Neural Langevin Dynamics: Towards Interpretable Neural Stochastic Differential Equations

Simon Koop*[1], Mark Peletier[1], Jim Portegies[1], and Vlado Menkovski[1]

[1]Einddhoven University of Technology

Abstract

Neural Stochastic Differential Equations (NSDE) have been trained as both Variational Autoencoders, and as GANs. However, the resulting Stochastic Differential Equations can be hard to interpret or analyse due to the generic nature of the drift and diffusion fields. By restricting our NSDE to be of the form of Langevin dynamics and training it as a VAE, we obtain NSDEs that lend themselves to more elaborate analysis and to a wider range of visualisation techniques than a generic NSDE. More specifically, we obtain an energy landscape, the minima of which are in one-to-one correspondence with latent states underlying the used data. This not only allows us to detect states underlying the data dynamics in an unsupervised manner but also to infer the distribution of time spent in each state according to the learned SDE. In general, restricting an NSDE to Langevin dynamics enables the use of a large set of tools from computational molecular dynamics for the analysis of the obtained results.

1 Introduction

In recent years there has been widespread interest in Neural SDEs for generative modelling. Neural SDEs (NSDEs) learn the parameters in stochastic differential equations, which are natural extensions of ordinary differential equations to situations where uncertainty arises from many small and unobserved interactions, such as stock prices and molecular dynamics [1, 2]. As such, Neural SDEs can be seen as incorporating uncertainty in Neural Ordinary Differential Equations.

NSDEs are said to combine the strengths of deep learning and classical modelling [2, 3]. Deep learning brings performance and capacity, whereas classical modelling derives a lot of value from its interpretability. However, although from some perspective the dynamical system learned by an NSDE is itself a classical model, it is still a very complicated object, which makes it questionable whether NSDEs actually provide interpretability.

Indeed it is often overlooked in Neural Differential Equation literature that finding the differential equation is only the start of the process. For interpretation, one needs to analyse stability, find stationary solutions, and analyse asymptotic behaviour, to name a few. In the particular case of NSDEs, one would want to find basins of attraction — as they could correspond to discrete states in the dynamics — and would like to extract how long the system spends in each state and at what rates transitions occur. For general dynamical systems, these tasks are prohibitively complicated, which severely hampers interpretability.

To make NSDEs more interpretable, we introduce Neural Langevin Dynamics (NLD), in which we replace the generic vector field, which serves as the drift, by a gradient field, and let the system evolve by Langevin dynamics. This has the following advantages.

There is a direct interpretation of "states" in the original system as the basins of attraction around local minima of the energy [4]. These local minima can be found using the vast toolbox of scalar optimisation methods [5–8]. Sub-level sets of the energy function can then be used as approximations of the basins of attraction. This is illustrated by e.g. Figure 2(f), in which the contour lines clearly define regions that belong to specific states.

Moreover, there exist many more tools for visualising scalar functions than for visualising vector fields, aiding in the understanding by users of these models.

In this work, after introducing NLD in more detail, we show its potential and validity by training it as a Variational Autoencoder (VAE) with a learned prior[1] [1, 9] on an example problem. We show that the local minima indeed coincide with the true states underlying the data, and that information on the relative occurrence of states can be extracted from the learned energy landscape.

Our results indicate that NLD allows us to extract interpretable information about the data dynamics, and in such a way that it opens up the use of tools from optimization and analysis, effectively providing a bridge between deep learning and classical modelling.

*Corresponding Author: s.m.koop@tue.nl

[1]See Section 3 for further details.

Proceedings of the 5th Northern Lights Deep Learning Conference (NLDL), PMLR 233, 2024.

2 Related Work

The authors of [10, 11] view NSDEs as infinitely deep neural networks but don't use them as time series models. In [2], NSDEs are trained as a GANs instead of as VAEs, and the authors of [12, 13] train an NSDE in a GAN-related way. The authors of [1] introduce the training of an NSDE as a VAE on sequential data. We use their training method to train our NLD models. The authors of [3] also train NSDEs as VAEs but focus mostly on the dimensionality of the data space. To address problems around sample paths being far from observations during the start of training, the authors of [14] devise a method of training NSDEs using importance sampling which could prove useful in the future to make the training of NLD models more stable. Another work addressing the learning of NSDE in data-space is [15], where snapshots are used instead of trajectories.

