Data Assimilation in Chaotic Systems Using Deep Reinforcement Learning

Mohamad Abed El Rahman Hammoud[1], **Naila Raboudi**[1], **Edriss S. Titi**[2,3],
Omar Knio[1] and **Ibrahim Hoteit**[1]

[1]King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia
[2]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge CB3 0WA, UK
[3]Department of Mathematics, Texas A & M University, College Station, TX 77843, USA

Key Points:

- Deep reinforcement learning (RL) is introduced for data assimilation
- RL generalizes to new situations unseen during training through actively learning from the data and system dynamics
- The RL agent allows for nonlinear state-adaptive correction of the forecast using the observations
- The performance of the proposed RL algorithm surpasses that of the ensemble Kalman filter (EnKF) with the Lorenz '63

Corresponding author: Ibrahim Hoteit, `ibrahim.hoteit@kaust.edu.sa`

Abstract

Data assimilation (DA) plays a pivotal role in diverse applications, ranging from climate predictions and weather forecasts to trajectory planning for autonomous vehicles. A prime example is the widely used ensemble Kalman filter (EnKF), which relies on linear updates to minimize variance among the ensemble of forecast states. Recent advancements have seen the emergence of deep learning approaches in this domain, primarily within a supervised learning framework. However, the adaptability of such models to untrained scenarios remains a challenge. In this study, we introduce a novel DA strategy that utilizes reinforcement learning (RL) to apply state corrections using full or partial observations of the state variables. Our investigation focuses on demonstrating this approach to the chaotic Lorenz '63 system, where the agent's objective is to minimize the root-mean-squared error between the observations and corresponding forecast states. Consequently, the agent develops a correction strategy, enhancing model forecasts based on available system state observations. Our strategy employs a stochastic action policy, enabling a Monte Carlo-based DA framework that relies on randomly sampling the policy to generate an ensemble of assimilated realizations. Results demonstrate that the developed RL algorithm performs favorably when compared to the EnKF. Additionally, we illustrate the agent's capability to assimilate non-Gaussian data, addressing a significant limitation of the EnKF.

Plain Language Summary

Reliable forecasts of the state of chaotic systems, such as environmental flows, require combining observational data and dynamical model outputs through a process called data assimilation. The ensemble Kalman filter (EnKF) is the most commonly adopted algorithm for this task, however, is subject to some limitations when applied to nonlinear/non-Gaussian systems. Recently, there has been interest in using deep learning (DL), particularly within a supervised learning setup, for DA. However, making DL models work well in new situations that differ from those experienced during training is challenging. In this work, we propose a new DA approach that leverages reinforcement learning (RL). RL helps the system make corrections to its predictions based on observed data, even if the model hasn't been trained for those specific scenarios. Compared to the state of the art DA algorithms, RL offers a novel framework for nonlinear corrections of the forecast using the incoming observations. Numerical results show that the proposed RL algorithm outperforms the EnKF and demonstrates the RL agent's ability at addressing some shortcomings of the EnKF.

1 Introduction

Assimilating observational data is essential for improving predictability and understanding complex dynamics in chaotic and dynamic physical systems. Chaotic dynamical systems, such as those describing climate and weather, involve inherent imperfections and extreme sensitivity to initial conditions, whereas the observational data available for such systems often carry significant uncertainties (Eckmann & Ruelle, 1985). To address the associated challenges, data assimilation (DA) combines real-world observations with numerical model outputs, continually refining model predictions by aligning them with newly acquired observations to enhance the accuracy and reliability of the predictions (Ott et al., 2004). DA techniques are broadly categorized as variational and filtering methods (Le Dimet & Talagrand, 1986; Ghil & Malanotte-Rizzoli, 1991; Lorenc, 2003; Hoteit et al., 2018). The ensemble Kalman filter (EnKF) represents one of the most popular filtering DA techniques, especially in the context of large-scale nonlinear systems (Evensen, 2003). Operating within a Bayesian probabilistic framework, the EnKF squentially splits the filtering (state estimation) process into cycles that alternate between forecast steps, driven by the system's dynamical model, and analysis steps, which updates the forecast with incoming data (Evensen, 2003). This approach enables Gaussian-based Monte Carlo (MC) approximations of both state forecast and analysis distributions through an ensemble of state samples (Hoteit et al., 2008).

EnKF schemes are considered as the gold standard when assimilating uncertain observations of the system states across diverse fields due to their robustness, capacity to handle complex and high-dimensional systems, and computational efficiency (Houtekamer & Mitchell, 1998). However, their applicability is not without constraints, particularly when the underlying assumptions are compromised. In particular, challenges may arise from the EnKF's inherent linear assumption, and the necessity for maintaining a Gaussian distribution within the ensemble, both of which become challenging in the presence of strong nonlinearities (Kalnay, 2002; Hoteit et al., 2008). Additionally, whereas the Gaussian assumption for both model and observational noise offers convenience, it may not universally hold in real-world scenarios, thereby limiting EnKF's performance, especially when errors deviate significantly from Gaussian patterns. In such cases, it is necessary to explore alternative approaches that are better suited for these scenarios; e.g. van Leeuwen (2009).

Reinforcement Learning (RL) is a paradigm of artificial intelligence that deals with how an agent can learn to make decisions through interactions with an environment, namely to achieve a specific objective (Recht, 2019). It is inspired by behavioral psychology and focuses on learning how to take actions in an environment to maximize some notion of cumulative reward. Within the RL framework, an agent engages in trial-and-error exploration, testing various actions and observing their outcomes (Mnih et al., 2015). The agent's goal is to formulate an optimal strategy, often referred to as a policy, that guides its actions to maximize the cumulative reward over a time horizon. It is noteworthy to point out that the RL framework is inherently different from the supervised learning approaches because the latter require a pre-computed reference database for training, which in this context consists in minimizing a global objective function (Glorot & Bengio, 2010; Karniadakis et al., 2021). RL finds extensive applications in domains necessitating dynamic control and decision-making capabilities, encompassing fields such as robotics (Kober et al., 2013), gaming (Mnih et al., 2013; Vinyals et al., 2019), autonomous navigation (Sallab et al., 2017), fluid dynamics (Novati et al., 2021; Bae & Koumoutsakos, 2022), and beyond.

In this work, we introduce a novel DA formalism utilizing RL as a strategy to actively update a nonlinear forecast correction scheme with the incoming data. The RL agent learns through interactions with the environment, adapting to its changes, and actively applies nonlinear corrections to handle complex processes. Numerical experiments were conducted with the Lorenz '63 chaotic system (Lorenz, 1963), and the RL agent's performance was benchmarked against the EnKF algorithm using a large cardinality ensemble under various experimental conditions. These include tracking a reference solution and assimilating normally-distributed noisy observations at various noise levels and observation frequencies. Furthermore, we investigate the performance of the RL agent at assimilating observations with different noise distribution models, namely uniform, log-normal and Gaussian noise. We further explore the RL agent's effectiveness at assimiliating partial state observations.

