Proceedings of Machine Learning Research vol 145:1–20, 2022 3rd Annual Conference on Mathematical and Scientific Machine Learning

Adaptive sampling methods for learning dynamical systems

Zichen Zhao E0444180@U.NUS.EDU
Department of Mathematics, National University of Singapore, Singapore 119076

Qianxiao Li QIANXIAO@NUS.EDU.SG
Department of Mathematics, National University of Singapore, Singapore 119076

Editors: Bin Dong, Qianxiao Li, Lei Wang, Zhi-Qin John Xu

Abstract

Learning dynamical systems from observed trajectories is a fundamental problem in data-driven science and engineering. While many existing works focus on improving model architectures or training methods, less attention has been directed at how to effectively sample training data to give rise to accurate models. In particular, one of the most basic problems is to select the length of sampled trajectories that balances computational overhead due to sampling and the quality of learned models. This paper deals with the task of improving sampling efficiency for learning dynamics. We first formulate proper target risks to evaluate the model performance of learning in the dynamical setting. This allows us to connect generalization to matching empirical measures with specific target measures. In line with this observation, we propose a class of adaptive algorithms to find effective sampling strategies that control the length of sampled trajectories. Through numerical experiments, we show the adaptive algorithms can achieve more accurate results given a sampling budget compared to baseline sampling methods.

Keywords: Adaptive Sampling; Machine Learning; Dynamical Systems.

1. Introduction

Modelling dynamical systems based on observed trajectories is an important goal in numerous domains of science and engineering. With growing data availability and computational power, data-driven methods have become a promising alternative to traditional modelling techniques for studying dynamical systems. This gives rise to a host of new mathematical and algorithmic problems on the intersection of machine learning and dynamical systems. On this front, many previous works (Brunton et al. (2016); Hamzi and Owhadi (2021); Lin et al. (2022a); Raissi (2018); Williams et al. (2015)) focus on developing new model architectures or training methods, such as (sparse) regression of nonlinear dynamics using pre-defined dictionary functions (Brunton et al. (2016)), combining neural networks and partial differential equations to model spatio-temporal data, e.g. for solution of high-dimensional PDEs (Raissi (2018); Han et al. (2018)), system identification (Long et al. (2018)) or computing important quantities associated with the dynamics, including committor functions (Khoo et al. (2019); Li et al. (2019)) , quasi-potentials (Lin et al. (2022a)), and invariant distributions (Chen et al. (2021); Lin et al. (2022b)).

However, relatively little attention has been paid to studying efficient data (trajectory) sampling methodologies. On the one hand, for some applications where abundant training data can be directly generated from numerical simulations at little computational overheads, sampling issues often do not pose significant challenges. On the other hand, in real-life scenarios where the cost to generate training trajectories can be very high, efficient data sampling strategies become a crucial aspect of

a scalable data-driven workflow. Examples of such include *ab initio* molecular dynamics simulation (Tuckerman (2002)), quasi-potential landscape construction (Zhou and Li (2016)) and learning dynamical models from experimental data. In these applications where data generation cost is significant, it is crucial to balance the trade-off between sampling cost and the performance of the learned dynamical model. In fact, one of the most basic yet under-explored question is how long the sampled trajectories should be to learn a sufficiently accurate approximation of the dynamical system.

Lacking systematic sampling algorithms for dynamics learning may lead to less accurate models. This is a chicken-and-egg problem in which a correct model (or at least some knowledge of the model) is often a prerequisite for an efficient sampling strategy, while properly sampled data is indispensable for learning an accurate model. Without adaptive sampling methods, generalization performance of learning-based approaches (Brunton et al. (2016); Raissi (2018)) rely on strong prior knowledge, such as proper choices of basis functions to reduce the dynamical complexity, to achieve high accuracy with possibly limited number and diversity of samples. Although there are many generalization analyses in traditional statistical learning (Hastie et al. (2001)), these guarantees are often based on the assumption that the training data are independent and identically distributed (IID). More importantly, they follow the same distribution as the test data that ultimately determine the model's performance. It turns out that neither of these assumptions are appropriate for a large variety of tasks involving learning dynamics. We will make this notion precise in Sec. 2. To deal partially with this issue, references (Foster et al. (2020); Mania et al. (2020)) propose active learning methods for model identification with guarantees based on one long controllable trajectory. Their methods are currently limited to controllable dynamics which can be expressed as linear combinations of known features. However, in practical scenarios the sampler may be a black-box simulator, and may not be controllable. Thus, it is important to develop generic sampling methods that can be applied to a wide variety of scenarios involving learning dynamical systems.

To address these challenges, we propose a general and principled approach to adaptively select the length of sampled trajectories used to train machine learning models for modelling dynamics. To do so, we formulate the precise problem of learning dynamics by distinguishing different types of evaluation metrics appropriate for different practical goals. Based on the selected metric, the sampling strategy attempts to match the sampled data distribution (by selecting the trajectory length) with a target distribution associated with the evaluation metric. The sampling strategy is based on an upper bound of the generalization error in learning dynamics, which connects the latter with the 1-Wasserstein distance between the target distribution and the empirical distribution. Based on this observation, we propose adaptive sampling algorithms to determine an appropriate sampling trajectory length to learn an unknown dynamics given the initial condition (the starting points of trajectories). Through numerical experiments, we demonstrate our sampling algorithms lead to efficient learning of dynamical systems and show a substantial improvement over baseline sampling methods with pre-determined trajectory lengths. Moreover, we demonstrate that our algorithms are robust to different types of dynamical systems, practical goals and learning model architectures.

2. Mathematical Formulation of Learning Dynamics

In this section, we introduce the basic mathematical formulation of learning dynamics. We pay particular attention to different types of possible error metrics arising in practice, and how they translate to different sampling requirements.

Since most continuous dynamics are simulated by time discretization, here we consider a discrete time dynamical system on $\mathbb{R}^m$

$$x_{t+1} = f_*(x_t), \tag{1}$$

where $f_* : \mathbb{R}^m \to \mathbb{R}^m$ is Lipschitz continuous. We assume that we have access to a simulator for this dynamics that produces sample trajectories. More specifically, given a initial condition consisting of N points $\{x_0^i\}_{i=1}^N$ drawn IID from some initial distribution μ_0, the sampling data is collected from N trajectories of length T, and is partitioned as input-output pairs

$$\left\{(x_j^i, x_{j+1}^i) \mid x_{j+1}^i = f_*(x_j^i), i \in \{1, 2, \ldots, N\}, j \in \{0, 1, \ldots T-1\}\right\}. \tag{2}$$

We denote the empirical measure associated with this dataset as

$$\mu_{N,T} := \frac{1}{NT} \sum_{i=1}^{N} \sum_{j=0}^{T-1} \delta_{x_j^i}. \tag{3}$$

Here, δ_x denotes the Dirac point mass at x. Given a model hypothesis space $\mathcal{H} := \{\text{Lipschitz } \hat{f}(\cdot\,;\theta) : \mathbb{R}^m \to \mathbb{R}^m \mid \text{Lip}(\hat{f}) \le \hat{K}, \theta \in \Theta\}$, the training step is to learn an approximation $\hat{f}(x;\hat{\theta})$ of $f_*(x)$ by solving the empirical risk minimization problem over sampling measure. In particular, an empirical risk minimizer $\hat{\theta}$ is defined as

$$\hat{\theta} \in \operatorname*{argmin}_{\theta \in \Theta} L_{N,T}(\theta), \tag{4}$$

with

$$\begin{aligned} L_{N,T}(\theta) :&= \frac{1}{NT} \sum_{i=1}^{N} \sum_{j=0}^{T-1} d\left(x_{j+1}^i, \hat{f}(x_j^i;\theta)\right) \\ &= \int d\left(f_*(x), \hat{f}(x;\theta)\right) \, \mathrm{d}\mu_{N,T}. \end{aligned} \tag{5}$$

Here, the loss function $d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is a semi-metric satisfying *ρ-triangle inequality*

$$d(x_1, x_2) \le \rho\left(d(x_1, x_3) + d(x_3, x_2)\right) \tag{6}$$

for any $x_1, x_2, x_3 \in \mathbb{R}^m$, and $\rho \ge 1$ is a constant depending on d.

