Epistemic Uncertainty in Learning Chaotic Dynamical Systems

Nate Gruver [1] **Sanyam Kapoor** [1] **Miles Cranmer** [2] **Andrew Gordon Wilson** [1]

Abstract

Modeling conserved quantities like the *Hamiltonian* of a physical system provides strong inductive biases for efficient learning of the underlying dynamics. Chaotic systems, however, are sensitive to tiny perturbations in the initial conditions, such that point predictions can drastically diverge and lead to catastrophic outcomes. In this work, we take a first step towards quantifying uncertainty in the learned dynamics of such chaotic systems; we propose *CHNN-DE* which employs *Deep Ensembles*, and *CHNN-SWAG* which employs *Stochastic Weight Averaging Gaussian* to quantify uncertainty. With experiments on 3-body pendulum systems, we show that CHNN-DE and CHNN-SWAG are effective at providing long-horizon predictions in chaotic systems with well-calibrated uncertainty estimates.

1. Introduction

In engineering, accurate models are vital to keep a system in a desired state. *Predictive control* uses a known model to find a sequence of inputs that lead to these optimal states. When there is no known or accurate model of a system, we must perform *model learning* to identify how a system changes as a function of state and input.

Chaotic systems are particularly challenging to learn as small changes to the system's initial state, Δz_0, create large variations in the state at time t, Δz_t. Specifically, the error in a chaotic system grows exponentially as

$$|\Delta z_t| = e^{\lambda t} |\Delta z_0| ,$$

where λ is called the Lyapunov exponent. Chaotic dynamics are ubiquitous in nature and engineering systems, making accurate models and robust control of them imperative. Sensitivity to perturbations, however, makes modeling such systems difficult, as small errors can quickly cascade into large ones.

Naively applying deep learning techniques often leads to predictive performance that degrades quickly with increasing time horizon. Recent work combats compounding errors by directly modeling conserved quantities of the system (e.g. the Hamiltonian) (Greydanus et al., 2019; Cranmer et al., 2020), providing desirable inductive biases for learning. Finzi et al. (2020) extends this framework, introducing *Constrained Hamiltonian Neural Networks* (CHNNs) which use explicit physical constraints to further simplify the learning problem. Through their strong modeling assumptions, CHNNs are able to achieve to low error over long time horizons while still giving the flexibility of learning-based approaches.

From a control perspective, however, CHNNs and similar physics-inspired models have a noticeable shortcoming: they only provide point predictions. In planning and control, quantifying uncertainty is crucial for preventing catastrophic outcomes (e.g. hitting a pedestrian with an autonomous vehicle). When the system is chaotic, quantifying uncertainties has still greater importance, as compounding errors quickly create a gap between ground truth and predictions, which can go unnoticed without a metric of confidence.

We propose two approaches as extensions to CHNN: *CHNN-SWAG*, which applies *Stochastic Weight Averaging Gaussian* (SWAG) (Maddox et al., 2019) to compute an approximate posterior distribution over the model parameters, and *CHNN-DE*, which employs *Deep Ensembles* (Lakshminarayanan et al., 2017; Fort et al., 2019) to approximate the Bayesian model average. Through experiments on a 3-body pendulum (a chaotic system), we show that both CHNN-DE and CHNN-SWAG maintain high accuracy in long-horizon predictions and provide well-calibrated estimates of uncertainty.

2. Background

2.1. Learning dynamical systems

Differential equations and neural networks Ordinary differential equations (ODEs) allow modeling a system's state, z_t, by its rate of change, $\frac{dz}{dt}$. Given a specification $\frac{dz}{dt} = f(z, t)$, and a set of initial conditions z_0, z_t can

[1]New York University [2]Princeton University. Correspondence to: Nate Gruver <nvg7279@nyu.edu>, Sanyam Kapoor <sanyam@nyu.edu>.