The remaining of the manuscript is organized as follows. Section 2 introduces the RL-DA framework. The RL methodology for DA is then described in Section 3, where a comprehensive overview of the RL framework is first introduced, accompanied by a description of the Lorenz '63 system and the EnKF algorithm. Sections 4 and 5 present our numerical results. Finally, Section 6 summarizes the main conclusions of this study.

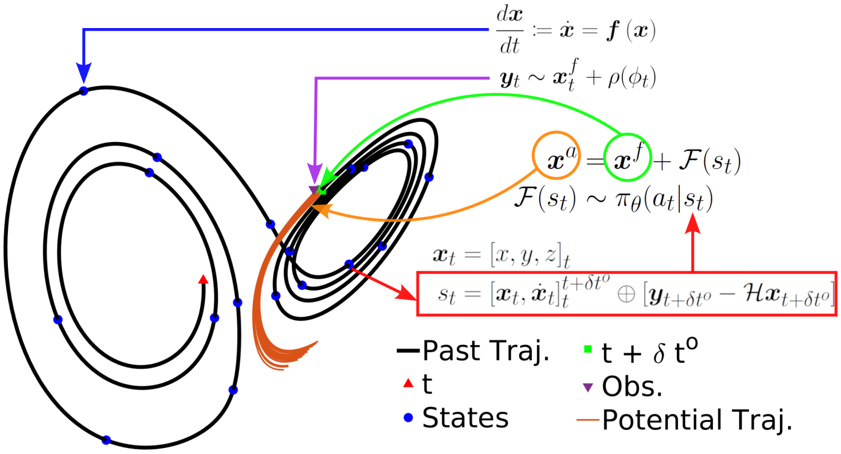

$\frac{d\boldsymbol{x}}{dt} := \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$
$\boldsymbol{y}_t \sim \boldsymbol{x}_t^f + \rho(\phi_t)$
$\boldsymbol{x}^a = \boldsymbol{x}^f + \mathcal{F}(s_t)$
$\mathcal{F}(s_t) \sim \pi_\theta(a_t|s_t)$
$\boldsymbol{x}_t = [x, y, z]_t$
$s_t = [\boldsymbol{x}_t, \dot{\boldsymbol{x}}_t]_t^{t+\delta t^o} \oplus [\boldsymbol{y}_{t+\delta t^o} - \mathcal{H}\boldsymbol{x}_{t+\delta t^o}]$
Past Traj.
t
States
t + δ t^o
Obs.
Potential Traj.

Figure 1. Schematic of the proposed reinforcement learning-based data assimilation framework using the Lorenz '63 as the main example. The plot illustrates the Lorenz '63 solution trajectory (black curve) with an arbitrary assimilation window start time t (red triangle) and corresponding end time $t + \delta t^o$ (green square) when a new observation is available and assimilated. The three dimensional state variables ($\boldsymbol{x}$) of the model are shown at every model time step δt (blue circles). At the last time step, the noisy observational data point ($\boldsymbol{y}$) is shown (inverted purple triangle) alongside the different evolution trajectories (orange curves) following several corrections ($\mathcal{F}(s_t)$) sampled from the policy function $\pi_\theta(a_t|s_t)$. The policy $\pi_\theta(a_t|s_t)$ considers as input state vector the extended state vector composed of the concatenation of the forecast state variables ($\boldsymbol{x}$) and their time derivatives ($\dot{\boldsymbol{x}}$) at each time step δt between t and $t + \delta t^o$ alongside the innovation term, defined as the difference between the observation and its correspondent forecast. The concatenation operation is denoted by $\oplus$, and for the sake of conciseness, concatenation of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ at each δt is represented by the sub- and super-scripts of $[\boldsymbol{x}, \dot{\boldsymbol{x}}]$. Since a stochastic policy is considered in the DA framework, an ensemble of $\mathcal{F}(s_t)$ correction terms are sampled from $\pi_\theta(a_t|s_t)$ when a noisy observation is available. Note that the state variables might not be fully observed, hence $\mathcal{H}$ projects the forecast onto the observation space. Moreover, the observation $\boldsymbol{y}$ is considered to be a noisy estimate of the forecast with no restriction on the distribution of the additive noise.

2 Reinforcement Learning For Data Assimilation

In RL, agents make sequential decisions to achieve specific goals, with the focus on maximizing cumulative rewards over time (Sutton & Barto, 2018; Bertsekas, 2019). This aligns with decision-making scenarios where actions have consequences, and objectives must be met. RL is particularly relevant to control systems (Azouani & Titi, 2014; Kalantarov & Titi, 2018), where agents learn control policies to influence the behavior of systems (Silver et al., 2014). The key concept in RL is the trade-off between exploration, where the agent experiments with new actions, and exploitation, where the agent chooses known actions with high rewards, mirroring real-world decision-making challenges (Sallab et al., 2017). RL agents learn from feedback, adapt to changing environments, and generalize knowledge to make decisions in new situations.

DA is an essential process used in scientific fields such as meteorology, oceanography, and environmental modeling to guide the state of complex systems with incoming observations (Ghil & Malanotte-Rizzoli, 1991; Hoteit et al., 2008). It involves merging observational data with numerical models to enhance predictions once observational information is available (Kalnay, 2002). This process continuously drives the computed system state to align with observations, thereby ensuring accurate and robust state estimates. DA accounts for model and observational uncertainties, offering more reliable predictions for chaotic systems, making it indispensable for tasks such as weather forecasting (Rabier, 2005) and climate modeling (Pedatella et al., 2014). Hence, adopting an RL framework for DA is a natural progression in the domain, enabling for a nonlinear correction scheme that is also free from restrictive assumptions on the statistics of the observations and model.

In RL, an agent exists in an environment that is described by a set of dynamical rules characterizing its evolution, for example, a system of differential equations (Sutton & Barto, 2018). The agent's responsibility is to make decisions affecting its environment in a way that it maximizes the cumulative reward, or achieves a particular goal. The ultimate outcome of the RL's training procedure is an agent policy $\pi_\theta(a_t|s_t)$, a mapping from the observation space to the action space, which is evaluated to actively control the behavior of the agent at state s_t in a dynamical system. The policy function is generally characterized by a neural network parameterized with θ. Policy functions can be categorized as either deterministic or stochastic; in a deterministic policy, the action with the highest probability is chosen, whereas a stochastic policy relies on random sampling to select an action. In the present framework, a stochastic policy was adopted from which the DA correction term was sampled, where actions are sampled from a Gaussian policy (Schulman et al., 2017). Hence, after training, a policy function is obtained that could be used to sample potential correction terms from a distribution that adapts to the agent's state, and allowing to generate an ensemble of states via MC sampling. In contrast with most efforts put for developing efficient DA schemes; eg. (Lermusiaux, 2007; Farchi et al., 2021; Buizza et al., 2022), the RL machinery relies on a nonlinear neural network to provide a correction without being restricted to a pre-computed dataset for supervising its training. Furthermore, the RL agent does not require any assumption on the noise distribution of the observational errors nor restrictive assumptions on the model.