The authors of [1, 16] extend the so-called adjoint sensitivity method to NSDEs and provide algorithms for storing Wiener processes. We do not use the adjoint sensitivity method but do use their algorithms for training NSDEs in the form of the torchsde package —https://github.com/google-research/torchsde.

NSDEs have found practical use, e.g. for modeling turbulence [17] and for modeling quasar variability [18].

NSDEs can be viewed in the wider context of neural differential equations. The seminal paper in this area was [19]. These were extended to models for sequential data in, among others [20–22].

Moreover, NLD can be viewed in a line of physics-inspired neural network models [23–26]. The focus in the cited works however was more on performance and theoretical generalisability whereas our work focuses more on interpretability.

Langevin dynamics themselves have been extensively researched since the start of the 20th century and remain a topic of active research, more on this topic can be found in e.g. [27–29]. Besides their relevance for physical modelling, they play an important role in MCMC sampling [4].

3 Method

3.1 Neural Stochastic Differential Equations

When training Neural Stochastic Differential Equations as Variational Auto-Encoders on some data $X = \left\{x^{(i)} = \{x_t^{(i)}\}_{t=0}^T \mid i = 1, \ldots, N\right\}$ with $N, T \in \mathbb{N}$ the number of data points and sequence length respectively, as in [1], one has two stochastic differential equations parameterising the prior $p_\theta(z \mid z_0)$ and approximate posterior $q_{\phi,\theta}(z \mid x, z_0)$ distributions:

$$
\begin{aligned}
p_\theta(z \mid z_0) &\sim \mathrm{d}z_t && (1)\\
&= h_\theta(z_t, t)\mathrm{d}t + g_\theta(z_t, t)\mathrm{d}W_t && (2)\\
&\text{(prior)}, && (3)\\
q_{\phi,\theta}(z \mid x, z_0) &\sim \mathrm{d}z_t && (4)\\
&= f_\phi(z_t, t, c_t)\mathrm{d}t + g_\theta(z_t, t)\mathrm{d}W_t && (5)\\
&\text{(approximate posterior)}, && (6)
\end{aligned}
$$

with both differential equations starting in z_0.

Here c_t is a function of the data x obtained by some encoder architecture, in this work a GRU [30]. Throughout this work, $c_t^{(i)}$ will only depend on $x_{s\ s\leq t}^{(i)}$ so the models all operate in an online fashion. The network being integrated against t, i.e. h_θ in the prior and f_ϕ in the approximate posterior, is called the **drift**, and the network being integrated against the Wiener Process W_t, i.e. g_θ, is called the **diffusion**.

As shown in [1], the Kullback-Leibler divergence between the approximate posterior and the prior is then given by

$$
\begin{aligned}
D_{KL}\left(q_{\phi,\theta}(z \mid x, z_0) \,||\, p(z \mid z_0)\right) && (7)\\
= \frac{1}{2}\int_0^T |u(z_t, t, c_t)|^2 \mathrm{d}t, && (8)\\
g_\theta(z_t, t)\, u(z_t, t, c_t) && (9)\\
= f_\phi(z_t, t, c_t) - h_\theta(z_t, t), && (10)
\end{aligned}
$$

where T is the final time.

Additionally, one has a decoding distribution $p_\theta(x_t \mid z_t)$, a prior distribution over the initial latent state $p_\theta(z_0)$, and an approximate prior distribution over the initial latent state $q_\phi(z_0 \mid x)$[2].

Then the evidence lower bound (ELBO) can be written as

$$
\begin{aligned}
&\log p_\theta(x_0^{(i)} \ldots x_T^{(i)}) \geq \mathbb{E}_{z^{(i)} \sim q_\phi(z \mid x^{(i)})}\Big[&& (11)\\
&\sum_{t=0}^{T} p_\theta(x_t^{(i)} \mid z_t^{(i)}) - \frac{1}{2}\int_0^T |u(z_t, t, c_t)|^2 \mathrm{d}t\Big] && (12)\\
&- D_{KL}\left(q_\phi(z_0 \mid x^{(i)}) \,||\, p_\theta(z_0)\right), && (13)
\end{aligned}
$$

where $q_{\phi,\theta}(z \mid x) = q_{\phi,\theta}(z \mid x, z_0) q_\phi(z_0 \mid x)$.