Preliminary work. Under review by the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning. Do not distribute.

be found via integration, often approximately through numerical methods like Runge–Kutta (Runge, 1895; Kutta, 1901). Since the true $f(z,t)$ can be highly non-linear, recent work has explored using deep neural networks to parameterize f as f_θ. Predictions are calculated as $\hat{z}_t = \text{ODESolve}(z_0, f_\theta, t)$, yielding $\mathcal{L}(z, \hat{z})$ for loss function $\mathcal{L}$. An optimal setting of parameters θ is found via gradient descent using the adjoint method to back-propagate through the ODE solver (Chen et al., 2018). Thus, complex time and state-dependent dynamics can be learned directly from observations of the system, z_t.

Hamiltonian Neural Networks Physical systems can be expressed in terms of a more general operator that describes the total energy, known as the *Hamiltonian* $\mathcal{H}(z)$. The time evolution of such a system is described by (1).

$$\frac{dz}{dt} = J\nabla\mathcal{H}(z), \text{ where } J = \begin{matrix} 0 & I_{D/2} \\ -I_{D/2} & 0 \end{matrix} \quad (1)$$

The state can be decomposed as $z = (q, p)$ with generalized coordinates $q \in \mathbb{R}^{D/2}$ and generalized momenta $p \in \mathbb{R}^{D/2}$, which implicitly encode all system constraints. Importantly, systems governed by a Hamiltonian conserve total energy. Greydanus et al. (2019) proposed *Hamiltonian Neural Networks* (HNNs), which learn a forward model of the system by directly parametrizing $\mathcal{H}$ with a neural network. The resulting models, therefore, conserve energy by design.

Explicitly Constrained HNNs Finzi et al. (2020) extend the HNN framework to explicitly account for constraints, calling their new method *Constrained Hamiltonian Neural Networks* (CHNNs). In many physical systems, constraints have a significant effect on the dynamics. For example, an N-body pendulum consists of rods of length l_k, which constrain the movement of its attached masses m_i (Figure 1). CHNNs incorporate constraints present in the Cartesian coordinate system into the Hamiltonian of the generalized coordinates $\mathcal{H}(z)$. The increased structure provides strong inductive biases for more efficient model learning.

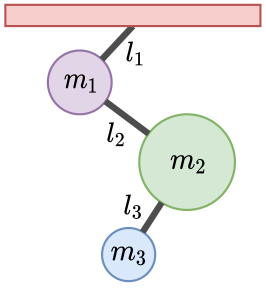


Figure 1. Example pendulum with 3 masses, m_i, and rods of length l_i.

In general, we can incorporate constraints into the learning of HNNs by using Lagrange multipliers, λ. For C holonomic constraints (relations between position variables) $\{\Phi(x)_j = 0\}_{j=1}^{C}$, we can derive $\Psi(z) = (\Phi, \frac{d\Phi}{dt})$, and the ODE

$$\frac{dz}{dt} = J \ \nabla\mathcal{H}(z) + (D\Psi(z))^\top\lambda \quad (2)$$

where $D\Psi$ is the Jacobian of Ψ wrt z. We can further solve for λ in terms of only J, $\nabla\mathcal{H}(z)$, and $D\Psi(z)$. See Finzi et al. (2020, Appendix F.2) for a full derivation of the constrained dynamics for n-pendulum systems.

To learn the dynamics from a set of trajectories, $\{\tau^i\}_{i=1}^N$ with $\tau = \{(z_0, t_0), ..., (z_T, t_T)\}$, we parameterize $\mathcal{H}$ as $\mathcal{H}_\theta$ with a neural network, and integrate through the dynamics in (2):

$$\begin{aligned} f_\theta(z,t) &= J \ \nabla\mathcal{H}_\theta(z) + (D\Psi(z))^\top\lambda \\ (\hat{z}_1^i, ..., \hat{z}_T^i) &= \text{ODESolve}(z_0^i, f_\theta, \{t_0, ..., t_T\}) \end{aligned} \quad (3)$$

We then minimize the L_1 error between the ground truth trajectory and the simulated trajectory.

3. Uncertainty in CHNNs

While HNNs and CHNNs are significant developments in learning system dynamics, they only provide point estimates. Chaotic systems rapidly grow unpredictable, and thus accounting for uncertainty is crucial in any practical modeling task. To quantify uncertainty in such models, we apply approximate Bayesian inference to marginalize over the neural network parameters, θ.