In this study, the chaotic Lorenz '63 system of differential equations was considered to examine the performance of RL at DA for a chaotic dynamical system (Lorenz, 1963). The system describes the solution of a three-dimensional state vector, $\boldsymbol{x} = [x, y, z]$; it is characterized by a chaotic attractor, where the solution is sensitive to initial conditions and experiences a nonperiodic behavior (Eckmann & Ruelle, 1985; Bakarji et al., 2023). In this setting, the agent receives information, in the form of an extended state vector describing the system, denoted by states, that includes the forecast states and their derivatives $\boldsymbol{x}^f$ and $\dot{\boldsymbol{x}}^f$, respectively, at each model time step δt starting from the time t of the previous observation till the next observational time step $t + \delta t^o$, and the innovation term $\boldsymbol{y} - \mathcal{H}\boldsymbol{x}^f$. Here, $\mathcal{H}$ represents the observation operator that projects the model forecast $\boldsymbol{x}$ onto the observation space and $\boldsymbol{y}$ denotes a noisy observation of the system state.

The agent interacts with the environment to change its course of evolution and adapts to these changes to maximize the cumulative reward, as later defined, gathered over some period of time (Silver et al., 2014). This interaction is formulated mathematically as:

$$\boldsymbol{x}^a = \boldsymbol{x}^f + \mathcal{F}\left(s_t\right), \tag{1}$$

where the corrected state vector, denoted by a superscript a for analysis, $\boldsymbol{x}^a$ is the sum of the model forecast, $\boldsymbol{x}^f$, and the correction term $\mathcal{F}\left(s_t\right)$, which is sampled from $\pi_\theta(a_t|s_t)$. Note that this form of the update is similar to that of the Kalman Filter and the EnKF algorithm, however, the latter rely on a linear update term (Kalman, 1960). In the current configuration, the RL agent is not provided with statistical information regarding the noisy observations. Instead, it employs an MC strategy, using an RL agent that employs random stochastic policy sampling. This approach generates an ensemble of assimilated solutions, which are subsequently averaged to produce an improved estimate of the system state, denoted by RL-50 in the following sections.

The training cycle is defined by specifying the reward function (Lillicrap et al., 2015). We test out several reward functions in our preliminary investigation, where the agent's performance was evaluated using the mutual information, negative of the root-mean-squared error (RMSE) and RMSE^{-1}. While these reward functions are mathematically similar (Seidler, 1971; Guo et al., 2005), the associated training stability is different. Accordingly, the agent was trained to maximize the negative of the RMSE, which strikes a satisfactory balance between interpretability, computational cost and agent's performance.

3 Methods

3.1 Reinforcement Learning

The framework for RL involves training an agent through several interactions with the environment, in the present context, the dynamical system. Training an RL agent requires a large number of interactions with the environment and consequently a large unavoidable computational load often several orders of magnitude greater than solving the underlying differential equations. However, the field of RL has become more accessible in recent times, thanks to open-source libraries like `smarties` (Novati & Koumoutsakos, 2019) and `stable baselines3` (Raffin et al., 2021), among others. In this work, we leverage the capabilities of `stable baselines3`, a high-performance RL software designed to exploit parallel computing, distributing the training process across multiple computational nodes. In the present configuration, each node simulates a distinct trajectory of the Lorenz '63 system, providing a large set of agent-environment interactions that are used to train the agent. In this parallelized setup, each computational node accumulates experiences by independently interacting with various instances of the environment. These experiences are then structured into episodes defined as:

$$\tau = \{s_t, r_t, a_t, s_{t+1}\}_{0:T}, \tag{2}$$

where τ is the ordered set of interactions across a time horizon, t represents the time at which the environment is at state s_t, a_t is the action the agent takes at that time, r_t is the reward the agent receives from performing action a_t and s_{t+1} is the subsequent state.

The RL agent's training objective is to maximize the expected cumulative discounted reward function, defined as:

$$R_t = \sum_{t=0}^{T} \gamma^t r_t, \tag{3}$$

where $\gamma \in [0, 1)$ is the discount factor. In our specific setting, a smaller value of γ proves advantageous, given the random noise sampling. This choice of reducing the emphasis on distant future rewards results in a more stable agent performance.

The policy function π_θ is a mapping between the agent's state and the action space, which can be structured either as a set of discrete actions or as a probability distribution function for continuous actions. As previously mentioned, policy functions are either deterministic, the action to most likely result in the highest reward is chosen, or stochastic, where actions are randomly sampled from a distribution that is typically approximated by a surrogate model. Here, the policy π_θ is represented as a densely connected multi-layer perceptron (Chen & Chen, 1995) parameterized by θ. Furthermore, actions assume continuous values, leading π_θ to output a probability distribution over possible actions. Hence, the agent's actions can be either sampled from this distribution, allowing the agent to explore the environment and seek potentially rewarding outcomes, otherwise, the action with the highest probability can be chosen.

3.2 Proximal Policy Optimization

In the present framework, we adopt the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) and briefly describe it here for completeness. PPO trains an agent using two key components, each parameterized by distinct neural networks: an actor network that takes the environment's state as input and produces the corresponding action, and a critic network that also takes the environment's state as input and predicts the discounted reward (Mnih et al., 2016). In our study, both the actor and critic networks are represented by multi-layer perceptrons, each composed of two hidden layers, each containing 128 neurons.

The essence of the PPO algorithm revolves around optimizing the actor network to maximize the cumulative reward obtained by the agent, and the critic network to minimize the mean squared error between the predicted and actual expected cumulative rewards, starting from a given state. This optimization can be mathematically expressed through two distinct loss functions. The actor network is optimized by maximizing the actor's objective function:

$$J_{actor} = \mathbb{E}\left[\min\left(q_t\left(\theta\right)\hat{A}_t, \text{clip}\left(q_t\left(\theta\right), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right], \tag{4}$$

where $q_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{old}(a_t|s_t)$ is the ratio of the probability of adopting an action a_t at state s_t using π_θ to that of the previous policy π_{old}. Furthermore, the present setting relies on policy clipping with an $\epsilon = 0.2$ (Schulman et al., 2017), where $q_t(\theta) \in [1-\epsilon, 1+\epsilon]$. This policy clipping mechanism helps maintain policy stability during parameter updates, stabilizing the training process. On the other hand, the critic loss is given as:

$$L_{critic} = \mathbb{E}\left[\hat{A}^2\right], \tag{5}$$

where, $\mathbb{E}$ is the expectation operator and $\hat{A}$ is the advantage (Mnih et al., 2016), which quantifies how favorable the observed outcome of selecting a particular action is compared to the estimated discounted reward of the current state. The advantage is described as:

$$\hat{A} = V_{target} - V_{\theta,old}, \tag{6}$$

where, $V_{target} = \sum_{i=0}^{T-1} r_i \gamma^i + \gamma^T V_{\theta,old}(s_T)$ is the discounted reward computed using the agent's interactions with the environment and $V_{\theta,old}$ is the discounted reward predicted by the critic network.