To find the optimal parameters θ, ϕ, we can approximate the ELBO for a batch of datapoints by sampling from $q_\phi(z_0 \mid x)$ using the reparameterisation trick [9], and sampling z using an SDE solver such as the Euler-Maruyama method. To obtain gradients for the stochastic integral term, we can either back-propagate through the SDE solver or use the adjoint method [1, 16]. We can then use stochastic gradient ascent to optimise the ELBO.

[2] Again, in this work z_0 only depends on x_0, so that the model can operate online.

Note that $p_\theta(z)$ is a learned distribution, parameterised by an NSDE, representing the latent dynamics of the dataset as a whole. It is learned by trying to statistically match the encoded data sampled from the approximate posterior $q_{\phi,\theta}(z \mid x)$ through the Kullback Leibler divergence. However, it is not easy to extract useful information about these latent dynamics from this representation of the distribution due to the generality of Equation (2).

3.2 Neural Langevin Dynamics

We introduce Neural Langevin Dynamics as a more interpretable alternative to NSDE. In NLD, we replace the drift h_θ in Equation (2) by the scaled gradient of an energy $E_\theta : \mathbb{R}^d \to \mathbb{R}$ that is given by a neural network[3], and modify the evolution to either overdamped or underdamped Langevin dynamics.

In the case of overdamped Langevin dynamics, the prior and approximate posterior evolutions are determined by

$$\mathrm{d}z_t = -\gamma^{-1}\nabla_z E_\theta(z_t)\mathrm{d}t + \sqrt{2\beta^{-1}\gamma^{-1}}\mathrm{d}W_t \quad (14)$$

$$\mathrm{d}z_t = -\gamma^{-1}\nabla_z E_\theta(z_t)\mathrm{d}t \quad (15)$$

$$+ f_\phi(z_t, t, c_t)\mathrm{d}t + \sqrt{2\beta^{-1}\gamma^{-1}}\mathrm{d}W_t \quad (16)$$

whereas in the case of underdamped Langevin dynamics, they are determined by

$$\begin{cases} \mathrm{d}z_t^q = & M^{-1}z_t^p\mathrm{d}t \\ \mathrm{d}z_t^p = & \left[-\nabla E_\theta(z_t^q) - \gamma M^{-1}z_t^p\right]\mathrm{d}t \\ & +\sqrt{2\gamma\beta^{-1}}\mathrm{d}W_t \end{cases} \quad (17)$$

$$\begin{cases} \mathrm{d}z_t^q = M^{-1}z_t^p\mathrm{d}t \\ \mathrm{d}z_t^p = -\left[\nabla E_\theta(z_t^q) + \gamma M^{-1}z_t^p\right]\mathrm{d}t \\ \qquad +f_\phi(z_t^q, z_t^p, t, c_t)\mathrm{d}t + \sqrt{2\gamma\beta^{-1}}\mathrm{d}W_t. \end{cases} \quad (18)$$

Here, M is a positive diagonal matrix representing mass. The constants M, γ, and β can either be fixed or can be learned while training the model.

In the underdamped case, z^q plays the role of location, and z^p plays the role of momentum. Only the location, z^q, should be used for decoding. Note that z^q follows a second-order stochastic differential equation.

One of the great advantages of using Langevin dynamics is that this way, the SDE of the prior distribution has a known stationary distribution of the form

$$p_\infty(z) \sim \frac{\exp(-\beta E_\theta(z))}{\int_{\mathbb{R}^d}\exp(-\beta E_\theta(y))\mathrm{d}y} \quad (19)$$

[3]To ensure the existence of a stationary measure, we parameterize E_θ by the sum of a neural network and a quadratic term: $E_\theta(z) = \mathrm{NN}_\theta(z) + c_\theta\|z\|^2$.