CHNN-DE Following Lakshminarayanan et al. (2017), we learn an ensemble $\{\theta_1, \theta_2, \ldots, \theta_E\}$. To sample from the predictive distribution we compute the trajectory for each ensemble member θ_e, $e \in \{1, 2, \ldots, E\}$,

$$(\hat{z}_1^{(e)}, \ldots, \hat{z}_\tau^{(e)}) = \text{ODESolve}(z_0, f_{\theta_e}, \{t_0, \ldots, t_\tau\}) \quad (4)$$

We can then approximate the mean and variance of the predictive distribution, $P(\hat{z}_t)$, as the empirical mean and variance of the sampled states, mean$(\hat{z}_t^{(1)}, ..., \hat{z}_t^{(E)})$ and var$(\hat{z}_t^{(1)}, ..., \hat{z}_t^{(E)})$. We call this method *CHNN-DE*.

CHNN-SWAG Applying SWAG (Maddox et al., 2019) to CHNN, we first learn a posterior over the model parameters $q(\theta)$ by performing SGD around a MAP estimate of the parameters. We can then construct the predictive distribution by sampling K parameters $\theta_k \sim q(\theta)$, $k \in \{1, \ldots, K\}$ and rolling out respective trajectories,

$$(\hat{z}_1^{(k)}, \ldots, \hat{z}_\tau^{(k)}) = \text{ODESolve}(z_0, f_{\theta_k}, \{t_0, \ldots, t_\tau\}) \quad (5)$$

We can then approximate the mean and variance of the predictive distribution as in CHNN-DE. We call this method *CHNN-SWAG*.

CHNN-OU Finally, we compare our methods which use epistemic uncertainty with a simpler approach which only attempts to model aleatoric uncertainty. Instead of using

(a) CHNN-OU (b) CHNN-DE (c) CHNN-SWAG

Figure 2. For a trajectory starting at one initial condition (dashed black line) in a 3-body pendulum, we plot the chaos (shaded gray) in the ground-truth dynamics, using ten uniform $U[-0.01, 0.01]$ perturbations around the initial condition, corrected for system's rigid-body constraints (length of pendulum's rod), alongside its mean trajectory (solid gray). For every trace plot, we additionally show the learned mean trajectory (solid color) and corresponding two standard deviations (dashed color). We note that CHNN-OU does not exhibit any perceptible uncertainty.

multiple settings of the parameters, we simply have the neural network f output both the Hamiltonian $H(z)$ and a covariance matrix $\Sigma(z)$, with

$$\Sigma(z) = \text{diag}(\sigma^{(0)}(z), \ldots, \sigma^{(Z)}(z)) \quad (6)$$

where Z is the dimension of the state z. We can then use the Hamiltonian H to derive each $\dot{z}_t$ as in (3). Taking $\mu(z_0, t) = \hat{z}_t$, we can put $P(z_t | z_0; t) = \mathcal{N}(\mu(z_0, t), \Sigma(z))$. We call this baseline method with output uncertainty CHNN-OU.

4. Experiments

In this section, we study the characteristics exhibited by our uncertainty quantification methods. We use a 3-body pendulum (Figure 1) to demonstrate the characteristics of uncertainty quantification in both CHNN-DE and CHNN-SWAG. The training set consists of segments with $T = 25$ and $t_T - t_0 = 0.75$. These 0.75 second segments are sampled at random from 800 complete trajectories of length 10 seconds each. We note that 10 seconds is considerably longer timescale than the 2 second window used by (Finzi et al., 2020), and predictions over longer windows become increasingly infeasible in chaotic systems. Integration to create the training data and fit the models is performed with RK4, a 4^{th}-order Runge-Kutta method (Runge, 1895; Kutta, 1901), using a time discretization of $\Delta t = 0.03$.

For CHNN-DE, we independently train $E = 10$ ensemble members, and for CHNN-SWAG we use $K = 10$ samples from the SWAG posterior. For both models we use 3-layer neural network with 256 hidden units and tanh nonlinearities. All neural network parameters are optimized with Adam using an initial learning rate of 0.003 and a cosine schedule for a total of 50 epochs. For CHNN-SWAG, we use an additional 20 epochs (i.e. $E_s = 20$) to collect the iterates used to estimate the statistics for the Gaussian posterior. All experiments have been made public[1].

