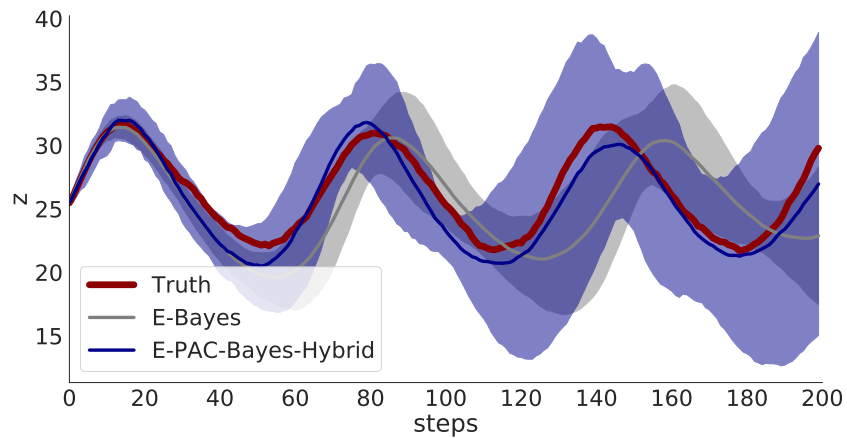
Figure 5: Predicting 100 time steps ahead.



(a) x coordinate over time

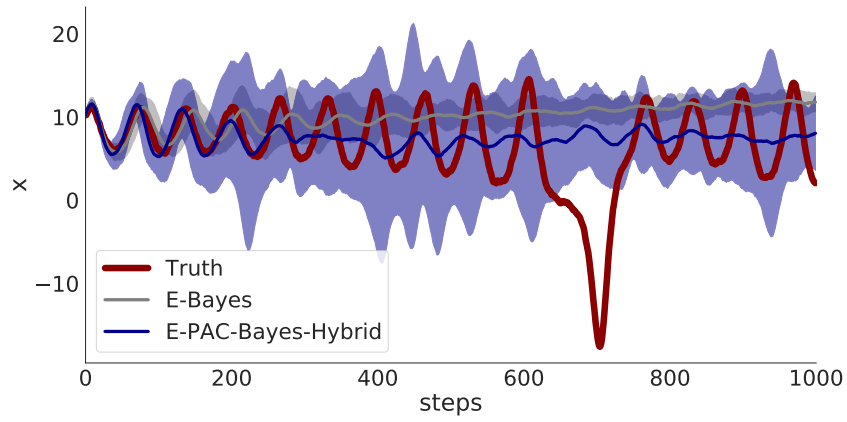


(b) y coordinate over time

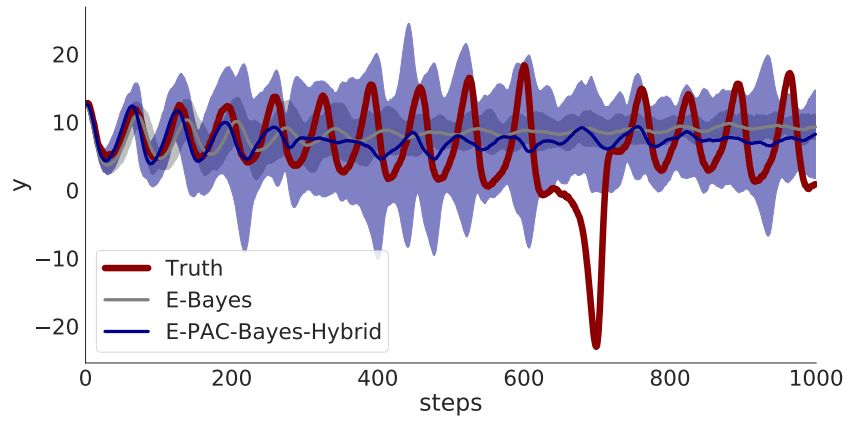


(c) z coordinate over time

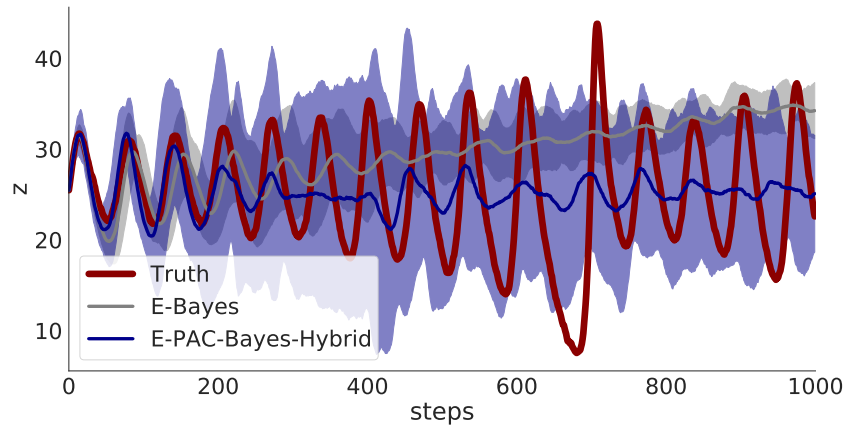
Figure 6: Predicting 200 time steps ahead.



