A Data Driven Approach for Resolving Time-dependent Differential Equations with Noise ⋆

Donglin Liu * **Alexandros Sopasakis** *

* *Department of Mathematics, Lund University, 22362 Lund, Sweden, (e-mail: donglin.liu@math.lth.se)*

Abstract: We propose data-driven surrogate models to solve systems of time-dependent differential equations coupled with noise. Using a feedforward neural network, we separately learn the noise and solution, tackling approximations across regimes with bifurcations and rare events. Focusing on irregular data generated by a stochastic noise model on a one-dimensional spatial lattice coupled to a differential equation, we examine two profiles: the periodic complex Ginzburg-Landau equation and a saddle bifurcation equation exhibiting rare events. This coupling introduces conditional data, enabling solutions to reach new states while posing challenges for accurately learning the underlying dynamics.

Keywords: Modeling and identification, Artificial intelligence and machine learning, Dynamic modelling and simulation for control and operation

1. INTRODUCTION

We explore classical differential equation solvers combined with neural networks in order to produce long-time solutions for time-dependent differential equations with noise. This approach draws inspiration from research in various physical applications, including catalysis Vlachos et al. (1990), polymeric flows, and stochastic parameterizations in tropical and open ocean convection Majda and Khouider (2002) . These dynamical systems have been modeled by the coupled equations we consider here. Furthermore, these equations have been used to better understand phenomena such as metastability, Katsoulakis et al. (2006) or rare events Katsoulakis et al. (2005b), as well as averaged behavior through mean field approximations Majda and Khouider (2002).

A number of recent works Kidger et al. (2021); Li et al. (2020); Ni et al. (2021); Yang et al. (2020) have studied the use of machine learning approaches to solve Stochastic Differential Equations (SDEs) or Partial Differential Equations (PDEs) with noise, showcasing promising results in resolving solutions for various equation types, complex geometries, and diverse initial/boundary conditions. Some of these approaches, like physics-informed generative adversarial networks (PI-GAN) Yang et al. (2020), combine data and mathematical techniques to reduce computational and training costs. These methods however lack robustness and have been shown to fail in certain parameter regimes Krishnapriyan et al. (2021) or for long-time dynamics Karumuri et al. (2020); Meng and Karniadakis (2020); Meng et al. (2020). Moreover, the complexity of the noise in the equations should be limited (i.e. not heteroscedastic Psaros et al. (2023)) to avoid introducing errors in the approximated solutions. Similarly, if the system's noise has a large correlation length, the neural network may struggle to learn the system dynamics Karumuri et al. (2020).

In this manuscript, we propose an approach that learns the distribution of the random variable representing the noise by combining a multi-Layer perceptron (MLP) with a simple first-order numerical integration formula, such as Euler's method. We call this method E-MLP and showcase its effectiveness for long-time solutions and irregularly sampled data.

2. A PROTOTYPE COUPLED SYSTEM

In many real-world applications, phenomena occur at multiple scales, where macroscopic behavior is influenced by underlying microscopic processes. We propose a data-driven approach to simulate a time-dependent coupled system with two pieces, without requiring prior knowledge of the system. The first piece is an ODE that serves as a caricature of overlying gas-phase dynamics. The second piece consists of a stochastic process $\{\sigma_t\}_{t\geq 0}$ defined on a spatial lattice $\mathcal{L}$. This process, as an example, can be thought of as modeling the adsorption and desorption of particles on a surface (see also Fig. 1). Both of these pieces are coupled and directly influence and change the overall system behavior. The coupled system is written as,

$$\begin{aligned} \tfrac{d}{dt}\boldsymbol{X} &= \tfrac{1}{\tau}F(\boldsymbol{X},\overline{\sigma}) \\ \tfrac{d}{dt}\mathbb{E}f(\boldsymbol{X},\sigma) &= \mathbb{E}Lf(\boldsymbol{X},\sigma), \end{aligned} \tag{1}$$

where $\mathbf{X}$ is the state vector, σ is the microscopic stochastic process defined on a spatial lattice $\mathcal{L}$ as in Fig. 1, $\bar{\sigma} = \frac{1}{N}\sum_{x\in\mathcal{L}}\sigma(x)$ is the spatial average of σ, τ is a characteristic time, $\mathbb{E}$ represents the expected value, F is

⋆ The work is partially supported by grants from eSSENCE no. 138227, FORMAS no. 2022-151862 and AgTech Sweden. The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

a function defining the ODE dynamics (examples of which are given below in (2) or (3)), f is a test function, and L is the generator of the stochastic process σ. For more details see Katsoulakis et al. (2006).

System (1) is based on noise originating from a jump stochastic process typically found in applications within micromagnetics Katsoulakis et al. (2005a), chemical reactors Vlachos et al. (1990), or climate models Majda and Khouider (2002). The noise in such a system, however, is not necessarily Gaussian. Therefore, methodologies rely on the assumption of Gaussian noise, such as the neural SDE method Li et al. (2020) , are unsuitable for application in this context.