3.3 Lorenz '63

The Lorenz '63 is a set of three deterministic ordinary nonlinear differential equations developed to simulate simplified atmospheric convection (Lorenz, 1963). This system is renowned for its manifestation of chaotic behavior, where even minuscule perturbations in initial conditions lead to substantially divergent solution trajectories over time (Eckmann & Ruelle, 1985). The Lorenz equations have been extensively studied in chaos theory and nonlinear dynamics, and have been the fundamental benchmark to develop new data assimilation techniques (Foias et al., 2001; Hayden et al., 2011). The Lorenz '63 equations are given by:

$$\dot{x} = \sigma(y - x), \tag{7}$$

$$\dot{y} = x(\rho - z) - y, \tag{8}$$

$$\dot{z} = xy - \beta z, \tag{9}$$

where, σ, ρ and β are typically positive constants. This system is known to exhibit a chaotic attractor for $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$. In this study, the system of equations were solved using an 2^{nd} order Runge-Kutta scheme with a time step $\delta t = 0.001$, which offers a suitable balance between solution accuracy and computational time for the application at hand.

3.4 Data assimilation using Reinforcement Learning

The present study explores a novel data assimilation framework that leverages RL to assimilate noisy observations of the system states and improve estimates of the system states. In this investigation, the environment is represented by the chaotic Lorenz '63 system (Lorenz, 1963). The RL agent receives noisy information about the system's state variables, and its policy, $\pi_\theta(a_t|s_t)$ that is contingent upon the environment's state s_t takes an action according to the preassigned strategy. The state upon which the agent's policy is evaluated consists of the extended vector composed by the concatenation $\left[\boldsymbol{x}^f, \dot{\boldsymbol{x}}\right]_t^f \oplus \left[\boldsymbol{x}^f, \dot{\boldsymbol{x}}\right]_{t+\delta t}^f \oplus \ldots \left[\boldsymbol{x}^f, \dot{\boldsymbol{x}}\right]_{t+\delta t^o}^f \oplus \left[\boldsymbol{y}_{t+\delta t^o} - \mathcal{H}\boldsymbol{x}_{t+\delta t^o}^f\right]$. Notably, this selection preserves the Markovian assumption inherent in the EnKF, as $\mathcal{F}\left(\boldsymbol{x}|_t^{t+\delta t^o}\right) = \mathcal{F}\left(\boldsymbol{x}(t+\delta t^o)\right)$. However, including forecast information from previous steps significantly enhances training stability, even though it comes at the cost of a higher dimensional input. This gives rise to the question of how long back-in-time should forecast states be considered.

In this context, we introduce an RL agent responsible for correcting model forecasts of the dynamical system states using the update equation:

$$\boldsymbol{x}_{t+\delta t^o}^a = \boldsymbol{x}_{t+\delta t^o}^f + \mathcal{F}_\theta\left(\boldsymbol{x}|_t^{t+\delta t^o}, \dot{\boldsymbol{x}}|_t^{t+\delta t^o}, \boldsymbol{y}_{t+\delta t^o} - \mathcal{H}\boldsymbol{x}_{t+\delta t^o}^f\right), \tag{10}$$

where, $\mathcal{F}_\theta$ represents the RL agent's policy, parameterized by θ. The policy takes as input the state vector $\boldsymbol{x}$ and the first-order derivatives $\dot{\boldsymbol{x}}$ at all time steps from t to $t + \delta t^o$ at δt increments, as well as the innovation term $\boldsymbol{y} - \mathcal{H}\boldsymbol{x}^f$. Since a stochastic policy function is considered, the study examines the performance of a single RL agent by taking maximum probability action, and the performance of an ensemble of agents by randomly sampling the policy function for actions.

3.5 Training the DA agent

The present experimental setup encompasses various hyper-parameters that require tuning to achieve satisfactory performance. The parameters subjected to tuning include the learning rate, γ, number of assimilation steps per episode ($n_{a,train}$), total number of episodes, value function coefficient (v_f), gradient clipping coefficient. Experiences have shown that the performance of a stable agent is most sensitive to γ, v_f and gradient clipping.

The process of hyper-parameter optimization commenced with a Latin hypercube sampling strategy to establish a baseline assessment of the acceptable range of values for these parameters. Subsequently, the training process is repeated using a new set of hyperparameters selected from within a finer-scale parameter space. For all experiments conducted, we employed the ADAM stochastic optimization algorithm (Kingma & Ba, 2017) to optimize the loss function for the parameters of the actor and critic networks. The parameters utilized for training the agents, which underpin the results presented in this study, are detailed in Supplementary Table 1.

The RL agent's training objective centered on maximizing the cumulative rewards accrued over a specific time horizon. At each assimilation step, the reward was calculated as the negative RMSE between the observation and the forecast generated by the preceding action. This choice was made because minimizing the RMSE is equivalent to maximizing the mutual information between the compared quantities and because the RMSE is ultimately the measure that is used to evaluate the performance of the agent. More specifically, since the experiments in this study feature a well-defined reference solution, we report the RMSE of both the RL and EnKF solutions with respect to the noise-free reference solution. The RMSE hence provides quantitative estimates that help examine the assimilated solution in terms of forecast and analysis.

3.6 Ensemble Kalman Filter

The EnKF algorithm is commonly employed to estimate a discrete-time state process, denoted as $\mathbf{x} = \{\mathbf{x}_n\}_{n\in\mathbb{N}}$, based on observations from a corresponding process $\mathbf{y} = \{\mathbf{y}_n\}_{n\in\mathbb{N}}$. These processes are conventionally connected through a state-space system described as follows:

$$\left\{\begin{array}{lcl} \mathbf{x}_t & = & \mathcal{M}(\mathbf{x}_{t-1}) + \mathbf{u}_t \\ \mathbf{y}_t & = & \mathcal{H}(\mathbf{x}_t) + \mathbf{v}_t, \end{array}\right. , \tag{11}$$

where $\mathcal{M}$ represents the nonlinear dynamical model, that advances the system state from time t to $t+\delta t$, and $\mathcal{H}_t$ the observation operator that projects $\mathbf{x}_t$ from the state space onto the observation space. Here, we make the simplifying assumption that $\mathcal{H}$ is linear, although EnKF algorithms can readily accommodate cases with nonlinear $\mathcal{H}$. The noise terms, $\mathbf{u} = \{\mathbf{u}_t\}_{t\in\mathbb{N}}$ and $\mathbf{v} = \{\mathbf{v}_t\}_{t\in\mathbb{N}}$ are respectively the model and observation process noises. The EnKF algorithm assumes $\mathbf{u}_t$ and $\mathbf{v}_t$ to follow Gaussian distributions with zero means and covariances $\mathbf{Q}_t$ and $\mathbf{R}_t$, respectively. Furthermore, $\mathbf{u}$ and $\mathbf{v}$ are assumed to be independent, jointly independent and independent of the initial state $\mathbf{x}_0$.