Remark 1 *We consider the slightly more general formulation where d is a semi-metric to cover the popular choice of $d(x_1, x_2) = \|x_1 - x_2\|^2$, i.e. mean-squared error. For other applications, e.g. L^1 regression, $d(x_1, x_2) = \|x_1 - x_2\|_1$ can be a metric.*

At this point, we emphasize a key difference from traditional supervised learning, in that the evaluation of the performance of $\hat{f}$ varies with application. In the classical supervised learning scenario, the model performance L_* is equal to the expectation of the empirical error over the IID training samples. In the current dynamical situation, there are two main differences. First, multiple performance metrics can be chosen to evaluate the learned model according to different practical goals. Second, there involve performance metrics that are associated with necessarily different distributions from the sampling distribution $\mu_{N,T}$.

In this simplest case where we are interested in one-step predictions from initial conditions drawn from μ_0 and the training data are also one-step trajectories, then the model performance is measured as in the usual supervised learning setting

$$\begin{aligned} L_*(\theta) &= \mathop{\mathbb{E}}_{x\sim\mu_0}\left[\,\mathrm{d}\left(f_*(x),\hat{f}(x;\theta)\right)\right] \\ &= \int d\left(f_*(x),\hat{f}(x;\theta)\right)\,\mathrm{d}\mu_0 \\ &= \mathbb{E}[L_{N,1}(\theta)]. \end{aligned} \tag{7}$$

However, there are other practical problems for learning dynamics that involve distinct performance metrics. For example, one may be interested to recover (an approximation) of the right hand side vector field f_*, over some domain of interest S (e.g. Lin et al. (2022a)). Then, the model performance takes a different form

$$\begin{aligned} L_*(\theta) =& \int \left(f_*(x)-\hat{f}(x;\theta)\right)^2\,\mathrm{d}\mu_* \\ =& \frac{1}{|S|}\int \left(f_*(x)-\hat{f}(x;\theta)\right)^2 \mathbb{1}_S\,\mathrm{d}x; \end{aligned} \tag{8}$$

As a further example, we may be interested to predict the dynamics' long time behavior, such as the evolution of the Van der Pol oscillator on its limit cycle. To do so, it is necessary to approximate the vector field f_* over the invariant measure μ_{inv} of the dynamical system, i.e.

$$L_*(\theta) = \int \left(f_*(x)-\hat{f}(x;\theta)\right)^2\,\mathrm{d}\mu_{\text{inv}}. \tag{9}$$

This motivates us to define more precisely the types of evaluation methods we encounter in practical applications. From the discussions above, a natural candidate for model performance is

$$L_*(\theta) := \int d(f_*(x),\hat{f}(x;\theta))\,\mathrm{d}\mu_*, \tag{10}$$

where the target measure μ_* takes variable forms depending on application. For one-step prediction tasks we take $\mu_* = \mu_0$; for approximation of vector fields over compact domains we take $\mu_* = \mathbb{1}_S/|S|$, i.e. the uniform measure; for long-term trajectory analysis we take μ_* to be the invariant measure.

One important case not directly covered by (10) is trajectory prediction over a finite time interval $T > 1$. However, we observe that in this case there is an upper bound of the expected error in the form of (10). More precisely, the expected multiple-step trajectory prediction error starting from some distribution $\hat{\mu}_*$ at time t satisfies

$$\begin{aligned} L_*^T(\theta) &= \mathop{\mathbb{E}}_{x_t\sim\hat{\mu}_*}\left[\sum_{j=1}^{T} d\left(x_{t+j},\hat{f}^j(x_t)\right)\right] \\ &= \int \sum_{j=1}^{T} d\left(f_*^j(x),\hat{f}^j(x;\theta)\right)\,\mathrm{d}\hat{\mu}_* \\ &\le \int d\left(f_*(x),f(x;\theta)\right)\,\mathrm{d}\mu_*, \end{aligned} \tag{11}$$

where T is the length of trajectories for evaluation, and

$$\mu_* = \sum_{l=0}^{T-1} \hat{K}^{T-l-1} (f_* + \mathbb{1})^l \# \hat{\mu}_*. \tag{12}$$

That is, μ_* is a mixture of pushforward distributions of $\hat{\mu}_*$ by f_*. Recall that $\hat{K}$ is the uniform Lipschitz constant of all hypotheses in $\mathcal{H}$. The proof of (12) can be found in Appendix E. Given a mapping $h : \mathbb{R}^m \to \mathbb{R}^m$ and measure ν with bound support M in $\mathbb{R}^m$, the *push forward* of ν is defined to be the measure $h\#\nu$ by $(h\#\nu)(B) = \nu\left(h^{-1}(B)\right)$ for $B \subseteq M$.

In summary, (10) can be seen as a general formulation of the performance indicator of learned dynamical models, which is captured by the target measure μ_* that varies according to different applications. In particular, observe that unlike traditional supervised learning, the sampling measure $\mu_{N,T}$ does not in general converge, as $N \to \infty$ and/or $T \to \infty$, to μ_*. Hence, balancing sampling cost and model performance becomes an interesting question to be investigated. In the next section, we show that indeed, the discrepancy between $\mu_{N,T}$ and μ_* bounds the model performance error. Thus, this forms the basis of our proposed algorithms that aim to adaptively minimize this discrepancy.

Remark 2 *While (11) provides an upper bound on multi-step trajectory prediction error based on one-step prediction error, this may not be a tight bound in practice. For example, the dynamics may include fast varying temporal modes that average themselves for sufficiently long observation horizons, thus may become easier to learn when considering data spanning multiple time steps. Examples include Markov state models in Noé and Nuske (2013) and VAMP methods in Wu and Noé (2020).*

3. Adaptive Sampling Algorithms

Following the previous formulation, in this section we present a precise result that connects model performance and the discrepancy between the empirical and the target measure. This then allows us to devise sampling strategies to minimize the discrepancy. We start by recalling the definition of the 1-Wasserstein distance that we will use to measure distances between probability distributions.

Definition 3 (1-Wasserstein distance) *The 1-Wasserstein distance between two probability measures μ and ν is defined as*

$$W_1(\mu, \nu) := \inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathbb{R}^m \times \mathbb{R}^m} d(x, y)\, \mathrm{d}\gamma(x, y) \tag{13}$$

where $\Gamma(\mu, \nu)$ denotes the collection of all measures on $\mathbb{R}^m \times \mathbb{R}^m$ with marginals μ and ν on the first and second component respectively.

For μ, ν with bounded support , we also have the following dual representation

$$W_1(\mu, \nu) = \sup \left\{ \int_{\mathbb{R}^d} g(x) \mathrm{d}(\mu - \nu)(x) \mid \text{ Lipschitz } g : \mathbb{R}^d \to \mathbb{R}, \mathrm{Lip}(g) \leq 1 \right\}. \tag{14}$$

The model performance is generally caused by a combination of approximation, optimization and generalization errors. In the following proposition we make this precise and show that in lieu of approximation and optimization errors, the target risk is then bounded by the discrepancy between μ_* and $\mu_{N,T}$ measured in the 1-Wasserstein distance.

Proposition 4 (Upper bound of the target risk) *Given the hypothesis space* $\mathcal{H} := \{\hat{f}(\cdot\,;\theta) : \mathbb{R}^m \to \mathbb{R}^m \mid \theta \in \Theta\}$*, for any parameter* $\theta \in \Theta$*, the target risk with respect to* μ_* *can be bounded by the summation of three parts, model bias, empirical error and distribution discrepancy:*

$$L_*(\theta) \leq \inf_{\theta\in\Theta} L_*(\theta) + \left| L_{N,T}(\theta) - \inf_{\theta'\in\Theta} L_{N,T}(\theta') \right| + 2C_* W_1(\mu_*, \mu_{N,T}) \tag{15}$$

where $C_* = \sup_{\theta\in\Theta} \mathrm{Lip}\left(d(f_*(\cdot), \hat{f}(\cdot\,;\theta)\right)$.