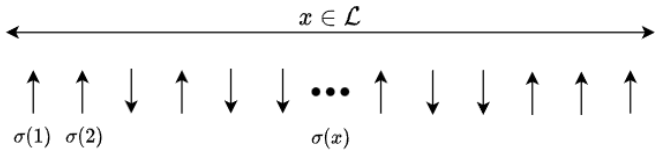

Fig. 1. Here $\sigma(x)$ denotes the value of the microscopic stochastic process σ at location x defined on an one dimensional spatial lattice $\mathcal{L}$. An arrow up(down) implies that a particle is present(absent) in that location. Note that $\bar{\sigma} = \frac{1}{N}\sum_{x\in\mathcal{L}} \sigma(x)$.

We consider and test our methods on two distinct ODE functions F in (1). The first one is an ODE exhibiting saddle-node bifurcation,

$$F(X, \sigma) = a(\bar{\sigma}) + \tilde{\gamma}X^2. \tag{2}$$

Bifurcation refers to the splitting or branching of a system into two or more distinct paths or directions.

The second ODE we consider is a spatially homogeneous complex Ginzburg-Landau (CGL) equation exhibiting Hopf bifurcations when excited by noise. In this case the noise is inserted in the ODE through $\bar{\sigma}$ - the spatial average of σ over the lattice $\mathcal{L}$,

$$F(\boldsymbol{X}, \sigma) = \left[\begin{pmatrix} a(\bar{\sigma}) + \gamma & -\omega \\ \omega & a(\bar{\sigma}) - \gamma \end{pmatrix} - \tilde{\gamma}|\boldsymbol{X}|^2\right]\boldsymbol{X}. \tag{3}$$

Here $a(\bar{\sigma}) = .5 - \bar{\sigma}$ where the value .5 is chosen to be in the middle of the range of $\bar{\sigma} \in [0, 1]$, $\gamma = .9$ and $\boldsymbol{X} = X + Yi$ is two-dimensional. Parameter details can be found in Appendix A. In (2) the system displays saddle behavior depending on the sign of $a(\bar{\sigma})/\tilde{\gamma}$. In (3) the Jacobian of the linearized system has eigenvalues $\lambda = \lambda(\bar{\sigma}) = a(\bar{\sigma}) \pm i\sqrt{\omega^2 - \gamma^2}$ and we have either a stable node at $(0, 0)$ for $a(\bar{\sigma}) < 0$ or a limit cycle. Each ODE example chosen above displays different behavior depending on how the coupling parameter $\bar{\sigma}$, which is stochastic, changes. The ODE at the top of (1) and the stochastic system at the bottom of (1) are coupled by a linear, external field $h(\boldsymbol{X}) = c\boldsymbol{X} + h_0$ and $\overline{\sigma} = \frac{1}{N}\sum_{x\in\mathcal{L}} \sigma(x)$ over a lattice $\mathcal{L}$ of N cells.

3. METHODS

Among the various approaches outlined in Section 1, the Neural SDE method has shown superior outcomes in capturing the dynamics of systems, however, have difficulty resolving noise for longer time dynamics for systems such as $\bar{\sigma}$ in the system (1) or those in other studies Psaros et al. (2023); Karumuri et al. (2020). Additionally, the Neural SDE method assumes that the noise in the data is Gaussian, which may not be suitable for general data or the many applications described in Section 1 which are modeled by the system (1).

3.1 E-MLP method with Gaussian noise

We assume that the ODE system (1) can be represented as a set of separable SDEs with unknown terms. Initially, we assume Gaussian noise in $\bar{\sigma}$, but this assumption is later relaxed. MLPs are employed to approximate both the system and the noise standard deviations, trained on datasets for $\boldsymbol{X}$ and $\bar{\sigma}$. The learned system will be numerically validated using the Euler–Maruyama method. We refer to this approach as E-MLP, and the model can be expressed as follows:

$$d\boldsymbol{X} = N_{\theta_1}(\boldsymbol{X}, \bar{\sigma})dt, \tag{4}$$

$$d\bar{\sigma} = N_{\theta_2}(\boldsymbol{X}, \bar{\sigma})dt + \eta(\boldsymbol{X}, \bar{\sigma})dW. \tag{5}$$

The neural network N_{θ_1} used to approximate $F(\mathbf{X}, \bar{\sigma})$ is a feedforward neural network with three hidden layers, each containing 50 neurons and using the hyperbolic tangent activation function. Similarly, N_{θ_2} is structured identically to approximate the dynamics of $\bar{\sigma}$. Here W is the Wiener process, and $\eta(\boldsymbol{X}, \bar{\sigma})$ denotes the unknown conditional standard deviation.