The filtering problem involves estimating the state, $\mathbf{x}_t$, based on observations up to time t. EnKF algorithms are primarily designed to provide a MC approximation of the system state distribution using an ensemble of system state realizations. From this ensemble, empirical estimates of the posterior mean state and associated error covariances are derived, typically in the form of sample means and covariances. The process starts with an analysis ensemble of size N_{ens} denoted as $\{\mathbf{x}_t^{a,i}\}_{i=1}^{N_{ens}}$ available at time t. Subsequently, the forecast ensemble at the next time step $t+\delta t$ is computed by advancing each member $\mathbf{x}_{t-1}^{a,i}$ forward in time using the dynamical model, described as:

$$\mathbf{x}^{f,i}_{t+\delta t} = \mathcal{M}(\mathbf{x}^{a,i}_t) + \eta^i, \tag{12}$$

where $\eta^i \sim \mathcal{N}(0, \mathbf{Q}_t)$. Upon receiving a new observation $\mathbf{y}_t$, each member of the forecast ensemble is adjusted using the Kalman gain $\mathbf{K}_t$ to generate the analysis ensemble $\{\mathbf{x}^{a,(i)}_t\}^{N_{ens}}_{i=1}$ according to:

$$\begin{aligned} \mathbf{x}^{a,i}_t &= \mathbf{x}^{f,i}_t + \mathbf{K}_t(\mathbf{y}^i_t - \mathcal{H}_t\mathbf{x}^{f,i}_t), & (13)\\ \mathbf{K}_t &= \mathbf{P}^f_t\mathcal{H}^T_t(\mathcal{H}_t\mathbf{P}^f_t\mathcal{H}^T_t + \mathbf{R}_t)^{-1}, & (14)\end{aligned}$$

where $\mathbf{P}^f_t$ denotes the sample forecast error covariance computed from the forecast members in (12) and $\mathbf{y}^i_n$ represents perturbed observations, i.e., $\mathbf{y}^i_t = \mathbf{y}_t + \mu^i_t$ with μ^i_t is a random noise sampled from the observational error distribution.

4 Tracking Reference Solutions

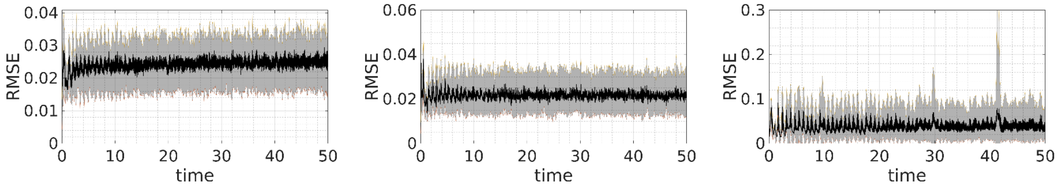
The RL-DA framework is systematically assessed under different experimental conditions. In the first scenario, an RL agent was trained to track a reference solution using coarse-in-time, noise-free observations of all state variables. Given the stochastic nature of the agent's policy function, the assimilated solution was not expected to precisely match the observations. Rather, the objective here was to investigate whether the corrections could maintain a reasonably close solution in comparison to the reference, and prevent them from diverging. Three training regimes were explored, involving observations every 5, 50, and 100 δt, corresponding to δt^o of 0.005, 0.05, and 0.1 time units, respectively. Evolution curves of the RMSEs of the RL solutions are presented in the top row of Figure 2. The average RMSE is represented by a solid black line, encircled by a shaded region denoting one standard deviation ($\pm\sigma$), based on 50 repetitions of the experiment involving different reference solutions. The plots indicate that the RMSE is on average approximately 0.025 for an assimilation frequency $\mathcal{T} = \delta t^o/\delta t$ values of 5 and 50, and increase to 0.05 for $\mathcal{T} = 100$. Furthermore, the top row of Figure 3 illustrates RL and reference solutions for the z-variable in the Lorenz '63 system, based on randomly selected reference trajectories. These curves highlight strong agreement between the RL solution and the reference, further corroborating the results presented in Figure 2.

5 Assimilating Noisy Observations

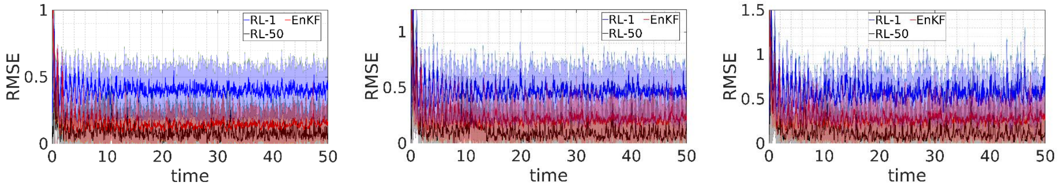
In a more realistic scenario, an ensemble of noisy observations are assimilated to improve the model forecast. This investigation explores the influence of noise levels (σ), $\mathcal{T}$, statistical noise distribution, and partial state observability on the RL agent's performance. Moreover, we conduct a comparative analysis by benchmarking the outcomes of the RL approach with those of the EnKF, which assimilates data from a relatively large ensemble comprising 50 realizations. To ensure robustness and statistical significance, each of the RL and EnKF experiments was repeated 50 times using different reference solutions, providing a statistically significant estimate of the RMSE.

5.1 Noise Level

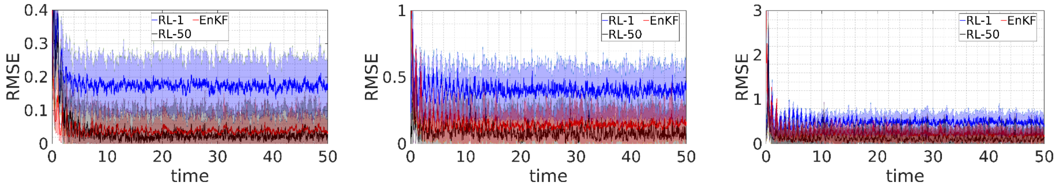
We examine the scenario of fully observed state available at regular intervals of $\mathcal{T} = 50$, with additive noise drawn from a Gaussian distribution characterized by zero mean and standard deviation σ. We investigate the influence of varying σ on the assimilated solution by computing the RMSE for the complete trajectory, encompassing both forecast and analysis phases. We compare the results obtained from a single RL agent, an average solution derived from 50 distinct RL trajectories with actions randomly sampled from the agent's policy, and the EnKF solution based on an ensemble comprising