Moreover, if $\mathcal{H}$ *is a universal approximator (i.e.* $\inf_\theta L_*(\theta) = 0$*) and* $\hat{\theta}$ *is an empirical risk minimizer, then its target risk* $L_*(\hat{\theta})$ *can be bounded by the 1-Wasserstein distance between empirical distribution and target distribution,*

$$L_*(\hat{\theta}) \leq 2C_* W_1(\mu_*, \mu_{N,T}). \tag{16}$$

Proposition 4 shows that one can select the sampling strategy that minimizes the 1-Wasserstein distance with target distribution to reduce target risk. More specifically, two parameters of the sampling strategy, the number of starting points N and trajectory length T, can be well designed to make the generated trajectories most efficiently explore the target measure. Compared with traditional supervised learning, the key difference is that the empirical measure $\mu_{N,T}$ need not monotonically approach μ_* when either of its arguments goes to infinity. Hence, selecting the best sampling parameters becomes an optimization problem that one needs to solve on the fly.

In practical scenarios, N starting points are usually IID drawn from some fixed initial distribution μ_0, so in this paper we only focus on the second part, i.e., given the initial points, how to find the appropriate trajectory length to sample for learning dynamics. The adaptive selection of the initial conditions is an interesting problem that is out of the scope of the current work.

3.1. Reformulate sampling as optimization

In view of the bound derived in Proposition 4, once μ_* is selected according to the goal of learning dynamics, we can formulate the sampling problem of finding optimal T as an optimization problem that minimizes $W_1(\mu_*, \mu_{N,T})$ with a cost term ϕ:

$$\begin{aligned} &\min_{T} \;\; W_1(\mu_*, \mu_{N,T}) + \phi(N, T) \\ &\text{subject to } \mu_{N,T+1} = \frac{1}{N(T+1)} \left(N\mu_{N,0} + NT\mathcal{L}(f_*)\mu_{N,T}\right) = \frac{1}{T+1} \sum_{t=0}^{T+1} \mathcal{L}(f_*)^t \mu_{N,0}, \end{aligned} \tag{17}$$

where $\mathcal{L}(f_*)$ is a transform operator satisfying $\mathcal{L}(f_*)\ \delta_x = \delta_{f_*(x)}$ for $x \in \mathbb{R}^m$.

Example 1 *As a warm-up, we take a dynamical system with a limit cycle as a toy model to demonstrate that the upper bound* (16) *and the subsequent optimization formulation* (17) *are sensible. The problem is to model the dynamics*

$$\begin{cases} \frac{dx}{dt} = x + y - x(x^2 + y^2), \\ \frac{dy}{dt} = y - x - y(x^2 + y^2), \end{cases} \tag{18}$$

over the unit circle. Here μ_0 *is selected as the uniform distribution over* $[1.5, 2]^2$*, and the target risk* L_* *is selected as mean square error over the uniform distribution* μ_* *on the unit circle.*

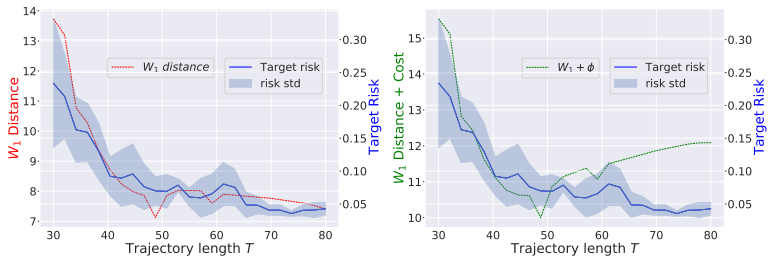

Figure 1: The red line represents the 1-Wasserstein distance via trajectory length. The blue line represents the average of target risks with the standard deviation shaded in light blue calculated from 10 experiment runs. The green line represents the distance with cost terms.

We feed data sampled from trajectories with different lengths into a shallow fully-connected neural network, and calculate corresponding 1-Wasserstein distances and target risks in Fig 1. We observe that the target risk and $W_1(\mu_*, \mu_{N,T})$ vary together as predicted from Proposition 4. Moreover, as the trajectory length increases, the decrement of both quantities become insignificant. This implies that trajectories longer than ~ 50 no longer gives meaningful improvement, but incurs computational cost (thus the rise in the green line). Hence, solving the optimization problem (17) may lead to a sampling strategy that selects trajectories of length close to 50, without wasting additional computation.

It should be noted that the optimization problem (17) cannot be directly solved since f_* is often unknown and μ_* may be inaccessible (i.e. one cannot sample from it efficiently) in some cases, such as learning dynamics over an unknown invariant measure. Hence, adaptive algorithms needs to be proposed for different situations. We discuss the basic sampling algorithm and its variants in the next section.

3.2. The basic sampling algorithm and its variants

In this section, we introduce two variants of the adaptive sampling algorithm according to the accessibility of target measure μ_*. The first variant assumes that we have access to the target distribution μ_*. In other words, we can independently sample data from μ_* at reasonable cost that is negligible compared with the cost of simulating the dynamics defined by f_*. An example is recovering the decomposition of vector field (Lin et al. (2022a)), where μ_* is taken as uniform distribution. In this case, we only need a stopping-criterion from (17) to determine if we continue to sample data in the next step or we stop. Here, the stopping-criterion is selected as the relative error of $W(T) := W_1(\mu_*, \mu_{N,T}) + \phi(N, T)$ for fixed N. Moreover, the basic setting considers selecting a common stopping time T for all N initial conditions. The basic algorithm is summarized in Alg. 1.

The cost function ϕ can be naturally taken as linear form $\phi(N, T) = \lambda NT$, $\lambda > 0$, under the assumption that the sampling cost is proportional to the number of data points and the time horizon. This assumes that the black-box simulator is computed in a serial manner. Different choices of ϕ may be required if one considers parallel computation in N, say $\phi(N, T) = \lambda T$ or $\phi(N, T) = \lambda T \log(N)$. In all numerical examples in this paper, we take $\phi(N, T) = \lambda NT$ with $\lambda = 0.001$. Nevertheless, all algorithms are applicable to general selections of the cost function.

Algorithm 1 Adaptive Sampler (μ_* accessible, common trajectory-wise)

0. given: initial samples $\mathcal{D}_0 = \{x_j^i : i = 1, \dots, N, j = 0, \dots, T_0\}$ with N starting points $\{x_i\}_{i=1}^N$ and trajectory length T_0, simulator of the dynamics, target distribution μ_* sampler, relative error tolerance $\epsilon > 0$, sampling step $\Delta T \in \mathbb{N}^+$;

1. set: sampling dataset $\mathcal{D} \leftarrow \mathcal{D}_0$, empirical measure $\hat{\mu} \leftarrow \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \delta_x$, $T \leftarrow T_0$;

2. **do**

sample $x_{T+1}^i, \cdots, x_{T+\Delta T}^i$ from x_T^i by the simulator for $i = 1, \dots, N$;

update $\mathcal{D} \leftarrow \mathcal{D} \cup \{x_{T+1}^i, \cdots, x_{T+\Delta T}^i : i = 1, \dots, N\}$;

update $\hat{\mu} \leftarrow \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \delta_x$, $T \leftarrow T + \Delta T$;

while $\frac{|W(T-\Delta T) - W(T)|}{|W(T)|} > \epsilon$;

3. output: NT samples $\mathcal{D} = \{x_j^i : i = 1, \dots, N, j = 0, \dots, T\}$.

In Alg. 1, we also need to calculate the 1-Wasserstein distance between target distribution and empirical distribution in the stopping-criterion. To do this, we select M points that are IID drawn from μ_* to construct the discretization of μ_*, and the optimal transport problem over two discrete sample sets can be solved by the *Sinkhorn* algorithm using the package (Flamary et al. (2021)). In all subsequent experiments, M is taken as $|\mathcal{D}|$, where $|\mathcal{D}|$ is the size of the sampling dataset.