4.1. Visualizing predictive uncertainty

For a qualitative sense of how useful our new predictive uncertainties are, we can compare the predictive distribution generated by the learned models to the ground truth. In order to visualize the effect of chaotic dynamics within the ground truth system, we create 10 perturbations around each initial condition and roll out a trajectory for each perturbed initial condition using the ground truth dynamics. The spread of these trajectories captures the underlying effect of chaos and allows us to see how much of our error might be due to an inherently difficult modeling problem.

Calibration Figure 2 displays distributions from CHNN-OU, CHNN-DE, and CHNN-SWAG alongside visualizations of how sensitive the system is to perturbation. When the system is more predictable, our models match ground truth with low predictive uncertainty over a long time horizon. As the system becomes more sensitive, each model's predictions drift away from the ground truth. CHNN cannot directly capture this discrepancy, while predictive uncertainty in our proposed Bayesian extensions becomes high as the system enters a chaotic region.

As the ground truth trajectory is largely contained within the high density region of the distribution, our predictors provide well-calibrated uncertainty estimates. Good calibration implies downstream consumption of this uncertainty (e.g. predictive control) will be well grounded, as we

[1] The experiments are accessible at snym/phy-unc-exps (Biewald, 2020).

Figure 3. We plot the mean root squared error over time for the trajectories generated by CHNN (black), CHNN-DE (green) and CHNN-SWAG (red), averaged over five different initial conditions, where the mean is denoted by solid lines and corresponding two standard deviations by dotted lines.

are not significantly over-confident or under-confident. Evidently, the Bayesian approach of CHNN-DE and CHNN-SWAG provides consequential advantages over simple maximum likelihood approaches, as CHNN-OU is consistently overconfident in its estimates.

4.2. Predictive advantages

As noted earlier, one fundamental appeal of CHNNs is their ability to model with high fidelity over long time horizons. Figure 3 compares the growth of error in a deterministic CHNN model with CHNN-DE and CHNN-SWAG. We see that error grows more slowly in CHNN-DE and CHNN-SWAG than in a vanilla CHNN model. The advantage of the Bayesian approach here is therefore not just the availability of useful predictive uncertainty but also improved predicted performance overall. As with many Bayesian approaches, the gains observed here are due to averaging over multiple settings of the parameters, causing errors in individual models to cancel each other out.

Low data Bayesian methods are especially relevant in applications with limited training data, as it becomes increasingly difficult for a single MAP estimate to sufficiently capture the data-generating process. One natural experiment, therefore, is to examine how our application of Bayesian methods to CHNN models affects performance under varying the amount of training data.

We evaluate performance by calculating the geometric mean of each prediction's relative error. Here the scale-independent relative error is taken to be

$$\epsilon(i;t) = \frac{\|\hat{z}_t^i - z_t^i\|_2}{\|\hat{z}_t^i\|_2 + \|z_t^i\|_2} \tag{7}$$

which allows us to quantify the deviation between a predicted value $\hat{z}_t$ and a ground truth value z_t. The relative error is close to one either when $\hat{z}_t \gg z_t$, or $\hat{z}_t$ is orthogonal to z_t. The geometric mean is calculated as

$$\mathrm{GM}(i) = \exp\left[\frac{1}{t_T - t_0}\int_{t=t_0}^{t_T} \log \epsilon(i;t)\, dt\right] \tag{8}$$

The geometric mean is used to aggregate error over time because log errors grow more manageably. We evaluate the error for the predictive mean of the CHNN-DE and CHNN-SWAG models and average over initial conditions z_0^i. Figure 4 shows a comparison of the geometric mean of relative error of CHNN against CHNN-DE and CHNN-SWAG over training datasets of varying size. Following Finzi et al. (2020), we compute this relative error over a shorter timescale of the first 3 seconds. We find that the Bayesian model averaging is able to provide improvements to the error over point estimates from CHNN, an already strong baselines, in low data settings.

Figure 4. We plot the geometric mean of relative error, under varying number of training data, evaluated over 25 trajectories (95% confidence interval). Lower is better. CHNN-DE and CHNN-SWAG show marginal improvements via the Bayesian model average in low data, but the predictive advantage quickly vanishes with increasing data, as expected.