(a) x coordinate over time

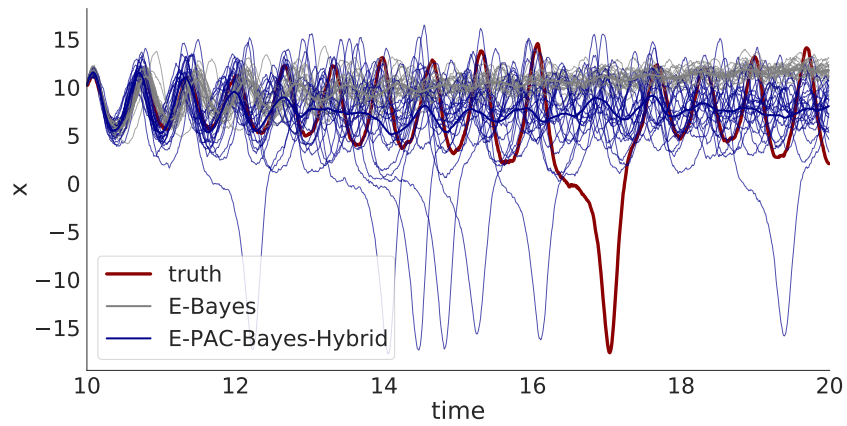


(b) y coordinate over time

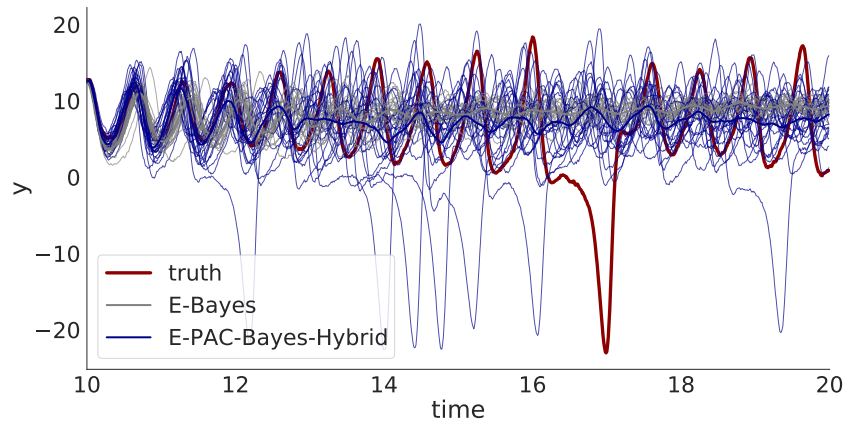


(c) z coordinate over time

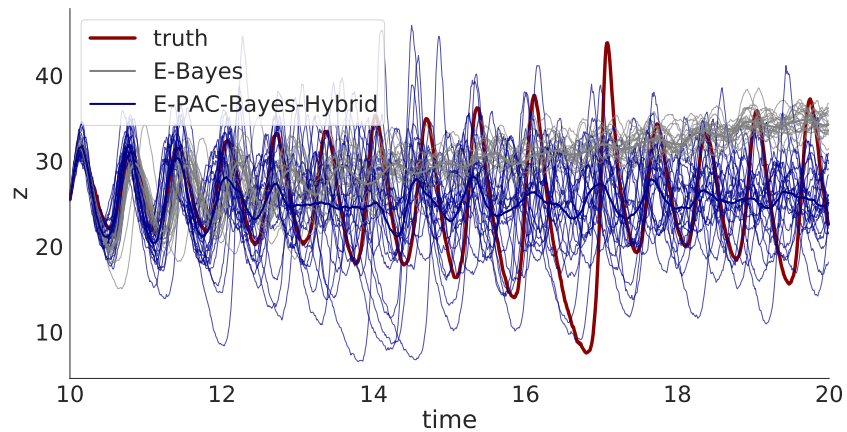
Figure 7: Predicting 1000 time steps ahead.



(a) x coordinate over time



(b) y coordinate over time



(c) z coordinate over time

Figure 8: Predicting 1000 time steps ahead. Shows individual trajectories.