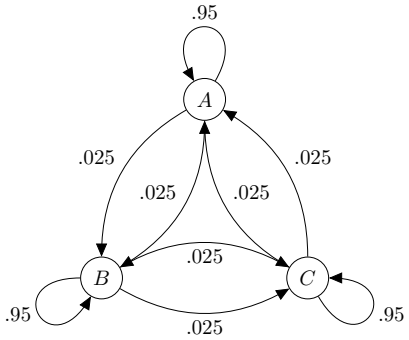


Figure 1. Graph of the Markov process underlying the data.

in the overdamped case, and

$$p_\infty(z_q, z_p) \sim \frac{\exp(-\beta E_\theta(z_q))}{\int_{\mathbb{R}^d}\exp(-\beta E_\theta(y))\mathrm{d}y} \otimes \mathcal{N}(z_p; 0, M^{-1}) \quad (20)$$

in the underdamped case. Due to ergodicity, the proportion of time spent in a region of the latent space, in the long run, corresponds to the mass attributed to that region by the stationary distribution, which only relies on the β and E_θ.

4 Experimental setup

We evaluate the capability of our method to develop an interpretable representation of data dynamics by training a model on data coming from a Markov chain. We then analyse the energy landscape of this model to recover the discrete states of the Markov chain and infer the time spent in each state.

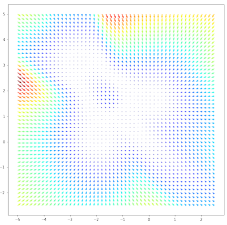
The data is generated as random walks on a graph with three nodes. At each time step, a signal is emitted as a random vector sampled from a 15-dimensional multivariate normal distribution with mean and covariance matrix specified by the node the random walk is at.

The transition probability from any state to any different state is 0.025 at each time step, resulting in a stationary distribution (1/3, 1/3, 1/3), and an expected transition time of 20 time-steps. An overview of this is shown in Figure 1.

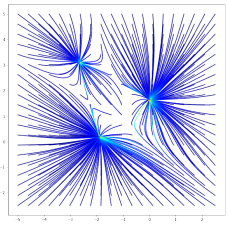
Code and data for the training of the models, as well as some trained models, are provided at https://github.com/SimonKoop/NLD-public.

5 Visualisation advantages of NLD

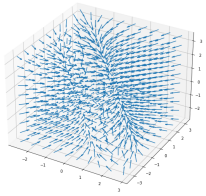
Visualisation is an important tool for obtaining human knowledge in general and can be useful for extracting knowledge from a trained model. To visualise trained models, if we only have a drift field,



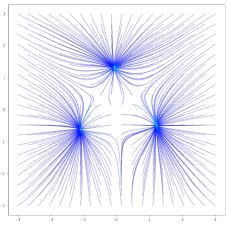
(a) Quiver plot drift field.



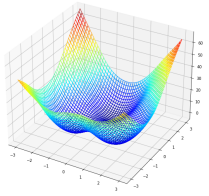
(b) Flow lines drift field.



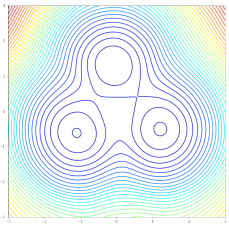
(c) Quiverplot of 3d drift field.



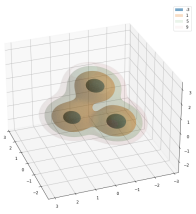
(d) Projection of flowlines starting in a 2d plane onto that same plane.



(e) A 3d plot of the energy landscape restricted to a plane through the three minima.



(f) A contour plot of the energy restricted to the plane through the three minima.



(g) A 3d contour plot of the energy landscape.

Figure 2. Visualisations the drift in the latent space of an NSDE (a-b) respectively an NLD (c-g) trained as a VAE.

as is the case for a general NSDE, all we can really do is make a quiver plot of the drift field or plot the flow lines of the drift. For the two-dimensional case, this is shown in Figure 2(a) and Figure 2(b).

As shown in Figure 2(c), a quiver plot of a three-dimensional vector field is already rather hard to interpret. One can restrict these visualisation techniques to a plane to make them more understandable, as is shown for a flow-line plot in Figure 2(d).