We begin by approximating network $N_\theta(\boldsymbol{X}, \bar{\sigma})$ with parameters θ. The ground truth for $N_{\theta_1}(\boldsymbol{X}, \bar{\sigma})$ is constructed using $\frac{\boldsymbol{X}_{i+1} - \boldsymbol{X}_i}{t_{i+1} - t_i}$ since it is supposed to approximate the derivative of $\boldsymbol{X}$. However, this idea is not as effective for $N_{\theta_2}(\cdot)$ due to the high noise in $\bar{\sigma}$. To reduce the noise, we apply a smoothing technique by increasing the step size k for the numerical derivative and use $\frac{\bar{\sigma}_{i+k} - \bar{\sigma}_i}{t_{i+k} - t_i}$ as the ground truth. This produces a coarse-grained version of the stochastic microscopic dynamics involved in the bottom part of system (1), which will be approximated here through $N_{\theta_2}(\boldsymbol{X}, \bar{\sigma})$. The loss function for either case is can be measured by,

$$L_n = \frac{1}{N}\sum_{i=0}^{N} ||N_\theta(\boldsymbol{X}_i, \bar{\sigma}_i) - y_i||^2, \tag{6}$$

where by y_i we denote either $\frac{\boldsymbol{X}_{i+1} - \boldsymbol{X}_i}{t_{i+1} - t_i}$ or $\frac{\bar{\sigma}_{i+k} - \bar{\sigma}_i}{t_{i+k} - t_i}$ depending on whether we are learning $\boldsymbol{X}$ or $\bar{\sigma}$ respectively. We also note that in the case that the mean-field model is available (i.e. assuming a large number of interacting particles we can effectively treat their contribution as a single entity, $u = \mathbb{E}\bar{\sigma}$ instead of the more noisy $\bar{\sigma}$ - see Appendix A for details), we can then add an extra contribution to the loss L_n,

$$L_m = \lambda\frac{1}{N}\sum_{i=0}^{N} ||N_{\theta_2}(\boldsymbol{X}_i, \bar{\sigma}_i) - f_m(\boldsymbol{X}_i, \bar{\sigma}_i)||^2,$$

where $f_m(\cdot)$ is the mean-field function of $\bar{\sigma}$ (see derivation (A.2) or (A.3) in the Appendix A) and λ is a weight hyper-parameter. Thus far, we have established how to approximate $\boldsymbol{X}$ and $\bar{\sigma}$. In order to proceed, we now also approximate the standard deviation $\eta(\cdot)$ for $\bar{\sigma}$. To do so we now discretize the $(\boldsymbol{X}, \bar{\sigma})$ space with a uniform grid and also make use of $N_{\theta_2}(\boldsymbol{X}_i, \bar{\sigma}_i)$ while rewriting (5) as,

$$\frac{\bar{\sigma}_{i+1} - \bar{\sigma}_i - N_{\theta_2}(\boldsymbol{X}_i, \bar{\sigma}_i)(t_{i+1} - t_i)}{\sqrt{t_{i+1} - t_i}} = \Lambda(\boldsymbol{X}_i, \bar{\sigma}_i). \tag{7}$$

Here, $\Lambda(\boldsymbol{X}_i, \bar{\sigma}_i) = \eta(\boldsymbol{X}_i, \bar{\sigma}_i)\epsilon$, where ϵ is standard Gaussian noise. We approximate $\eta(\boldsymbol{X}_i, \bar{\sigma}_i)$ by sampling from that grid. A practical implementation and sampling details for this grid can be seen in Section 4. Following similar ideas as above and using, once again, the mean square error loss function, we approximate $\eta(\boldsymbol{X}_i, \bar{\sigma}_i)$ with another neural network, $H_\phi(\boldsymbol{X}_i, \bar{\sigma}_i)$, that is parameterized by ϕ. The complete simulation therefore of the coupled system (1) can be computed by Euler approximation, following classical ideas from SDEs solvers,

$$\begin{aligned} \boldsymbol{X}_{i+1} &= \boldsymbol{X}_i + N_{\theta_1}(\boldsymbol{X}_i, \bar{\sigma}_i)dt, \\ \bar{\sigma}_{i+1} &= \bar{\sigma}_i + N_{\theta_2}(\boldsymbol{X}_i, \bar{\sigma}_i)dt + H_\phi(\boldsymbol{X}_i, \bar{\sigma}_i)\sqrt{dt}\epsilon. \end{aligned} \tag{8}$$

Thus, for given initial conditions $\boldsymbol{X}_0, \bar{\sigma}_0$, we can now estimate $\boldsymbol{X}$ and $\bar{\sigma}$ once we train networks $N_{\theta_1}(\cdot), N_{\theta_2}(\cdot)$ and $H_\phi(\cdot)$ from available data.