5. Summary

In this work, we present a simple extension to CHNNs to quantify epistemic uncertainty via CHNN-DE and CHNN-SWAG. Both methods perform at least as well as CHNNs providing strong long-horizon predictions, while additionally quantifying the epistemic uncertainty in trajectories.

Uncertainty quantification can prove crucial for downstream applications like model-based control, which would otherwise not account for potentially catastrophic trajectories.

References

Biewald, Lukas. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

Chen, Ricky TQ, Rubanova, Yulia, Bettencourt, Jesse, and Duvenaud, David K. Neural ordinary differential equations. In Advances in neural information processing systems, pp. 6571–6583, 2018.

Cranmer, Miles, Greydanus, Sam, Hoyer, Stephan, Battaglia, Peter, Spergel, David, and Ho, Shirley. Lagrangian neural networks. arXiv preprint arXiv:2003.04630, 2020.

Finzi, Marc, Wang, Alex, and Wilson, Andrew Gordon. Simplifying hamiltonian and lagrangian neural networks via explicit constraints. NeurIPS, 2020.

Fort, Stanislav, Hu, Huiyi, and Lakshminarayanan, Balaji. Deep ensembles: A loss landscape perspective. arXiv preprint arXiv:1912.02757, 2019.

Greydanus, Samuel, Dzamba, Misko, and Yosinski, Jason. Hamiltonian neural networks. In Advances in Neural Information Processing Systems, pp. 15379–15389, 2019.

Kutta, Wilhelm. Beitrag zur naherungsweisen integration totaler differentialgleichungen. Z. Math. Phys., 46:435–453, 1901.

Lakshminarayanan, Balaji, Pritzel, Alexander, and Blundell, Charles. Simple and scalable predictive uncertainty estimation using deep ensembles. In Advances in neural information processing systems, pp. 6402–6413, 2017.

Maddox, Wesley J, Izmailov, Pavel, Garipov, Timur, Vetrov, Dmitry P, and Wilson, Andrew Gordon. A simple baseline for bayesian uncertainty in deep learning. In Advances in Neural Information Processing Systems, pp. 13153–13164, 2019.

Runge, Carl. Über die numerische auflösung von differentialgleichungen. Mathematische Annalen, 46(2):167–178, 1895.

A. Likelihood

We can also gauge the quality of our predictor's uncertainty by examining the likelihood our models assign to the ground truth data. Figure 5 shows the time evolution of the log-likelihood of a given ground truth trajectory, under the uncertainty of CHNN-DE and CHNN-SWAG. The log-likelihood is computed over the 12-dimensional state (i.e. two dimensional position and velocity for each of the three masses in a 3-body pendulum). High likelihood during early phase of the trajectory denotes the agreement with the ground truth with very high certainty. As the trajectory progresses, the likelihood naturally decreases. For the learned models, however, occasionally low likelihood values are a consequence of a disagreement with high certainty (as often in Figure 2). Eventually, for the learned models, the likelihoods converge to similar values as the trajectory enters chaos, indicating that the predicted uncertainty covers the spread of the chaotic trajectories well, i.e. the uncertainty is well-calibrated, as visually corroborated by Figure 2.

Figure 5. This plot summarizes the degree of agreement of the learned trajectories with the ground truth trajectory in Figure 2, through the full-state (position and velocity for all bodies) likelihood. Higher values indicate agreement with high con dence. As the trajectory enters chaos, we nd the learned models converge to values similar to as estimated by true chaos, indicating good calibration of the learned uncertainty estimates.

B. Estimating Chaos with Epistemic Uncertainty

We have emphasized the correlation between variance from underlying chaotic dynamics, and the variance of our predictive distribution via both qualitative and quantitative methods. The exact extent to which these two quantities are connected is worth its own discussion, which we provide here.