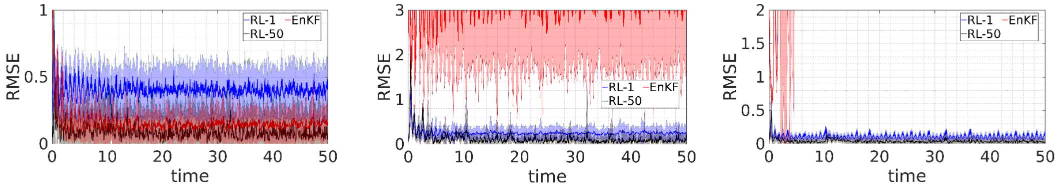
(a) (N/A, 5, $\mathrm{I}_{d,\,3x3}$) (b) (N/A, 50, $\mathrm{I}_{d,\,3x3}$) (c) (N/A, 100, $\mathrm{I}_{d,\,3x3}$)



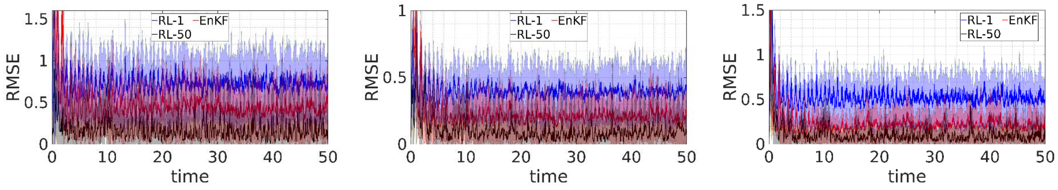
(d) ($\mathcal{N}(0,1)$, 50, $\mathrm{I}_{d,\,3x3}$) (e) ($\mathcal{N}(0,2)$, 50, $\mathrm{I}_{d,\,3x3}$) (f) ($\mathcal{N}(0,3)$, 50, $\mathrm{I}_{d,\,3x3}$)



(g) ($\mathcal{N}(0,1)$, 5, $\mathrm{I}_{d,\,3x3}$) (h) ($\mathcal{N}(0,1)$, 50, $\mathrm{I}_{d,\,3x3}$) (i) ($\mathcal{N}(0,1)$, 100, $\mathrm{I}_{d,\,3x3}$)



(j) ($\mathcal{N}(0,1)$, 50, $\mathrm{I}_{d,\,3x3}$) (k) ($\mathcal{L}(0,1)$, 50, $\mathrm{I}_{d,\,3x3}$) (l) ($\mathcal{U}(0,1)$, 50, $\mathrm{I}_{d,\,3x3}$)



(m) ($\mathcal{N}(0,1)$, 50, *diag*(1,0,0)) (n) ($\mathcal{N}(0,1)$, 50, *diag*(1,1,0)) (o) ($\mathcal{N}(0,1)$, 50, *diag*(1,0,1))

Figure 2. Evolution of the mean RMSE (solid lines) and its $\pm\sigma$ (shadowed) based on 50 experiment repetitions. Plotted are results for different experiments (a)-(c) tracking a noise-free reference solution, and for assimilating noisy observations in the case of (d)-(f) varying noise levels using normally-distributed noise, (e)-(i) different assimilation window lengths, (j)-(l) different noise distributions and (m)-(o) partial observability. The captions beneath each subplot describes the experimental condition in the order of noise distribution, $\delta t^o/\delta t$ the observation frequency and $\mathcal{H}$ the observation operator.

50 realizations. Note that this comparison places the RL agent at a slight disadvantage, as it was trained without any statistical information about the response of the system to observation noise. Nonetheless, we believe that the comparison with the EnKF prediction is meaningful as it represents the primary benchmark against which DA algorithms are evaluated, despite the more suitable comparison with the Kalman Filter. Notably, our algorithm consistently outperforms the Kalman Filter across all experiments and hence not shown.

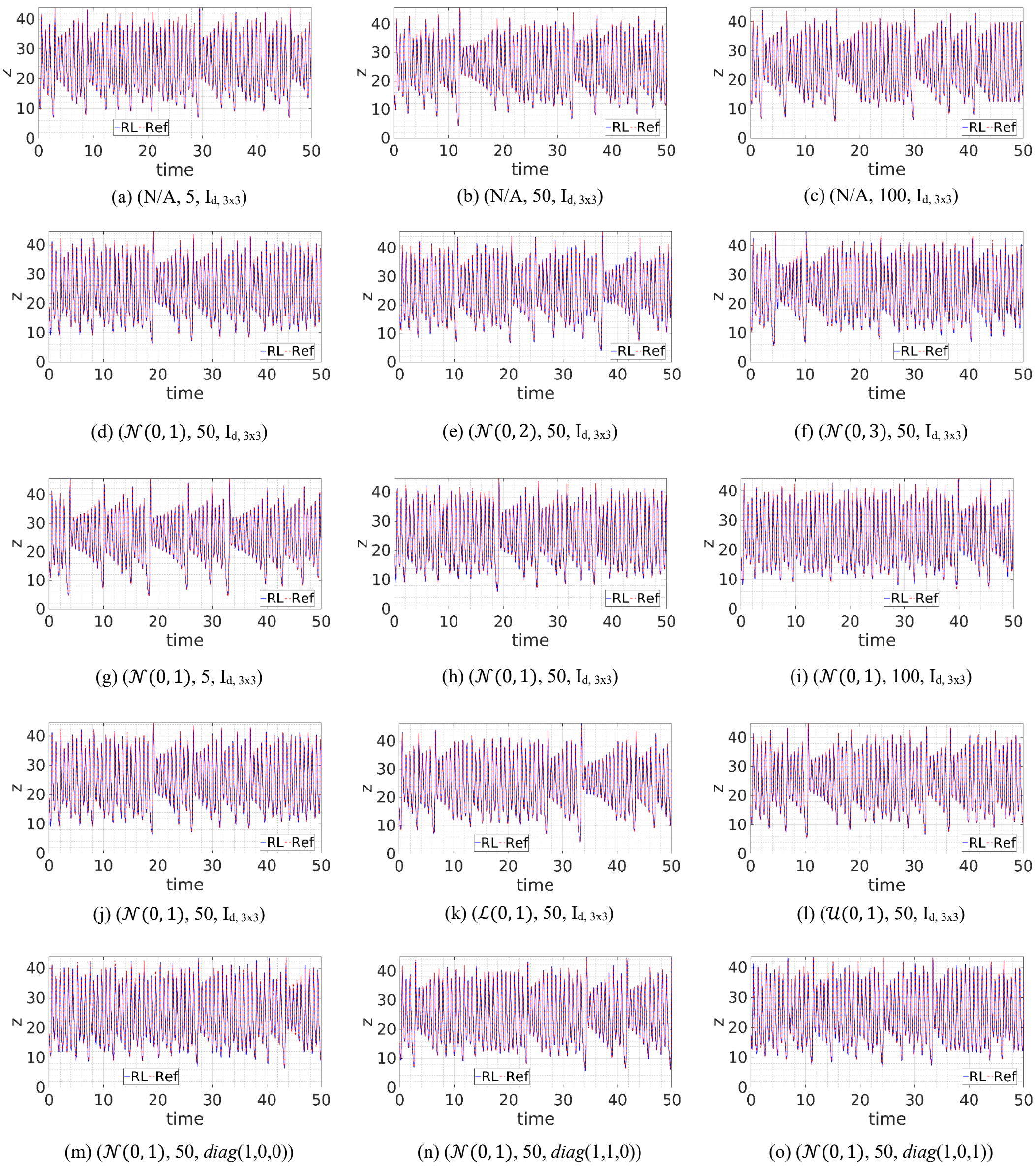
The second row of Figure 2 presents the RMSE evolution over time for the assimilated solution, resulting from RL and EnKF under different σ values. The plots suggest that, across all σ values considered, the EnKF solution exhibits slightly lower RMSE values than those of a single RL agent, and slightly larger RMSEs than the RL solution obtained by averaging 50 action realizations. This observation yields two significant insights: firstly, the potential computational efficiency gain from employing a single RL agent for DA, reducing computational overhead by a factor of at least N_{ens}, where N_{ens} represents the ensemble size. Secondly, using a single RL agent with a stochastic policy allows for sampling a diverse set of forecast corrections, yielding a new ensemble of state estimates that when averaged, generally results in a lower RMSE compared to an EnKF solution produced using an equivalent ensemble size.