Following the basic algorithm presented in Alg. 1, we now discuss some variants that are needed to handle the different scenarios we motivated earlier. The first variant situation is when the target distribution μ_* is inaccessible, i.e. one cannot (cheaply) draw samples from it. An example is to learn the evolution over *a priori* inaccessible limit cycle for Van der Pol oscillator. In this case, we cannot directly generate samples from μ_*, and so we have to devise a method to adaptively generate data whose distribution becomes closer and closer to μ_* as the learning progresses. The process is as follows. First, a temporary simulator is generated from $\hat{f}(x; \hat{\theta})$ by solving the empirical risk minimization over current sampled data. This serves as an approximation of f_* and we assume that calling $\hat{f}$ incurs negligible cost, and further that it can be used to create an approximate sampler from μ_*. Then, trajectories and approximate samples from μ_* are generated using the temporary simulator. Finally, substituting the temporary simulator and target measure sampler into Alg. 1, T is updated in the while loop. Repeating the *do-while* iteration, one can sample and learn f_* step by step.

Example 2 *Considering the dynamical system of uniform limit cycle in Example 1, our goal is to model the dynamics over its inaccessible invariant measure μ_{inv}, i.e., the uniform distribution over the unit circle. Here the Algorithm 2 is applied to find the sampling strategy for single starting points drawn from uniform measure over $[1.5, 2]^2$. The target risk is defined as mean square loss over μ_{inv} to compare the model performance after different iterations of adaptive sampling. Temporary invariant measure samplers are generated by infinite compositions of learned simulators.*

From the left of Fig. 2, we can see the learned invariant set from single point becomes closer and closer to the unit circle as the iteration of do-while processes, which illustrates our adaptive method is able to capture the inaccessible measure after refining sampling strategy via iterations.

This enables us to train the neural network to achieve good performance as shown in the right of Fig. 2.

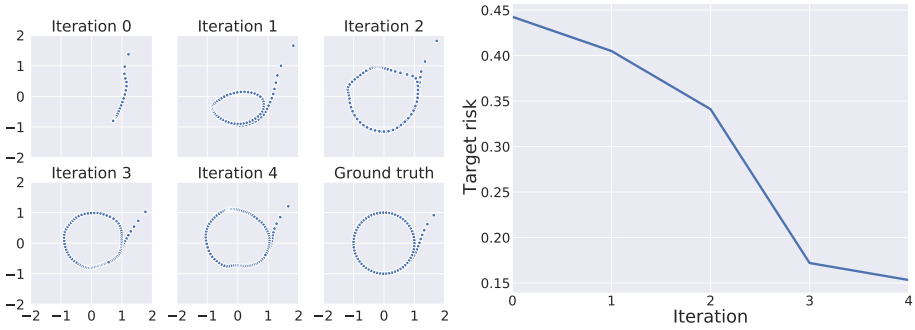

Figure 2: Alg. 2 for limit cycle learning. Left: the first five graphs show trajectories generated from the learned system after 0-4 iterations of *do-while* and the last one is the ground truth. Right: target risk via iterations.

For sampling problem with one-point initial measure, one trajectory length T as output is enough. However, for $N > 1$, there is usually no uniform trajectory length T for all initial points in practice. We have a modified optimization of (17) over $\{T_i\}_{i=1}^N$:

$$\begin{aligned} &\min_{T_1,\cdots,T_N} \quad W_1(\mu_*, \frac{1}{\sum_i T_i}\sum_i T_i \mu^i_{T_i}) + \phi(T_1,\cdots T_N) \\ &\text{subject to } \mu^i_{T_i+1} = \frac{1}{(T_i+1)}\left(\delta_{x_0^i} + T_i \mathcal{L}(f_*)\mu^i_{T_i}\right), \text{for } i \in \{1,\cdots N\}, \end{aligned} \tag{19}$$

where $\mu^i_{T_i}$ represents the empirical measure contributed by i^{th} trajectory. In the adaptive trajectory-wise case, we notice that if a trajectory overlaps with other one, the extension of this trajectory contributes less to the decreasing of W_1 distance since the overlapping part has been collected through some previous trajectory. Based on this observation, we propose algorithms including overlapping check to approximately solve (19). We denote $\mathcal{M}$ as the set consist of indexes in which trajectories have the potential to be explored further. Once some trajectory evolves to the region covered by previous trajectory, the corresponding index is deleted from $\mathcal{M}$ and the output length of this trajectory is labelled as the temporal location of first intersection. Following the above modification, the advanced version of common trajectory-wise Alg. 1 and 2 are adaptive trajectory-wise Alg. 3 and 4 respectively. More details about Alg. 3 and 4 can be founded in Appendix.E, and all four algorithms for different situation are summarized in Table 1.

Example 3 *Considering the dynamical system of uniform limit cycle in Example 1, our goal is to modelling the dynamics over its accessible invariant measure μ_{inv}. Here the Alg. 1 and 3 are respectively applied to find the sampling strategy for three starting points $A[1.0, 1.0]$, $B[-1.0, 1.5]$, $C[1.5, -1.5]$. The output trajectories' length for A,B,C is 20,25,40 respectively from Alg. 3 and the target risk of learned model is 0.090, whereas the uniform trajectory length 35 is generated from Alg. 1 and the target risk of learned model is 0.113. Each target risk is calculated as the mean of 10*

experiment runs. As shown in Fig. 3, three trajectories generated by Alg. 3 just cover the unit circle without any overlapping. This illustrates our adaptive trajectory-wise algorithm can substantially improve the sampling efficiency and model performance compared to the basic one.

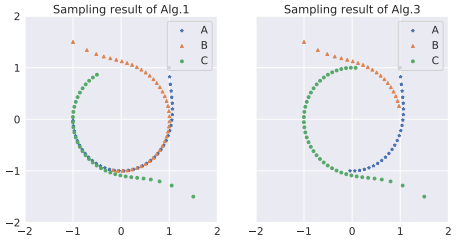

Figure 3: Alg.1 output and Alg.3 output for limit cycle learning with accessible invariant measure.

Algorithm 2 Adaptive Sampler (μ_* inaccessible, common trajectory-wise)

0. given: initial samples $\mathcal{D}_0 = \{x_j^i : i = 1, \dots, N, j = 0, \dots, T_0\}$ with N starting points $\{x_i\}_{i=1}^N$ and trajectory length T_0, simulator of the dynamics, W_1 relative tolerance $\epsilon > 0$, sampling step $\Delta T \in \mathbb{N}^+$;

1. set: sampling dataset $\mathcal{D} \leftarrow \mathcal{D}_0$, empirical measure $\hat{\mu} \leftarrow \dfrac{1}{|\mathcal{D}|} \displaystyle\sum_{x \in \mathcal{D}} \delta_x$, $T \leftarrow T_0$;

do

2. solve empirical risk minimization over $\hat{\mu}$: $\hat{\theta} \leftarrow \underset{\theta \in \Theta}{\operatorname{argmin}} L_{N,T}(\theta)$;

3. generate μ_* sampler from $\hat{f}(x; \hat{\theta})$;

4. solve the optimization problem (17) by Alg. 1: $\hat{T} \leftarrow \underset{T' \in [T+\Delta T, \infty)}{\operatorname{argmin}} W_1(\mu_*, \hat{\mu}) + \phi(N, T')$;

5. sample $x_{T+1}^i, \cdots, x_{\hat{T}}^i$ from x_T^i by the simulator for $i = 1, \dots, N$;

6. update $\mathcal{D} \leftarrow \mathcal{D} \cup \{x_{T+1}^i, \cdots, x_{\hat{T}}^i : i = 1, \dots, N\}$;

7. update $\hat{\mu} \leftarrow \dfrac{1}{|\mathcal{D}|} \displaystyle\sum_{x \in \mathcal{D}} \delta_x$, $T \leftarrow \hat{T}$;

while $\frac{|W(T-\Delta T) - W(T)|}{|W(T)|} > \epsilon$;

output: NT samples: $\mathcal{D} = \{x_j^i : i = 1, \dots, N, j = 1, \dots, T\}$.

Remark 5 *When μ_* is accessible, while one can directly sample N data points IID from μ_*, here we consider the case where N is fixed, but T is to be determined. For any finite N, $\mu_{N,T}$ with $T = 1$ (data generated from μ_*) need not be the closest empirical distribution to μ_* amongst all possible T choices (see Appendix.B for a numerical illustration). Thus, Alg.1 and Alg.3 remain relevant. Treating the sampling of initial conditions and length of the dynamics separately is especially important in applications where cost of the former is significant compared with latter, e.g. experimental measurement of complex processes.*

	μ_* accessible	μ_* inaccessible
common trajectory-wise	Alg. 1	Alg. 2
adaptive trajectory-wise	Alg. 3	Alg. 4

Table 1: Four algorithms for different situations are summarized here.