3.2 E-MLP method with empirical noise

One of the assumptions in the derivation of the method above is that $\bar{\sigma}$ in (5) has a Gaussian noise distribution. However, this assumption may not hold for all types of data in general, even though the L^2-norm loss function (6) assumes Gaussian noise. This discrepancy arises from the jump stochastic process, particularly in the Saddle system (see the "staircase" structure in Fig. 4), whereas in the CGL system, the jump process is less pronounced (see Fig.3). To relax the assumption of Gaussian noise, we introduced a neural network $K_\varphi(\mathbf{X}, \bar{\sigma}, \nu)$ that models the empirical cumulative distribution function (CDF) of the noise. By using inversion sampling, we can generate noise samples from the learned CDF during simulation, allowing us to capture non-Gaussian noise structures inherent in the data. Specifically, we propose to incorporate an MLP, $K_\varphi(\cdot)$, which indirectly estimates the distribution of the noise based on the provided data. The network learns the distribution of $\Lambda(\mathbf{X}_i, \bar{\sigma}_i)$ from Equation (7), parameterized by φ, and subsequently generates random variables from that distribution.

Specifically, we let d_j^i represent the empirical distribution from $\Lambda(\boldsymbol{X}_i, \bar{\sigma}_i)$ in (7), with $K_\varphi(\cdot)$ approximating d_j^i. We sample uniformly in $[0, 1]$ across n equidistant points ν_j for $j = 1, ..., n$, using inversion sampling of these points in d_j^i as the ground truth for $K_\varphi(\boldsymbol{X}_i, \bar{\sigma}_i, \nu_j)$. Hence, the conditional noise can be practically generated by inversion sampling from $K_\varphi(\boldsymbol{X}_i, \bar{\sigma}_i, \nu)$, where ν is a standard uniform random variable. Finally, following similar arguments, the proposed generalized MLP-based Euler approximation can be written as follows,

$$\begin{aligned} \boldsymbol{X}_{i+1} &= \boldsymbol{X}_i + N_{\theta_1}(\boldsymbol{X}_i, \bar{\sigma}_i)dt, \\ \bar{\sigma}_{i+1} &= \bar{\sigma}_i + N_{\theta_2}(\boldsymbol{X}_i, \bar{\sigma}_i)dt + K_\varphi(\boldsymbol{X}_i, \bar{\sigma}_i, \nu)dt^\alpha, \end{aligned} \tag{9}$$

where ν is a standard uniform random variable and α is a constant parameter that can be efficiently estimated through a multifractal detrended fluctuation analysis as is typically done to estimate the Hurst exponent in fractional Brownian motion Rydin Gorjão et al. (2022).

4. RESULTS

In this section, we present the results of our experiments using the proposed E-MLP method on examples exhibiting CGL (2) and Saddle (3) bifurcation profiles. To demonstrate the advantages of E-MLP, we compare its performance against several time series prediction models, including Vector Autoregression, LSTM, and Neural SDE. For the CGL example, we used two time series: one for training (91,665 points) and one for testing (91,732 points). For the Saddle example, we conducted experiments with both low and high noise datasets, using 20 time series for training and 80 time series for testing, each with approximately 90,000 points. We utilized long time series to provide the neural network with sufficient data to learn the distribution of noise, which varies depending on conditions involving $\boldsymbol{X}$ and $\bar{\sigma}$.

To ensure a fair comparison between the methods, we kept network settings similar across experiments. We used the *tanh* activation function and a batch size of 1024. The weights for the neural networks were initialized using Kaiming initialization He et al. (2015), and we used the Adam optimizer during training with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The network $N_\theta(\cdot)$ has 3 hidden layers of width 50, while both $H_\phi(\cdot)$ and $K_\varphi(\cdot)$ have 7 hidden layers of width 256. The Neural SDE and LSTM models have 8 layers of 256 neurons in hidden layers. We used vector autoregression (vector AR) with order $p = 600$ for both the CGL and Saddle cases, determined through grid search. We trained the networks $N_\theta(\cdot)$, $H_\phi(\cdot)$, and $K_\varphi(\cdot)$ with 1.5×10^4, 1.5×10^4, and 2×10^5 epochs, respectively, randomly selecting 1024 points from the dataset in each epoch. We used L_n as the loss function for both the CGL example (2) and the Saddle example (3) with low noise. For the Saddle case with high noise, we used $L_n + L_m$ as the loss function since we have an analytic description for the respective mean-field solution (A.2). The source code is available at `https://github.com/lindliu/Hybrid`.

First, given that $\boldsymbol{X}$ is noise-free, we approximated $\boldsymbol{X}$ using the specified $\bar{\sigma}$ to determine k. For the CGL (Saddle) case, we used a 3-dimensional (2-dimensional) input vector $[\boldsymbol{X}_i, \bar{\sigma}_i]$ to train the network $N_\theta(\cdot)$. This network approximates the numerical derivative vector space $[\frac{\boldsymbol{X}_{i+1}-\boldsymbol{X}_i}{t_{i+1}-t_i}, \frac{\bar{\sigma}_{i+k}-\bar{\sigma}_i}{t_{i+k}-t_i}]$. We divided our test data into 1000 randomly selected series, each containing 600 time points. The average MSE for $\boldsymbol{X}$ and $\boldsymbol{X}^{'}$ over these 1000 time series is presented in Table 1. The E-MLP method with a step size k of 10, 5, and 10 respectively for each of the three datasets tested achieved the best results.