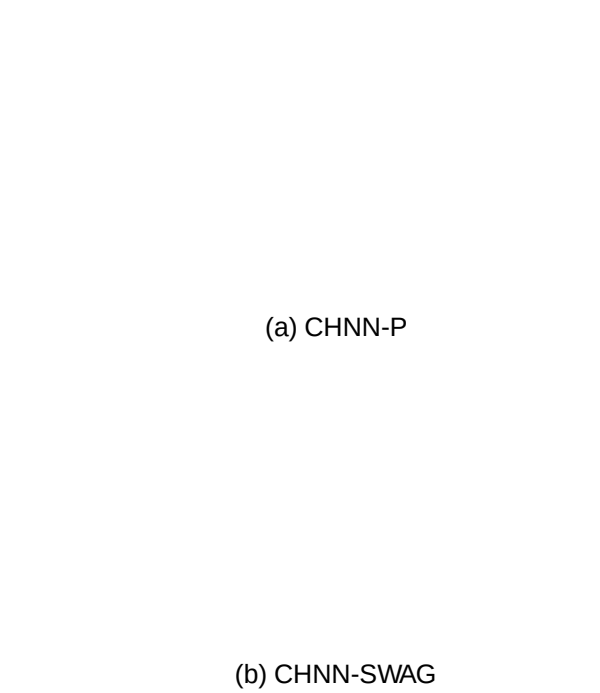
In general, if we want to quantify chaos, a simple approach is to perturb the initial conditions, and observe how quickly trajectories diverge. With a learned model, we can use the model to roll out perturbed initial conditions, and thereby get an estimate of chaos in the learned system. We refer to this method as *chaos by rollouts* (CR). We denote perturbations through a method using the suf x "-P", for instance CHNN-P denotes uncertainty estimated by perturbed initial conditions through a learned CHNN model.

In the limit of data, the magnitude of chaotic dynamics in the learned model should approach that of the underlying system. When we learn a model with well-calibrated uncertainty, however, the model's uncertainty estimates should also capture the magnitude of chaotic dynamics, as the model must spread density over all the outcomes that occur in practice. We can therefore identify chaos by an increase in the predictive variance of the model, and the samples from predictive distribution should diverge at a similar rate to trajectories with perturbed initial conditions. We call this method *chaos by uncertainty* (CU). Both CHNN-DE and CHNN-SWAG belong to this family of methods.

We compare these two approaches to quantifying chaos (CR and CU) in Figure 6. Both approaches evidently yield good estimates of the underlying chaos. In the case of CR, this result is not all that surprising, as we are able to learn a reasonably accurate model of dynamics that approaches the ground truth. The fact that CU re ects chaos to an almost equivalent extent, however, is quite signi cant, as using uncertainty estimates will often be much more practical than rolling out perturbations in practice. Although the computational demands are similar, design choices about the distribution of the perturbations will have signi cant consequences, whereas reasonable epistemic uncertainties are available without additional choices once the model is learned.

Rate of divergence It is notable that in Figure 2 and Figure 6, that the predictive uncertainty increases not only around the same time chaotic divergence emerges, but also at a similar rate. We further verify the similarity of rates by approximating the Lyapunov exponent in both cases. Figure 7 shows a linear t to log distance between trajectories. The slope of this line approximates the Lyapunov exponent. A detail of note here is the reduced time horizon for the ts presented in Figure 7, in comparison to Figure 2. The error in a chaotic system will eventually asymptote, such that it is no more predictable. For these plots, we are most interested in quantifying the transition into chaos, rather than chaos itself. Therefore, we pick a time-scale which is most faithfully depicts the region of transition into chaos.

From Figure 7, we nd that perturbed trajectories from



(a) CHNN-P

(b) CHNN-SWAG

Figure 6. Similar in spirit to Figure 2, we visualize the trace plots for one initial condition of a 3-body pendulum, using uniform $U[-0.01, 0.01]$ perturbations and the subsequent chaos by rollouts (CR) through a CHNN, denoted as CHNN-P. We skip CHNN-DE as the results are comparable to CHNN-SWAG. CHNN-SWAG mirrors the chaos through uncertainty quite well.