4. Numerical Experiments

We now illustrate using various numerical examples that the proposed algorithms can efficiently sample data for learning dynamics. In section 4.1, we show that our algorithms are readily to dynamics learning problems with different goals. Next, in section 4.2, through comparisons under fixed error tolerance and fixed budget, we show our methods improve sampling efficiency substantially over baseline sampling. Finally in section 4.3, we combine our methods with data-driven quasi-potential learning (Lin et al. (2022a)) and sparse identification of nonlinear dynamics (SINDy) (Brunton et al. (2016)), to show our sampling algorithms is flexible to be applied to most learning methods.

4.1. Adapting to different types of dynamical systems

In this part, we apply our sampling algorithms to learning problems with different goals: (1) recovering vector field for Bi-Stable system by Alg. 3; (2) modelling the Von der Pol oscillator over its inaccessible limit cycle by Alg. 4; (3) modelling dynamics of Lorenz system on its inaccessible strange attractor by Alg. 2. The hypothesis spaces in this part are generated by fully-connected neural networks with ReLu activations.

Dynamical systems	Target distribution	Goal	Algorithm
Lorenz 63	inaccessible μ_*	prediction over μ_*	2
Bi-Stable system	accessible uniform μ_*	recovering the vector field	3
Van der Pol oscillator	inaccessible μ_*	prediction over μ_*	4

Table 2: The settings of three dynamics learning numerical experiments.

4.1.1. LORENZ 63

We first apply adaptive sampling to learning dynamics from one long trajectory problem. In the dynamic learning problem of Lorenz 63: $\dot{x} = 10(y - x), \dot{y} = x(8/3 - z) - y, \dot{z} = xy - 28z$, the setting $\mu_* = \dfrac{1}{T_2 - T_1} \displaystyle\sum_{t=T_1}^{T_2-1} \mathcal{L}(f_*)^t[\mu_0]$ is inaccessible and our goal is to model the dynamics over the strange attractor from one trajectory, which starts from a single point drawn from the uniform distribution μ_0 over $[10, 11]^3$. The target risk $L_* = \dfrac{1}{T_2 - T_1} \displaystyle\int \sum_{t=T_1}^{T_2} \left(f_*^t(x) - \hat{f}^t(x; \theta) \right)^2 \mathrm{d}\mu_0$. Here we take $T_1 = 190,\ T_2 = 200$. Sampling results similar to Example 3 can be found in Fig. 4, in which our algorithm captures the strange attractor gradually.

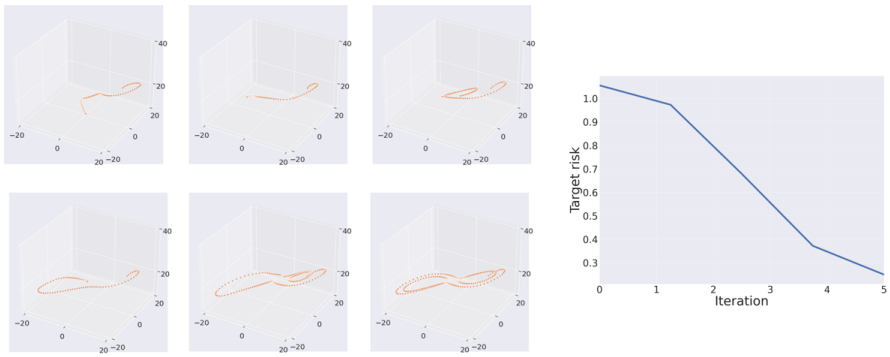

Figure 4: Alg. 2 for Lorenz 63 learning. Left: the first five graphs show trajectories generated from the learned system after 0-4 iterations and the last graph shows the sampling result based on the ground truth. Right: target risk via iterations.

4.1.2. BI-STABLE SYSTEM

In the dynamics learning of the following system

$$\begin{cases} \dot{x} = \frac{1}{5}x\left(1-x^2\right) + y(1+\sin x) \\ \dot{y} = -y + 2x\left(1-x^2\right)(1+\sin x). \end{cases} \tag{20}$$

with two stable states $(-1,0),(1,0)$, our goal is to recover the right hand side vector field. The accessible μ_* is taken as $\frac{1}{|S|}\mathbb{1}_S$ where the region of interest $S=[-3,3]^2$ and the target risk $L_* = \frac{1}{|S|}\int \left(f_*(x)-\hat{f}(x;\theta)\right)^2 \mathbb{1}_S \ \mathrm{d}x$. 5 starting points are drawn from the uniform distribution on the domain $\mathcal{S}$: A $[-2,2]$, B $[0,1.5]$, C $[2.0,-2.0]$, D $[1.0,-2.0]$, E $[-2.0,-2.0]$. Alg. 3 is applied here to find the sampling strategy and the output length for A,B,C,D,E is 14, 33, 20, 16, 15 respectively. From the sampling result in the right of Fig. 5, we can see trajectories converge faster to the left stationary point than the right one. After 20 steps of exploration from A,C,D,E, these trajectories are close to the left stable stationary and contribute less to the descent of W_1, which leads to earlier stop compared the one from B. This demonstrates that our algorithm can efficiently save sampling cost.

4.1.3. VAN DER POL OSCILLATOR

In the dynamic learning problem of Van der Pol oscillator,

$$\begin{cases} \dot{x} = y \\ \dot{y} = (1-x^2)y - x\dot{y} = (1-x^2)y - x, \end{cases} \tag{21}$$

our goal is to model the dynamics over the inaccessible limit cycle. The target risk L_* is taken as $\int \left(f_*(x)-\hat{f}(x;\theta)\right)^2 \mathrm{d}\mu_{\text{inv}}$. Three initial points are taken as A $[2.0,2.0]$, B $[-2.0,2.0]$, C $[-2.0,2.0]$ and Alg. 4 is applied to find trajectories sampling strategy. The output trajectory length for A,B,C is 22, 28, 44 respectively. From the sampling results in the right of Fig. 4, we can the invariant set

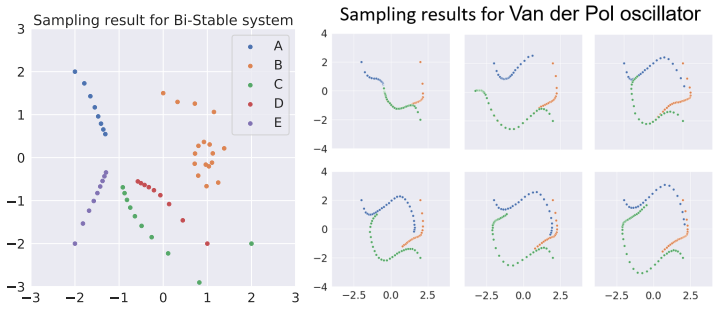

Figure 5: Left: sampling results by Alg. 3 for Bi-Stable system. The distance between points is two steps. Right: sampling results by Alg. 4 for Van der Por oscillator.

converges to the limit cycle gradually and finally three trajectories exactly right cover the limit cycle, which illustrates Alg. 4 is able to both capture the inaccessible target measure and be trajectory-wise adaptive for saving cost.

4.2. Out-performing baseline methods

In this part, we compare the sampling strategies generated from our algorithms and passive baseline sampling methods with fixed number of initial points N and trajectory length T. Random feature model with Gaussian kernel is also applied here. First, we fix the sampling budget NT to compare the target risks we defined before. For baseline sampling, we assume there are two different cases. One case has short trajectories and many initial points, i.e, large N, and small T. The other case deals with long trajectories with few initial points, i.e., small N and large T.