If on the other hand, the drift field of the model is a gradient field of an energy function, as is the case with the NLD models, we can additionally visualise the energy function, e.g. by plotting a graph or making a contour plot. If the dimensionality of the latent space is more than three, making it impossible to just plot the graph or the contours, we can restrict ourselves to a two-dimensional plane within the larger space, as is shown in Figure 2(e) and Figure 2(f), or to a three-dimensional subspace, resulting in a visualisation like the one in Figure 2(g).

These additional tools can aid a practitioner in interpreting what the model has learned and translating the obtained results to either existing or new knowledge about the data.

5.1 Visualisation procedure in our experiments

For the visualisation of the energy landscapes obtained in our experiments, we found the relevant subspace using the following procedure. First, we encoded part of the training data, say $x^{(1)}, \dots, x^{(k)}$ to the latent space using the approximate posterior $q_\phi(z \mid x)$, observing the latent code at pre-determined times $t_1, \dots, t_n$ to get $k \cdot n$ codes $z_{t_1}^{(1)}, \dots z_{t_n}^{(1)}, \dots, z_{t_1}^{(k)}, \dots z_{t_n}^{(k)}$. Then we performed gradient flow (or, really, gradient descent) starting in these codes:

$$\dot{z}_{i,j}(t) = -\nabla E_\theta(z_{i,j}(t)), \tag{21}$$

$$z_{i,j}(0) = z_{t_j}^{(i)}. \tag{22}$$

Finally, we perform PCA on the set of resulting vectors $\{z_{i,j}(T) \mid i = 1, \dots, k, j = 1, \dots, n\}$ for some large enough T. In our experiments, we parameterise E_θ as $E_\theta(z) = c|z|^2 + \mathrm{MLP}(z)$, where c is a learned parameter. As a rule of thumb for T, we take T such that for $\dot{\tilde{z}}(t) = -c \cdot \tilde{z}$ with $\tilde{z}(0) = 3\sqrt{1/c}$, $\tilde{z}(T) = 0.1$. We can view PCA as a projection onto an affine plane, and this plane is the one we use for visualising the energy landscape. Most of the $z_{i,j}(T)$ will be close to one of the three minima, so most of the variation is from having multiple energy wells. The plane obtained through this method will therefore be approximately through the three energy minima.

Finding the relevant subspace for visualisation becomes trickier when more energy minima are present,

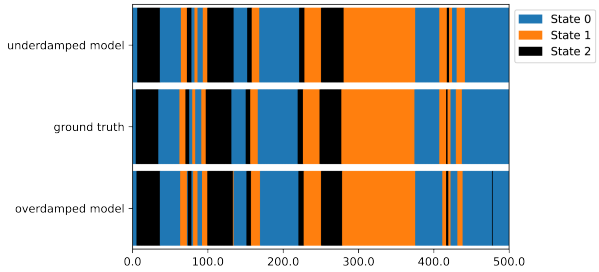


Figure 3. Results of unsupervised sequence segmentation of the first test sequence by both an underdamped and an overdamped NLD model.

or when the wells have more complicated shapes. Nonetheless, the first two steps above can be used to find the locations of the wells, and based on that, one can either choose multiple affine subspaces to visualise, or pick one (or more) non-linear surfaces through the minima on which to visualise the energy landscape. Alternatively, one could approximate minimum energy paths between the minima through string methods, and plot the energy along those paths.

6 Results

We evaluate whether the learned energy landscape accurately represents the distribution of states underlying the data.

The data-generating Markov chain has an invariant distribution of $(1/3, 1/3, 1/3)$, and we investigate whether this distribution can be recovered from the prior in the trained Neural Langevin Dynamics network.

We identify the three states in the Markov chain with the three wells in the energy landscape. To evaluate whether the energy landscape accurately represents the distribution of states underlying the data, we estimate the distribution $\underline{p} = (p_1, p_2, p_3)$ over the three wells according to the learned prior in three different ways and compare the results to the invariant distribution of the Markov chain. The three ways of estimating $\underline{p}$ are:

(a) by taking a thousand samples from the stationary distribution of the prior SDE, performing gradient flow on them, and seeing what percentage of samples ends up in each well;

(b) by taking the zeroth order approximation $p_i \approx \exp(-\beta E_\theta(x_i))/\sum_j \exp(-\beta E_\theta(x_j))$ where x_i is the location of the local minimum in well i;