Table 1. MSE error for $\boldsymbol{X}$ and $\boldsymbol{X}'$ for given $\bar{\sigma}$.

		E-MLP(k=5)	E-MLP(k=10)	E-MLP(k=20)
CGL	$\boldsymbol{X}$	6.03e-3	**1.67e-3**	4.24e-3
	$\boldsymbol{X}^{'}$	5.74e-4	**8.57e-5**	6.35e-4
Saddle (low noise)	X	**2.57e-7**	2.18e-6	2.05e-6
	$X^{'}$	**4.77e-6**	1.90e-5	1.77e-5
Saddle (high noise)	X	4.98e-5	**3.28e-6**	7.23e-6
	$X^{'}$	4.11e-4	**2.91e-4**	2.96e-4

Secondly, using the estimated k for each specific dataset, we trained two models, $H_\phi(\cdot)$ and $K_\varphi(\cdot)$, to learn the standard deviation of the Gaussian distribution and the empirical cumulative distribution function (ECDF), respectively. To approximate the ECDF, we used a mesh

grid for the $[\boldsymbol{X}, \bar{\sigma}]$ space with 30 columns and 20 rows, while ν consisted of 50 equidistant samples.

Figure 2 shows several simulated solutions produced by the E-MLP method compared to the data. The simulations for the Saddle case involve long-time predictions, with a time interval of 0.01, leading to as many as 10,000 time points.

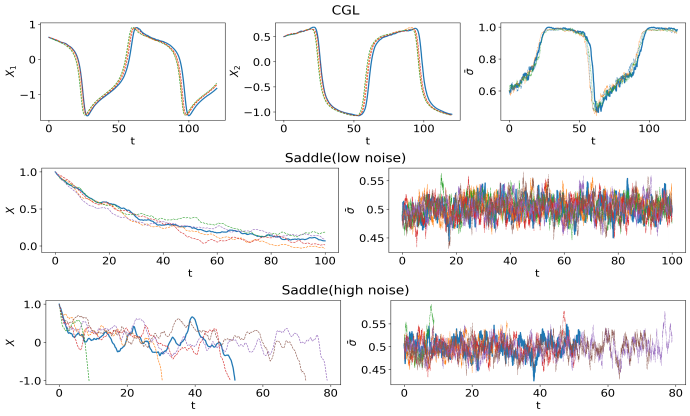

Fig. 2. **Synthetic data X, $\bar{\sigma}$ (blue line) versus several E-MLP simulations with empirical noise (other colors).** Top row: CGL case, simulations generated with $k = 10$ and $\alpha = 0.21$. Middle row: Saddle with low noise case, simulations generated with $k = 5$ and $\alpha = 0.427$. Bottom row: Saddle with high noise case, simulations generated with $k = 10$ and $\alpha = 0.425$. The E-MLP method can reproduce a rare event, identified numerically when $X < -1$.

5. DISCUSSION

To facilitate a comprehensive comparison across various baseline methods, we randomly selected 1000 time series from the test dataset, each consisting of 600 time points, to serve as the real target. For each method, we simulated 1000 instances using identical initial values. The distance between the simulated results and the target was calculated using the maximum mean discrepancy metric Gretton et al. (2012). The methods were also applied to an additional dynamical system, the Lorenz system from Li et al. (2020) (see Appendix B), as presented in Table 2.

Most of the baseline methods perform well for periodic-type systems such as CGL. However, for non-periodic systems, these methods accumulate errors quickly. This could be due to methods such as LSTM and vector autoregression (Vector AR) not being specifically designed to treat randomness, in contrast to the proposed E-MLP method. Similar results are observed for the latent Neural SDE method, which also fails to learn the distribution of the noise from such long time series. This is evident in the Lorenz system results shown in Table 2. Additionally, the E-MLP method had a faster average training time than the other methods tested.

From Table 2, we observe that the Lorenz and CGL systems perform better with E-MLP using Gaussian noise, whereas the Saddle system with high noise levels prefers E-MLP with empirical noise. To investigate the reasons behind this behavior, we examined the distribution of noise more closely. For the CGL bifurcation case (Fig. 3), the distribution of the noise in the data can be represented as part of a Gaussian distribution, which changes as $[\boldsymbol{X}, \bar{\sigma}]$ changes. In contrast, for the Saddle case, the distribution is not truly Gaussian (Fig. 4) and displays an underlying "staircase" structure. These "steps" originate from the dynamics of the stochastic jump process and are characteristic of data from classic models in micromagnetics and chemical processes. The E-MLP method with empirical noise was able to capture these challenging structures within the noise distribution.