Dynamical systems	Adaptive sampler	Small T large N	Large T small N
Fully connected neural network			
Lorenz 63	**0.2471± 0.0614**	0.4758± 0.0179	0.2471± 0.0614
Bi-Stable system	**0.0396± 0.0065**	0.1048± 0.0271	0.0863± 0.0471
Van der Pol oscillator	**0.0108±0.0072**	0.0340±0.0093	0.0810±0.0102
Random feature model			
Lorenz 63	**0.4230± 0.1114**	0.6587± 0.0109	0.4230± 0.1114
Bi-Stable system	**0.0470± 0.0021**	0.0741± 0.0361	0.0919± 0.0572
Van der Pol oscillator	**0.0840±0.0070**	0.0502±0.0093	0.0661±0.0151

Table 3: Results of target risks with fixed budget NT from 10 experiment runs. The target risk in column 2 is generated by our adaptive algorithms, while the target risk in column 3 (column 4) is generated by passive sampling with small T and large N (large T and small N).

Next, we do comparison under fixed risk tolerances, i.e., calculate the budget NT (number of data points) to achieve the target fixed error. From Table 4, we can see long and few trajectories can-

not learn the Bi-Stable system with respect to uniform distribution while short and many trajectories cannot learn Van der Pol oscillator with respect to the invariant distribution.

Dynamical systems/ error tolerance	Adaptive sampler	Small T large N	Large T small N
Fully connected neural network			
Lorenz 63 (0.5)	**130± 30**	170±30	350± 70
Bi-Stable system (0.05)	**200± 20**	—	350±20
Van der Pol oscillator (0.05)	**120 ± 20**	200± 30	—
Random feature model			
Lorenz 63 (0.5)	**170± 40**	240±50	170± 40
Bi-Stable system (0.05)	**270± 20**	—	330±20
Van der Pol oscillator (0.05)	**310 ± 30**	400± 30	—

Table 4: Results of numbers of required data points with fixed errors' tolerances from 10 experiment runs. The budget in column 2 is generated by our adaptive algorithms, while the budget in column 3 (column 4) is generated by passive sampling with small T and large N (large T and small N).

4.3. Effective combination with existing learning models

4.3.1. LORENZ SYSTEMS IDENTIFYING BY SINDY

Here we first apply our adaptive samplers and sparse identification of nonlinear dynamical systems (SINDy) method to Lorenz 63 learning for two different tasks. The SINDy method approximates f by a generalized linear model of some candidate nonlinear functions, such as constant, polynomial, and trigonometric functions. Here the first task is the long time prediction. Two starting points are drawn from uniform distribution of $\mathcal{S} = [-20, 20]^2 \times [10, 40]$ and target distribution is combination of multi-step push-forward measure of $\mu_{2,0}$: $\mu_*^{(1)} = \dfrac{1}{T_2 - T_1} \displaystyle\sum_{t=T_1}^{T_2-1} \mathcal{L}(f_*)^t(\mu_{2,0})$. Here we take $T_1 = 150$ and $T_2 = 200$, and apply Alg. 4 for this task. The second task is recovering vector filed over $\mathcal{S} = [-20, 20]^2 \times [10, 40]$, so $\mu_*^{(2)}$ is taken as $\mathbb{1}_{\mathcal{S}}/|\mathcal{S}|$ and Alg. 3 is applied. The library of candidate functions in SINDy is constructed with polynomial terms up to third order. As shown in Fig. 6, for the same dynamics learning with different goals, trajectories generated by our adaptive samplers have significant differences. From 10 experiment runs, the target risk of task 1 is 0.1346 ± 0.0231 while the risk the baseline sampling with $T = 200$ is 0.1267 ± 0.0194. The target risk of task 2 is 0.7846 ± 0.1780 while risk of the baseline sampling with $N = 16$ and $T = 50$ is 0.6756 ± 0.0821. All information of the strange attractor is collected through only two trajectories which are fully explored, and too much overlapping is also avoided by the property of trajectory-wise adaptability. In the task of recovering vector, N is taken as 16. We randomly partition them into four subsets and apply Alg. 3 independently to each subset. The sampling result of one subset is shown in the right graph and all trajectories are no longer than 50. The two tasks illustrate that our sampling can be adaptive to the same dynamics learning with different goals.

Next, we show that our method can handle learning higher dimensional dynamical systems with non-trivial attractor sets. We use SINDy to identify the Lorenz '96 model (**?**) whose equation is

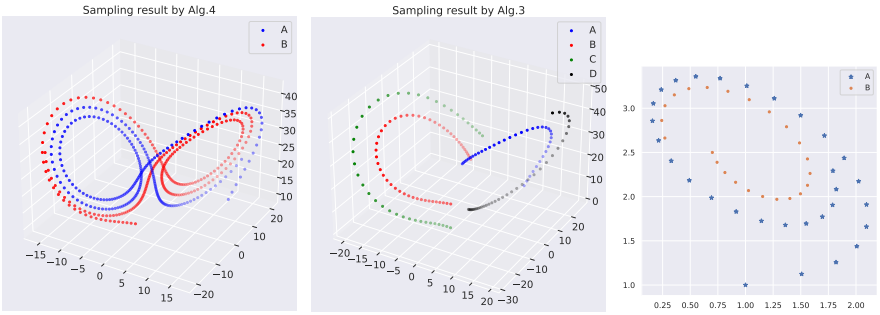

Figure 6: Left: sampling result by Alg. 4 for task 1. Center: sampling result by Alg. 3 for task 2. Right: sub-sampling result for quasipotential computing by Alg. 3.

given by

$$\dot{x}_i = (x_{i+1} - x_{i-2})\, x_{i-1} - x_i + F \tag{22}$$

for $i = 1, 2, \ldots, m$, where it is assumed that $x_{-1} = x_{m-1}, x_0 = x_m, x_1 = x_{m+1}$. The forcing term F is set as 8 and the dimension m is set as 5. The task is set as modeling the long time behavior of the chaotic dynamics. Two starting points are drawn from $[-4, 4] \times [8, 10]^4$ and target distribution is combination of multi-step push-forward measure of $\mu_{2,0}$: $\mu_*^{(1)} = \dfrac{1}{T_2 - T_1} \displaystyle\sum_{t=T_1}^{T_2-1} \mathcal{L}(f_*)^t(\mu_{2,0})$. Here we take $T_1 = 80$ and $T_2 = 100$, and apply Alg. 4 for this task. The library of candidate functions in SINDy is constructed with polynomial terms up to third order. The target risk of adaptive sampling is 0.737 ± 0.142 while the risk of baseline sampling with $N = 2, T = 200$ is 0.614 ± 0.052. This demonstrates the effectiveness of our method in learning high dimensional chaotic systems.

4.3.2. RECOVERING VECTOR FILED BY QUASIPOTENTIAL COMPUTATION

Instead of directly approximating the right hand side vector field, the data-driven quasipotential computing method learns an orthogonal decomposition of the vector field $f(x)$. More specifically, $f(x)$ can be decomposed as $f(x) = -\nabla V(x) + g(x)$ with $\nabla V(x)^T g(x) = 0$. Then two $V(x)$ and $g(x)$ are parameterized by two neural networks $V_\theta(x)$ and $g_\theta(x)$. Considering the following dynamics,

$$\dot{x} = -\frac{1}{2}\frac{\partial U}{\partial x}(x, y) - 2(x + 2y - a - 2b),\ \dot{y} = -\frac{1}{2}\frac{\partial U}{\partial y}(x, y) + 2(2x + y - 2a - b), \tag{23}$$

with the quasipotential $U(x, y) = \left[(x - a)^2 + (x - a)(y - b) + (y - b)^2 - \frac{1}{2}\right]^2$, Alg. 3 is applied is to recover the right hand side vector field. Here $a = 1, b = 2.5$ and μ_* is taken as the uniform distribution of the set $\mathcal{S} = [-0.5, 2.5] \times [1, 4]$. Quasipotential computing needs much more data points to enforce the orthogonal condition, compared with other methods in this paper. Each time we IID draw 2 points from the uniform distribution over $\mathcal{S}$ and apply Alg. 3 to generate two trajectories as shown in Fig. 6. After repeating the above sub-sampling 100 times, we feed all data into the quasipotential model. From 10 experiment runs, for adaptive sampling the target risk is 0.7297 ± 0.1332 and the budget is 3730 ± 635 while the risk of baseline sampling with $N = 100$ and $T = 60$

is 0.6313 ± 0.0720. This illustrate our methods can be applied to learning methods with complex structures.