Figure 4 illustrates the transition of the PDF after the correction is made alongside the distribution of the corrections for the RL and EnKF. The results indicate that the RL distribution is wider and covers more of the observations points than the EnKF, meaning that the RL ensemble is richer in terms of information it provides even though individual realizations perform poorer than the EnKF solution. On the other hand, the mean of the RL solutions is closer to the reference solution than the average EnKF solution, aligning well with the results obtained earlier. The plot also shows the distribution of the corrections, indicating that the distribution for the RL corrections is wider than that of the EnKF and suggesting that the EnKF is conservative when performing updates. Similar results for the remaining experiments are analyzed in the Supplementary.

As σ increases, noticeable high-amplitude, abrupt variations in RMSE are observed in the assimilated solutions, and the time-averaged RMSE increases. In the second row of Figure 3, we present the RL and reference evolution curves corresponding to the z-variable. The results demonstrate that the RL solution closely follows the reference solution for all σ values considered. However, as σ increases, slight deviations between the RL solution and the reference become evident, particularly at the peaks and troughs of the curves. Nevertheless, the RL agent successfully assimilates noisy data, at high noise levels.

5.2 Assimilation Frequency

Observational data may often become available at varying time frequencies, necessitating a DA scheme capable of accommodating different observation rates. In light of this requirement, we trained an RL agent to assimilate noisy data for distinct $\mathcal{T}$, thereby examining the influence of high-frequency ($\mathcal{T} = 5$), medium-frequency ($\mathcal{T} = 50$), and low-frequency ($\mathcal{T} = 100$) observations. The middle row of Figure 2 depicts the progression of RMSE under varying $\mathcal{T}$. Across all considered $\mathcal{T}$, the results suggest that a single RL agent exhibits slightly larger RMSE compared to those achieved by the 50-member EnKF solution. For all cases, the 50 RL agent-averaged solution demonstrates a lower time-averaged RMSE in contrast to the 50-member averaged EnKF solution. This indicates that even when the RL agents do not communicate among each other, an MC averaged solution achieves lower RMSEs than the EnKF solution with 50 members. Nevertheless, these results underscore the need to develop more sophisticated RL approaches,



(a) (N/A, 5, $\mathrm{I}_{\mathrm{d,\,3x3}}$)
(b) (N/A, 50, $\mathrm{I}_{\mathrm{d,\,3x3}}$)
(c) (N/A, 100, $\mathrm{I}_{\mathrm{d,\,3x3}}$)
(d) ($\mathcal{N}(0,1)$, 50, $\mathrm{I}_{\mathrm{d,\,3x3}}$)
(e) ($\mathcal{N}(0,2)$, 50, $\mathrm{I}_{\mathrm{d,\,3x3}}$)
(f) ($\mathcal{N}(0,3)$, 50, $\mathrm{I}_{\mathrm{d,\,3x3}}$)
(g) ($\mathcal{N}(0,1)$, 5, $\mathrm{I}_{\mathrm{d,\,3x3}}$)
(h) ($\mathcal{N}(0,1)$, 50, $\mathrm{I}_{\mathrm{d,\,3x3}}$)
(i) ($\mathcal{N}(0,1)$, 100, $\mathrm{I}_{\mathrm{d,\,3x3}}$)
(j) ($\mathcal{N}(0,1)$, 50, $\mathrm{I}_{\mathrm{d,\,3x3}}$)
(k) ($\mathcal{L}(0,1)$, 50, $\mathrm{I}_{\mathrm{d,\,3x3}}$)
(l) ($\mathcal{U}(0,1)$, 50, $\mathrm{I}_{\mathrm{d,\,3x3}}$)
(m) ($\mathcal{N}(0,1)$, 50, *diag*(1,0,0))
(n) ($\mathcal{N}(0,1)$, 50, *diag*(1,1,0))
(o) ($\mathcal{N}(0,1)$, 50, *diag*(1,0,1))

Figure 3. Evolution of the z-variable for a sample RL solution (solid blue lines) and corresponding reference (dashed red line). Plotted are results for different experiments (a)-(c) tracking a noise-free reference solution, and for assimilating noisy observations in the case of (d)-(f) varying noise levels using normally-distributed noise, (e)-(i) different assimilation window lengths, (j)-(l) different noise distributions and (m)-(o) partial observability. The captions beneath each subplot describes the experimental condition in the order of noise distribution, $\mathcal{T}$ the observation frequency and $\mathcal{H}$ the observation operator.

potentially utilizing multi-agent RL (Albrecht et al., 2023), that incorporate ensemble information when performing the correction step.

5.3 Noise Distribution

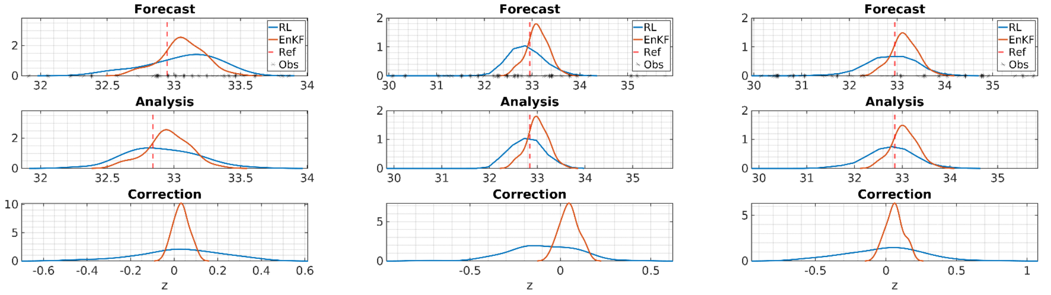
A major limitation of the EnKF is its reliance on normally-distributed observations of system states. We investigate the impact of different statistical distributions of observations on the DA performance of the RL agent. Specifically, we examine cases involving unbiased standard Gaussian, strongly positively biased standard log-normal, and weakly positively biased standard uniform observational noise. The 4^{th} row of Figure 2 presents the evolution curves of the RMSE for various observational noise distributions. The plots illustrate that for the case of standard Gaussian noise, both the single RL agent and EnKF solutions effectively assimilate noisy observational data with a slightly lower RMSE value achieved by the EnKF solution. On the other hand, the 50-realization averaged RL solution yields a lower RMSE compared to the 50-member EnKF solution. For log-normal and uniform noise distributions, the EnKF experiences large errors when assimilating noisy observations. Conversely, a single RL agent successfully assimilates these noisy observations, providing an assimilated solution that is close to the reference solution. Further improvements are observed when averaging the solutions obtained through policy sampling across 50 different realizations. The penultimate row of Figure 3 presents the RL and reference evolution curves for the z-variable. The plots indicate that the RL solution follows the reference solution reasonably well for all the noise distributions that were considered. The curves clearly illustrate that the RL agent is able to assimilate non-Gaussian noisy observations even when observations are perturbed with biased noise.