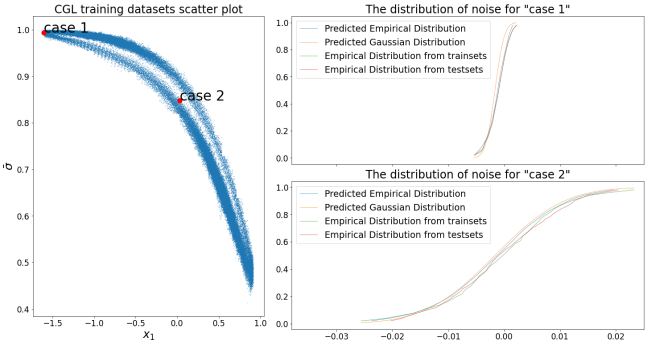

Fig. 3. **E-MLP method learning the distribution of noise for CGL system.** (Left) Scatter plot of all training data for the CGL case. (Right) Distributions of noise at two points ("case 1" and "case 2") based on the data from the left figure.

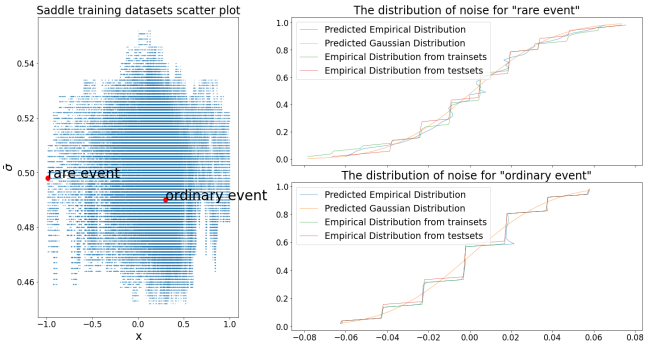

Fig. 4. **E-MLP method learning the distribution of noise for Saddle system.** (Left) Scatter plot of all training data for the Saddle case under high noise. (Right) Comparisons of the distributions of noise at a point near a "rare event" and a point near an "ordinary event" based on the left figure.

Figure 2 illustrates that the E-MLP method can learn the dynamics of both the CGL and the Saddle system, even in the case of rare events (i.e., finite-time blowup due to noise excitation). Notably, it is the only method among those tested that is able to learn rare events. To further demonstrate this phenomenon, we simulated 100 time series using (a) E-MLP with Gaussian noise (8) and (b) E-MLP with empirical noise (9). We found that it is sufficient for the model to see as few as 20 time series to learn and regenerate similar rare events. Specifically, the earth mover distance Rubner et al. (2000) of the rare event time between the ECDF and system (a) or (b) over 100 datasets is 6.24 and 2.86, respectively. This result reinforces the superiority of the E-MLP method with empirical noise for the high-noise Saddle system.

Table 2. The maximum mean discrepancy metric; mean ± standard deviation for 20 simulations. Here $ae-x$ implies $a \times 10^{-x}$.

	Training time(min)	Lorenz		CGL		Saddle(high noise)	
		$\boldsymbol{X}$	$\bar{\sigma}$	$\boldsymbol{X}$	$\bar{\sigma}$	X	$\bar{\sigma}$
Vector AR	0.3	1.0	1.0	3.1e-2	2.2e-2	1.0e-1	3.8e-2
LSTM	3.5	1.0e-1	6.4e-2	7.7e-1	6.3e-1	7.1e-1	1.1e-1
Neural SDE	35.2	2.5e-2±1.6e-3	2.0e-2±1.0e-3	9.6e-1±1.2e-3	9.4e-1±2.6e-3	8.6e-1±2.1e-3	9.2e-1±3.3e-3
E-MLP with $H_\phi(\cdot)$	1.5	1.4e-3±1.7e-5	**1.2e-3±3.8e-6**	**2.7e-2±4.9e-3**	**3.3e-2±5.0e-4**	**2.0e-2±5.6e-3**	2.2e-2±1.5e-3
E-MLP with $K_\varphi(\cdot)$	11.7	**1.4e-3±1.5e-5**	1.2e-3±5.5e-6	2.6e-2±5.3e-3	3.5e-2±8.6e-4	2.5e-2±2.1e-3	**7.7e-3±5.7e-4**

Our results indicate that the E-MLP method effectively captures the dynamics of both periodic and non-periodic systems, even in the presence of rare events. Compared to baseline methods, E-MLP demonstrates superior accuracy and computational efficiency. One notable strength is its ability to model non-Gaussian noise distributions, which is essential for systems where the noise cannot be assumed to be Gaussian. However, the method requires finely sampled data to accurately estimate derivatives and noise distributions. This could limit its applicability in scenarios where such data is unavailable. Future work will focus on extending the method to handle coarser data sampling and exploring its application to high-dimensional systems.