(c) by taking the second-order approximation:

$$p_i \approx \alpha \frac{\exp(-\beta E_\theta(x_i))}{\rho_i}, \tag{23}$$

$$\alpha = \Big(\sum_j \frac{\exp(-\beta E_\theta(x_j))}{\rho_j} \Big)^{-1}, \tag{24}$$

$$\rho_i(x_i) = ((2\pi)^d \det(\beta^{-1} \mathrm{d}^2 E_\theta(x_i)^{-1}))^{-1/2}, \tag{25}$$

where $\mathrm{d}^2 E_\theta(x_i)$ is the Hessian of the energy function at x_i, and d is the dimension of the latent space. This second-order approximation corresponds to approximating the three wells by a mixture of three Gaussians.

We report the ℓ^1-distance between $\underline{p}$ and $(1/3, 1/3, 1/3)$ for each approximation for the top five best performing models ranked by second-order approximation performance for both the underdamped and overdamped models in Table 1(a). For a more easily interpretable representation of the results, we also report the first coordinate of $\underline{p}$ in Table 1(b).

A second question is whether the discovered "states", i.e. the energy wells, in the trained prior dynamics coincide with the ground-truth latent states in the data-generating Markov chain and whether the trained encoder network successfully maps unseen incoming data to the latent states investigated above. This is a measure of the degree to which the training has managed to model the whole dynamics. We generated 500 test sequences from the same three-state Markov chain and mapped them to the latent space using the approximate posteriors, $q_{\phi,\theta}(z \mid x)$, of each of the models used for Table 1(a). Then we followed the gradient flow from each time step in each sequence of latent-space points to find out which well they belong to; this leads to a sequence of labels $(0, 1, 2)$. To compare these labels to the ground truth, we selected the permutation of state labels $(0, 1, 2)$ for which the overlap is maximised. We then computed the mean and standard deviation of the accuracy over the 500 sequences. The results are shown in Table 1(c).

We provide an example of the segmentation in Figure 3. Note that the models operate in an online fashion resulting in the encoded latent state lagging slightly behind the true latent state. Nonetheless, these results clearly show that the wells in the energy landscape coincide with the discrete states behind the data. Moreover, the models were trained on sequences of length 200, but are tested on sequences of length 500, generalising to sequences 2.5 times longer than seen during training.

Table 1. Results on estimating states from the learned energy landscape for the overdamped model (O) and underdamped model (U).

(a) ℓ^1 distance between the estimated distribution of states and actual distribution of states.

	Sampled (method (a))	0^{th} order (method (b))	2^{nd} order (method (c))
O	0.047 ± 0.037	0.051 ± 0.022	0.040 ± 0.013
U	0.072 ± 0.068	0.041 ± 0.029	0.021 ± 0.0044

(b) First coordinate of the various estimations of the distribution of the state.

	Sampled	0^{th} order	2^{nd} order
O	0.329 ± 0.013	0.322 ± 0.022	0.324 ± 0.017
U	0.335 ± 0.027	0.333 ± 0.011	0.336 ± 0.006

(c) Accuracy of unsupervised sequence segmentation for the same models as in Table 1(a)

	mean accuracy	standard deviation
O	$92.6\% \pm 0.6\%$	$1.6\% \pm 0.2\%$
U	$82.2\% \pm 5.6\%$	$3.6\% \pm 1.4\%$

7 Limitations and future research

There is an inherent limitation to the types of dynamics our method can represent. Langevin dynamics can not represent periodic behaviour. Filtering out periodic signals may be a feature, but this method is not a good match for cases where one wants to analyse periodic behaviour in dynamics.

We had some trouble getting our NLD models to train consistently: the loss would become NaNs or jump several orders of magnitude. We encountered the same problems with NSDE models we trained on the same data, which indicates how involved the training of NSDEs is in general, when using path-based methods like [1]. There are several other methods of training NSDEs, such as [14], [12], and [3], but there is no clear overview of which method works best in what cases. A comparison of the various training strategies for NSDEs would be invaluable.

Moreover, neither the value of the loss on the training or validation set nor the values of any of the loss components were a clear indication of how well the learned landscape or vector field represented the dynamics underlying the data. This need not be a problem if there is a downstream task, like segmentation, that can be used to evaluate the quality of the learned dynamics. Nonetheless, this does restrict the applicability of the method.