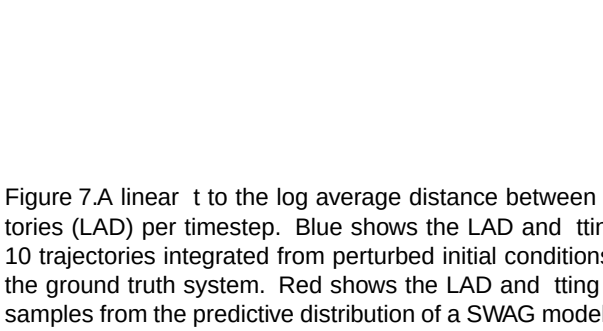


Figure 7. A linear fit to the log average distance between trajectories (LAD) per timestep. Blue shows the LAD and fitting for 10 trajectories integrated from perturbed initial conditions using the ground truth system. Red shows the LAD and fitting for 10 samples from the predictive distribution of a SWAG model.

the ground truth model and samples from a SWAG model yield nearly parallel lines, and therefore nearly identical exponents. Consequently, this implies that the rate of error growth is the same between methods in comparison. The offset, as compared to the true chaos, qualitatively indicates the timescale at which the learned method believes to have entered chaos. We present additional line fittings in Appendix F. While this method is by no means an exhaustive proof of the quality of learned models, it successfully demonstrates that chaos by epistemic uncertainty (CU) may be a viable approach.

To avoid clutter, we skip plotting results for CHNN-DE as they closely resemble the results for CHNN-SWAG.

C. Predictive advantages

Noisy data Alternatively, we might expect that Bayesian methods will be beneficial when data is corrupted with observation noise. Because it does not explicitly model uncertainty, CHNN might overfit to such noise, whereas our proposed variants are more capable of capturing inherent uncertainty as part of the prediction. To test this hypothesis, we added independent zero-mean Gaussian noise with standard deviation σ to the ground-truth states x_t for multiple values of σ. As above, we use the geometric mean of the relative error to evaluate performance across the modified training datasets.

Figure 8 shows a comparison of CHNN with CHNN-DE and CHNN-SWAG. We observe that CHNN-SWAG and CHNN-DE may often perform marginally better using the Bayesian model average under corrupted data. Noise corruption in the low data regime also leads to similar performance graphs, and are omitted for being redundant. Learning the dynamics becomes much harder with noise, as the inductive biases afforded by the Hamiltonian are violated. The advantage is potentially diminished by poor performance on certain trajectories. While the difference remains little for a 3-body pendulum, this is subject to further investigation on more complex chaotic systems.

C.1. Comparing CHNN-SWAG and CHNN-DE

We have focused primarily on comparing CHNN-DE and CHNN-SWAG against baseline methods (CHNN, CHNN-OU) without providing much comparison between the two methods. Looking at Figures 2, 5, 3, 4 and 8, we see that CHNN-DE and CHNN-SWAG exhibit very similar performance across the board. Because these models actually incur very different computational cost, this result is significant. In deep ensemble methods we must train E independent models, leading to computation that scales linearly with the size of the ensemble. In SWAG, by contrast, we only need to train one model and perform gradient steps around its optimum. If the predictive results are nearly identical for these two models, it is clear that SWAG will be favorable in most applications.

Figure 8. Under different levels of additive Gaussian noise corruption, we note the geometric mean of the relative error over 25 evaluation trajectories (95% confidence interval). Lower is better. CHNN-DE and CHNN-SWAG present marginal advantages via Bayesian model averaging. The Hamiltonian is violated, making the learning much harder even for small perturbations. We note that training diverges at higher noise levels.

D. Additional Trace Plots

We show additional trace plots for a 3-body pendulum in Figures 9 to 11. In each of these plots, we show the ground truth trajectory for a single initial condition, the chaotic behavior generated by tiny perturbations around the same initial condition, and the predictive uncertainty generated by trajectory rollouts from both CHNN-DE and CHNN-SWAG. We see that the predictive uncertainty is well-calibrated for both degrees of freedom (i.e. dimension "x" and "y") over a long-horizon of 10 seconds. We reiterate that the training happens only over randomly selected smaller chunks equivalent to a total horizon length of 0.75 seconds.

In Figures 9 to 11, we further notice that the predictive uncertainty aligns well with the region where chaos kicks in. We emphasize that we do not model chaos explicitly, and this behavior is learned entirely from the data alongside the inductive biases embodied in the Hamiltonian.

E. Relative Error

We evaluate the performance using relative error as defined in (7). Figures 12 and 13 show plots for more evaluation trajectories. We see that both CHNN-DE and CHNN-SWAG are similarly competitive.