6. CONCLUSIONS

We examined machine learning methods to handle micro- or sub-grid scale noise in dynamical systems interacting over long-time domains. The proposed E-MLP method uses neural networks to independently learn both the differential equation solution, $\boldsymbol{X}$, and the noise, $\bar{\sigma}$, while utilizing the Euler–Maruyama solver to capture temporal dynamics. Essentially therefore E-MLP represents the ODEs from Section 2 as a system of separable SDEs with unknown terms which are parametrized by neural networks. These neural networks are in turn trained from the available data. All methods were tested on three differential equations. The E-MLP method was shown to outperform other comparable methods, such as LSTM, vector autoregression, or Neural SDE, in terms of accuracy and/or computational cost. One of the limitations of the proposed approach however is that it requires input data which must be sampled at a fine temporal scale in order to keep errors sufficiently small. As a result, the E-MLP approach would not be suitable for high-dimensional data due to the exponential increase in the required training data. In subsequent investigations, the E-MLP technique holds promise for further enhancement in generalization capabilities through the incorporation of a Gaussian noise model during the pre-training phase. Additionally, we intend to delve into the examination of the error propagated by the numerical approximation of the time derivative $\boldsymbol{X}'_{i+1} = \frac{\boldsymbol{X}_{i+1}-\boldsymbol{X}_i}{t_{i+1}-t_i}$, with a specific focus on its impact on the numerical stability of the method. We envision a comprehensive analysis encompassing both theoretical and numerical perspectives.

ACKNOWLEDGEMENTS

The authors declare that they have no conflict of interest. All the co-authors have confirmed to know the submission of the manuscript by the corresponding author.

REFERENCES

Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *J. Mach. Learn. Res.*, 13(1), 723–773.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *IEEE International Conference on Computer Vision (ICCV 2015)*, 1502.

Karumuri, S., Tripathy, R., Bilionis, I., and Panchal, J. (2020). Simulator-free solution of high-dimensional stochastic elliptic partial differential equations using deep neural networks. *Journal of Computational Physics*, 404, 109120.

Katsoulakis, M.A., Majda, A.J., and Sopasakis, A. (2004). Multiscale Couplings In Prototype Hybrid Deterministic/Stochastic Systems: Part I, Deterministic Closures. *Communications in Mathematical Sciences*, 2(2), 255 – 294.

Katsoulakis, M.A., Plecháč, P., and Sopasakis, A. (2005a). Error control and analysis in coarse-graining of stochastic lattice dynamics. Retrieved from the University of Minnesota Digital Conservancy.

Katsoulakis, M., Majda, A., and Sopasakis, A. (2005b). Multiscale Couplings in Prototype Hybrid Deterministic/Stochastic Systems: Part II, Stochastic Closures. *Communications in Mathematical Sciences*, 3(3), 453 – 478.

Katsoulakis, M., Majda, A., and Sopasakis, A. (2006). Intermittency, metastability and coarse graining for coupled deterministic-stochastic lattice systems. *Nonlinearity*, 19(5), 1021–1047.

Kidger, P., Foster, J., Li, X., and Lyons, T.J. (2021). Neural SDEs as infinite-dimensional gans. In M. Meila and T. Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 5453–5463. PMLR.

Kipnis, C. and Landim, C. (1999). *Scaling limits of interacting particle systems.* Springer.

Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., and Mahoney, M.W. (2021). Characterizing possible failure modes in physics-informed neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J.W. Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, 26548–26560. Curran Associates, Inc.

Li, X., Wong, T.K.L., Chen, R.T.Q., and Duvenaud, D. (2020). Scalable gradients for stochastic differential equations. In S. Chiappa and R. Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, 3870–3882.

PMLR.
Majda, A.J. and Khouider, B. (2002). Stochastic and mesoscopic models for tropical convection. *Proceedings of the National Academy of Sciences*, 99(3), 1123–1128.
Meng, X. and Karniadakis, G.E. (2020). A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems. *Journal of Computational Physics*, 401, 109020.
Meng, X., Li, Z., Zhang, D., and Karniadakis, G.E. (2020). PPINN: Parareal physics-informed neural network for time-dependent PDEs. *Computer Methods in Applied Mechanics and Engineering*, 370, 113250.
Ni, H., Szpruch, L., Sabate-Vidales, M., Xiao, B., Wiese, M., and Liao, S. (2021). Sig-wasserstein gans for time series generation. In *Proceedings of the Second ACM International Conference on AI in Finance*, ICAIF '21. Association for Computing Machinery, New York, NY, USA.
Psaros, A.F., Meng, X., Zou, Z., Guo, L., and Karniadakis, G.E. (2023). Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *Journal of Computational Physics*, 477, 111902.
Rubner, Y., Tomasi, C., and Guibas, L.J. (2000). The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40, 99–121.
Rydin Gorjão, L., Hassan, G., Kurths, J., and Witthaut, D. (2022). Mfdfa: Efficient multifractal detrended fluctuation analysis in python. *Computer Physics Communications*, 273, 108254.
Vlachos, D.G., Schmidt, L.D., and Aris, R. (1990). The effects of phase transitions, surface diffusion, and defects on surface catalyzed reactions: Fluctuations and oscillations. *The Journal of Chemical Physics*, 93(11), 8306–8313.
Yang, L., Zhang, D., and Karniadakis, G.E. (2020). Physics-informed generative adversarial networks for stochastic differential equations. *SIAM Journal on Scientific Computing*, 42(1), A292–A317.