8 Conclusion

To provide a more interpretable alternative to Neural Stochastic Differential Equations, we introduce Neural Langevin Dynamics, in which we replace the general drift term by the gradient of a trainable energy function and let the system evolve by Langevin dynamics. The gain in interpretability comes both from the better options for visualisations of scalar functions over general vectors fields, and the better options for downstream processing in the form of classical methods from optimisation and analysis. With these methods, we can extract important properties such as stationary distributions and discrete states. When we apply NLD to an example problem, it recovers the discrete states underlying the data. These results show promise that NLD is able to combine the performance of neural networks with the interpretability of classical differential equation based models.

References

[1] X. Li, T.-K. L. Wong, R. T. Q. Chen, and D. Duvenaud. "Scalable Gradients for Stochastic Differential Equations". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by S. Chiappa and R. Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 26–28 Aug 2020, pp. 3870–3882. URL: https://proceedings.mlr.press/v108/li20i.html.

[2] P. Kidger, J. Foster, X. Li, and T. J. Lyons. "Neural SDEs as Infinite-Dimensional GANs". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 5453–5463. URL: https://proceedings.mlr.press/v139/kidger21b.html.

[3] A. Hasan, J. M. Pereira, S. Farsiu, and V. Tarokh. "Identifying Latent Stochastic Differential Equations". In: *IEEE Transactions on Signal Processing* 70 (2022), pp. 89–104. ISSN: 1941-0476. DOI: 10.1109/tsp.2021.3131723. URL: http://dx.doi.org/10.1109/tsp.2021.3131723.

[4] T. Lelièvre, G. Stoltz, and M. Rousset. *Free energy computations: a mathematical perspective*. London ; Imperial College Press, 2010. ISBN: 978-1-84816-247-1. DOI: 10.1142/p579. URL: https://doi.org/10.1142/p579.

[5] J. Nocedal and S. J. Wright. *Numerical optimization*. 2nd. Springer Series in Operation Research and Financial Engineering. 2006. DOI: 10.1007/978-0-387-40065-5. URL: https://doi.org/10.1007/978-0-387-40065-5.

[6] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. "Gradient Descent Only Converges to Minimizers". In: *29th Annual Conference on Learning Theory*. Ed. by V. Feldman, A. Rakhlin, and O. Shamir. Vol. 49. Proceedings of Machine Learning Research. Columbia University, New York, New York, USA: PMLR, 23–26 Jun 2016, pp. 1246–1257. URL: https://proceedings.mlr.press/v49/lee16.html.

[7] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan. "How to Escape Saddle Points Efficiently". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 1724–1732. URL: https://proceedings.mlr.press/v70/jin17a.html.

[8] C. Zhang and T. Li. "Escape saddle points by a simple gradient-descent based algorithm". In: 34 (2021). Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, pp. 8545–8556. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/47bd8ac1becf213f155a82244b4a696a-Paper.pdf.

[9] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2014. URL: http://arxiv.org/abs/1312.6114.

[10] B. Tzen and M. Raginsky. "Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit". In: *arXiv preprint arXiv:1905.09883* (2019).

[11] S. Peluchetti and S. Favaro. "Infinitely deep neural networks as diffusion processes". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by S. Chiappa and R. Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 26–28 Aug 2020, pp. 1126–1136. URL: https://proceedings.mlr.press/v108/peluchetti20a.html.

[12] H. Ni, L. Szpruch, M. Sabate-Vidales, B. Xiao, M. Wiese, and S. Liao. "Sig-Wasserstein GANs for Time Series Generation". In: *Proceedings of the Second ACM International Conference on AI in Finance*. ICAIF '21. Virtual Event: Association for Computing Machinery, 2022. ISBN: 9781450391481. DOI: 10.1145/3490354.3494393. URL: https://doi.org/10.1145/3490354.3494393.

[13] P. D. Lozano, T. L. Bagén, and J. Vives. *Neural SDEs for Conditional Time Series Generation and the Signature-Wasserstein-1 metric*. 2023. arXiv: 2301.01315 [stat.ML].