F. Additional slope fits

Plots fitting a line to the log average distance (LAD) of trajectory for four other initial conditions are shown in Figure 14

(i) x

(ii) y

Figure 9. For the first mass in a 3-body pendulum, we visualize the true trajectory (dashed black line) alongside two standard deviations (gray region) of the trajectories generated by a symmetric uniform perturbation $U[-0.01, 0.01]$ around the true initial conditions. (Left) The uncertainty in trajectories generated by CHNN-DE is represented by the mean trajectory for 10 runs (solid blue line), with two standard deviations (dotted blue line). (Right) The uncertainty in trajectories is similarly plotted for CHNN-SWAG in red. Evidently, both methods provide well-calibrated long-horizon uncertainty estimates.

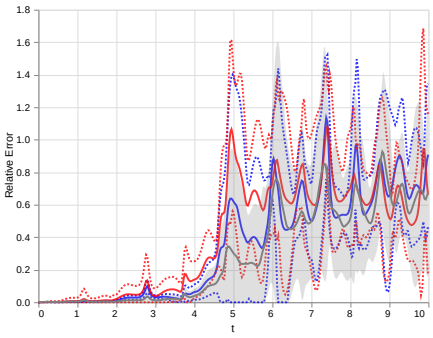
(i) x

(ii) y

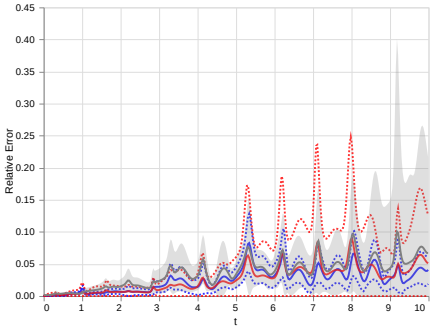
Figure 10. These figures are similar in spirit to Figure 9, but trace the uncertainty in trajectory for the second mass in a 3-body pendulum as generated by (Left) CHNN-DE, and (Right) CHNN-SWAG.

Figure 11. These figures are similar in spirit to Figures 9 and 10, but trace the uncertainty in trajectory for the third mass in a 3-body pendulum as generated by (Left) CHNN-DE, and (Right) CHNN-SWAG.

Figure 12. We compute the relative error (as discussed in Appendix E) for 10 independent trials. The solid gray curve represents the mean relative error of the perturbed initial conditions, with two standard deviations shaded. We make the same computations for both CHNN-DE (blue) and CHNN-SWAG (red), where the two standard deviations are bounded by respective dotted lines. We see that both methods perform similarly over longer timescales. The two plots (i) and (ii), correspond to different initial conditions.



(i)



(ii)

Figure 13. These figures show the growth of relative error for two different initial conditions (i) and (ii), similar in spirit to Figure 12.

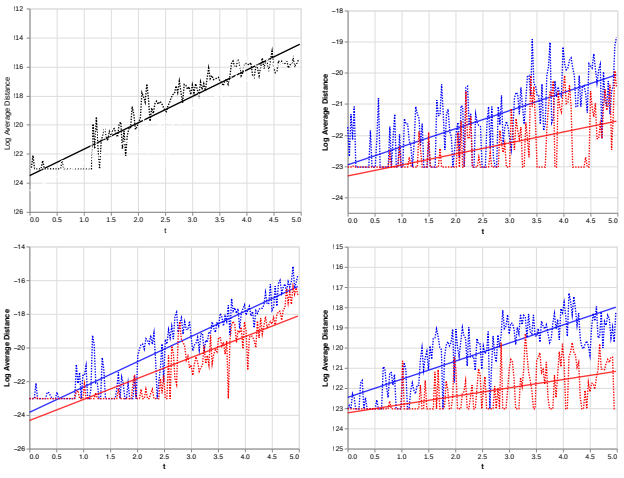


Figure 14. Linear fits of log average distance between trajectories vs. time. (blue) LAD and linear fit for trajectories from 10 perturbed initial conditions integrated forward using the ground truth model. (red) LAD and linear fit for 10 trajectories sampled from the predictive distribution of a SWAG model.