We further demonstrate the scalability of our method to high dimensional systems, possibly with a large number of initial sample points. We apply our adaptive sampling algorithm to learn the quasi-potential of a discretized partial differential equation, which corresponds to a 50 dimensional dynamical system. This is a standard numerical example for data-driven quasipotential computation methods in (Lin et al. (2022a)). After discretization of the Ginzburg-Landau equation $u_t = \delta u_{zz} - \delta^{-1} V'(u)$, we have a 50 dimensional system of ODEs:

$$\frac{du_i}{dt} = \delta \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} - \delta^{-1} V'(u_i), \quad 1 \le i \le 50, \tag{24}$$

where the double-well potential $V = \frac{1}{4}(1-u^2)^2$, and δ is taken as 0.1. The state of the system is denoted by $\mathbf{u} = (u_1, \ldots, u_{I-1})$. The numerical details of discretization and initial state generation can be founded in Appendix D. The task here is to model the dynamics around two stable points $\pm\mathbf{1}$ and the target measure is set as the uniform distribution of $S = \{\mathbf{u} \in \mathbb{R}^{50} | \ |\mathbf{u} \pm \mathbf{1}| < 1\}$. Here $N(= 100)$ initial states are generated and Alg. 1 is applied here to find the length of trajectory for each initial state independently. The baseline sampling is directly collecting 100 trajectories of lengths $T = 200$. The baseline target risk is $7.642 \times 10^{-2} \pm 9.302 \times 10^{-3}$. To achieve this target risk using Alg. 1, we require a sampling budget of 14735 ± 1590, which corresponds to a 26.3% reduction in the number of required state samples. This demonstrates that our sampling algorithms can be scaled to handle high dimensional systems with a large volume of trajectory data.

4.4. Computation cost of adaptive samplers

The computation cost of Wasserstein distance calculation by *Sinkhorn* is $O(N^3T^3)$ with respect to the number of initial points N and the (maximum) length of trajectories T. However N can be selected in advance and then fixed in our algorithms, which means the parameter N is controllable. For $N \gg 1$, we can randomly partition the dataset of initial points into small subsets with size $\ll N$ then apply our algorithms for each subsets independently, such as the examples of quasipotential computation in section 4.3.2. For $T \gg 1$, we can increase the time step to reduce the size of samples used in Wasserstein distance calculation. With the sub-sampling method, our algorithms can scale to handle high dimensional problems with a very large number of initial points and long trajectories.

5. Conclusion

In this paper, we propose adaptive sampling algorithms for learning dynamics. This work is motivated by the gap between model evaluation of generalization performance and sampling strategy. After formulating the learning target risk in an integral expression according to our learning goals, we find the model performance can be bounded by the 1-Wasserstein distance between the sampling empirical measure and the target measure. Based on this finding, we propose several adaptive algorithms with regard to the accessibility of the target measure and trajectory-wise adaptivity. Through numerical experiments, we show our algorithms have robustness to different dynamics with different learning goals, is more effective compared to baseline passive sampling methods, and can be readily applied in conjunction with many existing learning algorithms for data-driven dynamical

systems modelling. In future work, we plan to study adaptive sampling algorithms for such dynamical systems. Another direction of future investigation is establishing the theoretical guarantee of the proposed sampling methods.

Acknowledgments

QL is supported by the National Research Foundation, Singapore, under the NRF fellowship (NRF-NRFF13-2021-0005).

References

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

Xiaoli Chen, Liu Yang, Jinqiao Duan, and George Em Karniadakis. Solving inverse stochastic problems from discrete particle observations using the fokker–planck equation and physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(3):B811–B830, 2021.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *J. Mach. Learn. Res.*, 22(78):1–8, 2021.

Dylan Foster, Tuhin Sarkar, and Alexander Rakhlin. Learning nonlinear dynamical systems from a single trajectory. In *Learning for Dynamics and Control*. PMLR, 2020.

Boumediene Hamzi and Houman Owhadi. Learning dynamical systems from data: a simple cross-validation perspective, part i: parametric kernel flows. *Physica D: Nonlinear Phenomena*, 421: 132817, 2021.

Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., 2001.

Yuehaw Khoo, Jianfeng Lu, and Lexing Ying. Solving for high-dimensional committor functions using artificial neural networks. *Research in the Mathematical Sciences*, 6(1):1–13, 2019.

Qianxiao Li, Bo Lin, and Weiqing Ren. Computing committor functions for the study of rare events using deep learning. *The Journal of Chemical Physics*, 151(5):054112, 2019.

Bo Lin, Qianxiao Li, and Weiqing Ren. A data driven method for computing quasipotentials. In *Mathematical and Scientific Machine Learning*. PMLR, 2022a.

Bo Lin, Qianxiao Li, and Weiqing Ren. Computing the invariant distribution of randomly perturbed dynamical systems using deep learning. *Journal of Scientific Computing*, 91(3):1–17, 2022b.

Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. In *International Conference on Machine Learning*, pages 3208–3216. PMLR, 2018.

Horia Mania, Michael I Jordan, and Benjamin Recht. Active learning for nonlinear system identification with guarantees. *arXiv:2006.10277*, 2020.

Frank Noé and Feliks Nuske. A variational approach to modeling slow processes in stochastic dynamical systems. *Multiscale Modeling & Simulation*, 11(2):635–655, 2013.

Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, 19(1):932–955, 2018.

Mark E Tuckerman. Ab initio molecular dynamics: basic concepts, current trends and novel applications. *Journal of Physics: Condensed Matter*, 14(50):R1297, 2002.

Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

Hao Wu and Frank Noé. Variational approach for learning markov processes from time series data. *Journal of Nonlinear Science*, 30(1):23–66, 2020.

Peijie Zhou and Tiejun Li. Construction of the landscape for multi-stable systems: Potential landscape, quasi-potential, a-type integral and beyond. *The Journal of chemical physics*, 144(9): 094109, 2016.

Appendix A. Proof of multi-step risk bound (11).

Proof With $\rho \geq 1$ and $\mathrm{Lip}(\hat{f}) < \hat{K}$, we have

$$\begin{aligned}
L_*^T(\theta) &= L_*^{T-1}(\theta) + \int d\left(f_*^T(x), \hat{f}^T(x;\theta)\right)\ \mathrm{d}\hat{\mu}_* \\
&\leq L_*^{T-1}(\theta) + \rho \int \left[d\left(f_*^{T-1}\circ f_*, \hat{f}^{T-1}\circ f_*\right) + d\left(\hat{f}^{T-1}\circ f_*, \hat{f}^{T-1}\circ \hat{f}\right)\right]\ \mathrm{d}\hat{\mu}_* \\
&\leq L_*^{T-1}(\theta) + \rho \int d\left(f_*^{T-1}, \hat{f}^{T-1}\right)\ \mathrm{d}(f_*\#\hat{\mu}_*) + \rho\hat{K}^{T-1}L_*(\theta) \\
&\leq \int d\left(f_*^{T-1}, \hat{f}^{T-1}\right)\ \mathrm{d}(\rho f_*\#\mu_* + \mu_*) + \rho\hat{K}^{T-1}L_*(\theta) \\
&\cdots \\
&\leq \int d(f_*, f)\ \mathrm{d}\left(\sum_{l=0}^{T-1}\hat{K}^{T-l-1}\mathcal{F}^l\#\mu_*\right),
\end{aligned} \tag{25}$$

where the operator $\mathcal{F}$ is defined as $\mathcal{F}[\nu] = (f_* + \mathbb{I})\#\nu$ for any measure ν with bounded support in $\mathbb{R}^m$.

■

Appendix B. Numerical example for remark 5.

Here we give a numerical illustration on learning the *Lorenz 63* system with SINDy. We compare the one-step sampling strategy directly from a accessible μ_* (i.e. $T = 1$) with multi-step strategies. Here, we take $\mu_* = \mathbb{1}_{\mathcal{S}}/|\mathcal{S}|$, where $\mathcal{S} = [-20, 20]^2 \times [10, 40]$, and we fix $N = 50$. In Fig.7, we plot the target risks of models learned from sample trajectories of varying lengths T. We can see that multi-step sampling strategies perform much better than one-step sampling.