[14] S. A. Cameron, T. L. Cameron, A. Pretorius, and S. J. Roberts. "Robust and Scalable SDE Learning: A Functional Perspective". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=xZ6H7wydGl.

[15] F. Dietrich, A. Makeev, G. Kevrekidis, N. Evangelou, T. Bertalan, S. Reich, and I. G. Kevrekidis. "Learning effective stochastic differential equations from microscopic simulations: Linking stochastic numerics to deep learning". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 33.2 (Feb. 2023), p. 023121. ISSN: 1054-1500. DOI: 10.1063/5.0113632. eprint: https://pubs.aip.org/aip/cha/article-pdf/doi/10.1063/5.0113632/16744177/023121_1_online.pdf. URL: https://doi.org/10.1063/5.0113632.

[16] P. Kidger, J. Foster, X. (Li, and T. Lyons. "Efficient and Accurate Gradients for Neural SDEs". In: 34 (2021). Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, pp. 18747–18761. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/9ba196c7a6e89eafd0954de80fc1b224-Paper.pdf.

[17] A. Boral, Z. Y. Wan, L. Zepeda-Núñez, J. Lottes, Q. Wang, Y.-f. Chen, J. R. Anderson, and F. Sha. *Neural Ideal Large Eddy Simulation: Modeling Turbulence with Neural Stochastic Differential Equations*. 2023. arXiv: 2306.01174 [cs.LG].

[18] J. Fagin, J. W. Park, H. Best, and M. O'Dowd. "Latent Stochastic Differential Equations for Modeling Quasar Variability and Inferring Black Hole Properties". In: *ICLR 2023 Workshop on Physics for Machine Learning*. 2023. URL: https://openreview.net/forum?id=x2NOLHCPhg.

[19] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. "Neural Ordinary Differential Equations". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc.,

2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

[20] Y. Rubanova, R. T. Q. Chen, and D. K. Duvenaud. "Latent Ordinary Differential Equations for Irregularly-Sampled Time Series". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/42a6845a557bef704ad8ac9cb4461d43-Paper.pdf.

[21] C. Yildiz, M. Heinonen, and H. Lahdesmaki. "ODE2VAE: Deep generative second order ODEs with Bayesian neural networks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/99a401435dcb65c4008d3ad22c8cdad0-Paper.pdf.

[22] P. Kidger, J. Morrill, J. Foster, and T. Lyons. "Neural Controlled Differential Equations for Irregular Time Series". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 6696–6707. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/4a5876b450b45371f6cfe5047ac8cd45-Paper.pdf.

[23] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho. "Lagrangian Neural Networks". In: *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*. 2019. URL: https://openreview.net/forum?id=iE8tFa4Nq.

[24] S. Greydanus, M. Dzamba, and J. Yosinski. "Hamiltonian Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/26cd8ecadce0d4efd6cc8a8725cbd1f8-Paper.pdf.

[25] P. Toth, D. J. Rezende, A. Jaegle, S. Racanière, A. Botev, and I. Higgins. "Hamiltonian Generative Networks". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=HJenn6VFvB.

[26] A. Botev, A. Jaegle, P. Wirnsberger, D. Hennes, and I. Higgins. "Which priors matter? Benchmarking models for learning latent dynamics". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*. Nov. 2021. URL: https://openreview.net/forum?id=qBl8hnwR0px.

[27] H. A. Kramers. "Brownian motion in a field of force and the diffusion model of chemical reactions". In: *Physica* 7.4 (1940), pp. 284–304. DOI: 10.1016/S0031-8914(40)90098-2. URL: https://doi.org/10.1016/S0031-8914(40)90098-2.

[28] P. A. Markowich and C. Villani. "On the trend to equilibrium for the Fokker-Planck equation: an interplay between physics and functional analysis". In: *Mat. Contemp* 19 (2000), pp. 1–29.

[29] N. Berglund. "Kramers' law: Validity, derivations and generalisations". In: *Markov Processes And Related Fields* 19.3 (2013), pp. 459–490. URL: https://arxiv.org/abs/1106.5799.

[30] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014). DOI: 10.3115/v1/d14-1179. URL: http://dx.doi.org/10.3115/v1/D14-1179.