Appendix A. THE MICROSCOPIC ARRHENIUS DYNAMICS AND RESPECTIVE CLOSURES

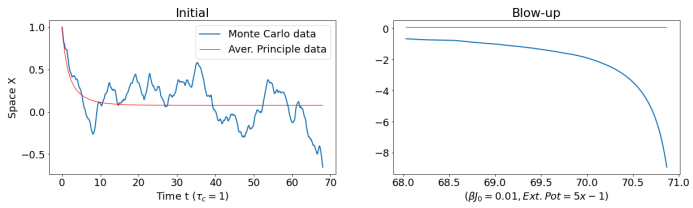

Fig. A.1. **Saddle ODE example and extended time runs.** Monte Carlo (blue) and averaging principle solution (red) for system (1). Parameters used in system (1): $b = 1, \tilde{\gamma} = -.05, \tau = 1$ and $\beta J_0 = .01$. Note the rare event jump which eventually drives the system to a finite time blow-up (right). Monte Carlo correctly captures this behavior of the solution while the stochastic averaging closure fails.

For completeness, we provide additional information on the stochastic dynamics behind the noise model in the prototype coupled system (1). Further details on this type of system are available in Katsoulakis et al. (2006).

We consider a microscopic stochastic model defined on a periodic lattice of size N which we denote by $\mathcal{L} = \{1, 2, \ldots, N\}$. At each lattice site $x \in \mathcal{L}$, an order parameter σ, is allowed to take the values 0 or 1. In accordance with the classical Ising model, we refer to the order parameter as spin. We assume that sites cannot be occupied by more than one particle. A spin configuration σ is an element of the configuration space $\Sigma = \{0,1\}^{\mathcal{L}}$ and we write $\sigma = \{\sigma(x) : x \in \mathcal{L}\}$ denoting by $\sigma(x)$ the spin at x. The stochastic process $\{\sigma_t\}_{t\geq 0}$ is a continuous time jump Markov process on $L^\infty(\Sigma, R)$ for any test function $f \in L^\infty(\Sigma, R)$ with generator, Kipnis and Landim (1999),

$$Lf(\sigma) = \sum_{x\in\mathcal{L}} c(x,\sigma)[f(\sigma^x) - f(\sigma)], \tag{A.1}$$

where,

$$\sigma^x(y) = \begin{cases} 1-\sigma(x), & \text{if } y = x \\ \sigma(y), & \text{if } y \neq x, \end{cases}$$

signifies the configuration after a flip at x. Note that $c(x,\sigma)$ denotes the rate of a spin flip at x for the configuration σ (see Vlachos et al. (1990); Katsoulakis et al. (2004)).

In order to better understand behavior of possible solutions it can be helpful to provide below closures Katsoulakis et al. (2006) of systems such as (1). We only present the final equations below and refer to Katsoulakis et al. (2004) for the details. The mean-field closure for system (1) is,

$$\begin{cases} \frac{d}{dt}\boldsymbol{Y} &= \frac{1}{\tau}G(\boldsymbol{Y}, \bar{u}) \\ \frac{d}{dt}u &= 1 - u - ue^{-\beta(J*u - h(\boldsymbol{Y}))}, \end{cases} \tag{A.2}$$

where $\bar{u}(t) = \int_0^1 u(y,t)\,dy$, for $t \in [0,T]$, $y \in [0,1]$ and a long-ranged potential J (a simple uniform potential can be used here). The respective stochastic averaging principle closure for system (1) is given by,

$$\frac{d}{dt}\bar{x} = \frac{b}{\tau}[z - u_{\beta,N}(h(\bar{x})] + \frac{\tilde{\gamma}}{\tau}\bar{x}^2 \text{ where } \bar{u}_{\beta,N} = \mathbb{E}\bar{\sigma}. \tag{A.3}$$

The solution of this stochastic averaging principle closure is shown in Fig. A.1 together with the exact solution from the Monte Carlo simulation of the coupled system (1) under the saddle ODE case (2). For short times, the values obtained by the stochastic averaging closure are in agreement with those from the Monte Carlo simulation, as can be seen in the left part of Fig. A.1. For a long enough time however a rare event jump occurs in the coupled system (1) which can not be followed by the stochastic averaging (A.3) (or the mean-field solution (A.2)). This rare event is responsible for the blow-up at finite times as Fig. A.1 also shows.

Appendix B. LORENZ SYSTEMS

We generate data from a Lorenz system,

$$\begin{aligned} dX_1 &= \gamma(X_2 - X_1)dt, \\ dX_2 &= (X_1(\rho - \bar{\sigma}) - X_2)dt, \\ d\bar{\sigma} &= (X_1X_2 - \beta\bar{\sigma})dt + \kappa dW, \end{aligned} \tag{B.1}$$

where $\gamma = 10, \rho = 28, \beta = 8/3$, and $\kappa = 1/2$. The initial point $(X_1(0), X_2(0), \bar{\sigma}(0))$ sampled from the standard Gaussian distribution as the ground-truth model. We generate 8,192 time series, sampled at 0.025 intervals from time 0 to 1, and add Gaussian noise with zero mean and a standard deviation of 0.01.