Figure 7: SINDy on *Lorenz 63* system: target risk of models learned from trajectories with different lengths T and fixed $N(= 50)$.

Appendix C. Proof of Proposition 4.

Proof For $\theta_* \in \underset{\theta\in\Theta}{\operatorname{argmin}}\, L_*(\theta)$ and $\hat{\theta} \in \underset{\theta\in\Theta}{\operatorname{argmin}}\, L_{N,T}(\theta)$,

$$L_*(\hat{\theta}) - L_*(\theta_*) = \int d\left(f_*(x), \hat{f}(x;\hat{\theta})\right) d\mu_* - \int d\left(f_*(x), \hat{f}(x;\theta_*)\right) d\mu_* \tag{26}$$

$$= \int d\left(f_*(x), \hat{f}(x;\hat{\theta})\right) d\mu_* - \int d\left(f_*(x), \hat{f}(x;\hat{\theta})\right) d\mu_{N,T} \tag{27}$$

$$+ \int d\left(f_*(x), \hat{f}(x;\hat{\theta})\right) d\mu_{N,T} - \int d\left(f_*(x), \hat{f}(x;\theta_*)\right) d\mu_{N,T} \tag{28}$$

$$+ \int d\left(f_*(x), \hat{f}(x;\theta_*)\right) d\mu_{N,T} - \int d\left(f_*(x), \hat{f}(x;\theta_*)\right) d\mu_*. \tag{29}$$

The approximation error is $L_*(\theta_*)$, generalization error (caused by distribution shift) is the term (27) and (29), optimization error is the term (28).

Since $\hat{\theta} \in \underset{\theta\in\Theta}{\operatorname{argmin}}\, L_{N,T}(\theta)$, (28) ≤ 0. The model bias $L_*(\theta_*) = 0$ since $\mathcal{H}$ is a universal approximator. By the dual representation of 1-Wasserstein distance, (27)(29) can be bounded by the 1-Wasserstein distance $W_1(\mu_*, \mu_{N,T})$. We have

$$L_*(\hat{\theta}) \leq 2C_* W_1(\mu_*, \mu_{N,T}) \tag{30}$$

where $C_* = \sup_{\theta\in\Theta} \left\{C |\ C = \text{Lip}\left(d(f_*(\cdot), \hat{f}(\cdot\,;\theta)\right)\right\}$. ■

Appendix D. Numerical details of the discretized PDE in 4.3.

Consider the Ginzburg-Landau equation

$$u_t = \delta u_{zz} - \delta^{-1} V'(u), \quad z \in [0,1] \tag{31}$$

with the boundary conditions $u(0,t) = u(1,t) = 0$ and the initial condition $u(z,0) = u^0(z)$, where $V(u) = \frac{1}{4}(1- u^2)^2$ is a double-well potential and δ is set as 0.1. We partition the interval $[0,1]$ using $I+1$ grid points $z_0, \ldots, z_I$, where $z_i = ih$ and $h = 1/I$. Then, we approximate the spatial derivatives in (31) using the central finite difference and obtain the following system of ODEs:

$$\frac{du_i}{dt} = \delta \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} - \delta^{-1} V'(u_i), \quad 1 \le i \le I-1 \tag{32}$$

with $u_0 = u_I = 0$ and the initial condition $u_i(0) = u^0(z_i)$ for $1 \le i \le I-1$, where u_i denotes the approximate solution at the grid point z_i. The state of the system is denoted by $\mathbf{u} = (u_1, \ldots, u_{I-1})$. The number of discretization points is taken as $I = 51$. The initial states are generated from $u^0(z) = \dfrac{a \cdot \tilde{u}(z)}{\max_y |\tilde{u}(y)|}$ where $\tilde{u}(z) = \sum_{k=1}^4 \hat{u}_k \sin(k\pi z)$ and $\{\hat{u}_k\}_{k=1}^4, a$ are drawn from the uniform distributions: $u_k \sim \mathcal{U}(-1,1), a \sim \mathcal{U}\left(0, \frac{3}{2}\right)$.

Appendix E. Adaptive algorithm 3 and 4.

Algorithm 3 Adaptive Sampler (μ_* accessible, adaptive trajectory-wise)

0. given: initial samples $\mathcal{D}_0 = \{x_j^i : i = 1, \dots, N, j = 0, \dots, T_0\}$ with N starting points $\{x_i\}_{i=1}^N$ and trajectory length T_0, simulator of the dynamics, target distribution μ_* sampler, stopping-criterion parameter $W > 0$, sampling step $\Delta T \in \mathbb{N}^+$, index set $\mathcal{M} = [N]$, overlapping tolerance ϵ_2.

1. set: sampling dataset $\mathcal{D} \leftarrow \mathcal{D}_0$, empirical measure $\hat{\mu} \leftarrow \dfrac{1}{|\mathcal{D}|} \displaystyle\sum_{x \in \mathcal{D}} \delta_x$, $T \leftarrow T_0$;

2. **do**

sample $x_{T+1}^i, \cdots, x_{T+\Delta T}^i$ from x_T^i by the simulator for $i \in \mathcal{M}$;

overlapping check:

for $i \in \mathcal{M}$ **do**

for $j < i$ **do**

for $T_i \in [0, \hat{T}]$ **do**

if $|x_i^{T_i} - x_j^{T_j}| \le \epsilon$ *for some* T_j **then**

delete i from $\mathcal{M}$, and label i^{th} point T_i.

end

end

end

end

while *stopping criterion not met*;

update $\mathcal{D} \leftarrow \mathcal{D} \cup \{x_{T+1}^i, \cdots, x_{T+\Delta T}^i : i \in \mathcal{M}\} \cup \{x_{T+1}^i, \cdots, x_{T_i}^i : i \in \mathcal{M}^c\}$;

update $\hat{\mu} \leftarrow \dfrac{1}{|\mathcal{D}|} \displaystyle\sum_{x \in \mathcal{D}} \delta_x$, $T \leftarrow T + \Delta T$;

3. output: sampling strategy T_i for i^{th} point, $i \in [N]$.

Algorithm 4 Adaptive Sampler (μ_* inaccessible, adaptive trajectory-wise)

0. given: initial samples $\mathcal{D}_0 = \{x_j^i : i = i \in \mathcal{M}, j = 0, \dots, T_0\}$ with N starting points $\{x_i\}_{i=1}^N$ and trajectory length T_0, simulator of the dynamics, stopping-criterion parameter $W > 0$, sampling step $\Delta T \in \mathbb{N}^+$;

1. set: sampling dataset $\mathcal{D} \leftarrow \mathcal{D}_0$, empirical measure $\hat{\mu} \leftarrow \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \delta_x$, $T \leftarrow T_0$;

do

2. solve empirical risk minimization over $\hat{\mu}$: $\hat{\theta} \leftarrow \underset{\theta \in \Theta}{\operatorname{argmin}} L_{N,T}(\theta)$;

3. generate μ_* sampler from $\hat{f}(x; \hat{\theta})$;

4. solve the optimization problem 17 by Alg 1: $\hat{T} \leftarrow \underset{T' \in [T + \Delta T, \infty)}{\operatorname{argmin}} W_1(\mu_*, \hat{\mu}) + \phi(N, T')$;

5. sample $x_{T+1}^i, \cdots, x_{\hat{T}}^i$ from x_T^i by the simulator for $i = i \in \mathcal{M}$;

8. overlapping check:

for $i \in \mathcal{M}$ **do**

for $j < i$ **do**

for $T_i \in [0, \hat{T}]$ **do**

if $|x_i^{T_i} - x_j^{T_j}| \leq \epsilon$ *for some* T_j **then**

delete i from $\mathcal{M}$, and label i^{th} point T_i.

end

end

end

end

6. update $\mathcal{D} \leftarrow \mathcal{D} \cup \{x_{T+1}^i, \cdots, x_{\hat{T}}^i : i = i \in \mathcal{M}\} \cup \{x_{T+1}^i, \cdots, x_{T_i}^i : i \in \mathcal{M}^c\}$;

7. update $\hat{\mu} \leftarrow \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \delta_x$, $T \leftarrow \hat{T}$;

while *stopping criterion not met*;

output: sampling strategy T_i for i^{th} point, $i \in [N]$.