5.4 Partial Observability

The practicality of DA lies in its ability to assimilate observations that partially or even indirectly characterize the evolution of state variables within a dynamical system. This is particularly valuable when the full system state cannot be directly observed, such as in real-world climate and weather applications. To examine this setting, an RL agent was trained to assimilate noisy observations of select state variables–specifically, the x-variable alone, the x- and y-variables, and the x- and z-variables. The final row of Figure 2 portrays the evolution of RMSE of the aforementioned experiments. The curves demonstrate that, in all cases, the RL agent provides a suitable correction that adequately guides the evolution of the full state. It is noteworthy that the RMSE of the solution obtained using a single RL agent is comparable to, albeit slightly higher than that of the EnKF with an ensemble of 50 realizations. As observed in previous experiments, the averaged RL solution exhibits a lower average RMSE compared to the EnKF. To provide a tangible illustration of the assimilated solution's behavior, the final row of Figure 3 presents curves depicting the temporal evolution of the z-variable for the case with partial system states observability. These plots depict that the RL assimilated solution generally tracks the reference, with occasional discrepancies that typically occur at the peaks and troughs, as expected.

6 Discussion

This paper introduces RL as a novel approach for learning DA corrections. Through extensive experimentation on the Lorenz '63 dynamical system across various scenarios, we showcase the potential of the proposed approach. Our investigation encompasses both deterministic and stochastic settings, where RL agents are adeptly trained to track reference solutions and assimilate noisy data under varying conditions of assimilation window lengths, observational noise distributions, noise levels, and observed state variables.



(a) ($\mathcal{N}(0,1)$, 50, $\mathrm{I}_{\mathrm{d,\,3x3}}$) (b) ($\mathcal{N}(0,2)$, 50, $\mathrm{I}_{\mathrm{d,\,3x3}}$) (c) ($\mathcal{N}(0,3)$, 50, $\mathrm{I}_{\mathrm{d,\,3x3}}$)

Figure 4. PDFs of the z-variable before (top) and after (middle) the correction step at time $t = 45$ alongside the PDF of the correction (bottom) for the EnKF and RL solutions. The plots are presented for the experiment analyzing the sensitivity of the data assimilation algorithms to noise level.

The proposed RL-DA framework offers a paradigm shift by introducing new degrees of freedom to forecast-correction schemes, allowing for a nonlinear update term that satisfies a predefined optimal criteria, such as minimizing the root-mean-squared error in this study, hence, facilitating the discovery of novel correction strategies that are informed by the dynamical system through agent-environment interaction experiences. Furthermore, RL imparts robustness to correction strategies, rendering them stable even in the presence of noisy perturbations and compounding errors. In this work, the RL agent minimizes the ℓ_2 norm of the innovation term, a formulation demonstrated to be equivalent to maximizing the mutual information between observed state variables and their forecast counterparts. Notably, this framework eliminates the need for a reference database as opposed to supervised learning approaches, which are commonly established through the assimilation of noisy observational data using methods such as the EnKF or variational methods (Talagrand & Courtier, 1987).

However, incorporating RL into DA raises critical questions warranting further exploration. While we employed the negative of the ℓ_2 norm of the innovation term as the reward function in this study, more sophisticated functions considering system dynamics or ensemble information could potentially enhance the RL agent's performance. Moreover, since the RL agent is trained using the system of differential equations describing the evolution of a dynamical system, we speculate that this would force the agent to adapt and overcome model errors, when present. An overarching concern pertains to the physical validity of RL-derived solutions, which remains an open, fundamental question as is the case with other data-driven approaches when applied to physics-based applications. Although we did not directly encounter violations of physical constraints in our present setup, this avenue remains unexplored and in need for further exploration.

7 Open Research

All software and data used in the study will be made available upon acceptance at `https://github.com/mhammoud115/DA-RL`.

Acknowledgments

Research reported in this publication was supported by the Office of Sponsored Research (OSR) at King Abdullah University of Science and Technology (KAUST) CRG Award #CRG2020-4336 and Virtual Red Sea Initiative Award #REP/1/3268-01-01. The work of E.S.T. was supported in part by NPRP grant # S-0207-200290 from the Qatar National Research Fund (a member of Qatar Foundation).

References

Albrecht, S. V., Christianos, F., & Schäfer, L. (2023). *Multi-agent reinforcement learning: Foundations and modern approaches.* MIT Press. Retrieved from `https://www.marl-book.com`

Azouani, A., & Titi, E. S. (2014). Feedback control of nonlinear dissipative systems by finite determining parameters - a reaction-diffusion paradigm. *Evolution Equations and Control Theory*, *3*(4), 579-594. Retrieved from `https://www.aimsciences.org/article/id/363766be-87ba-4897-b3ba-4cd8d2c0cf49` doi: 10.3934/eect.2014.3.579

Bae, H. J., & Koumoutsakos, P. (2022, Mar 17). Scientific multi-agent reinforcement learning for wall-models of turbulent flows. *Nature Communications*, *13*(1), 1443. Retrieved from `https://doi.org/10.1038/s41467-022-28957-7` doi: 10.1038/s41467-022-28957-7

Bakarji, J., Champion, K., Nathan Kutz, J., & Brunton, S. L. (2023). Discovering governing equations from partial measurements with deep delay autoencoders. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *479*(2276), 20230422. Retrieved from `https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2023.0422` doi: 10.1098/rspa.2023.0422

Bertsekas, D. (2019). *Reinforcement learning and optimal control.* Athena Scientific.

Buizza, C., Quilodrán Casas, C., Nadler, P., Mack, J., Marrone, S., Titus, Z., ... Arcucci, R. (2022). Data learning: Integrating data assimilation and machine learning. *Journal of Computational Science*, *58*, 101525. Retrieved from `https://www.sciencedirect.com/science/article/pii/S1877750321001861` doi: https://doi.org/10.1016/j.jocs.2021.101525

Chen, T., & Chen, H. (1995). Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE transactions on neural networks*, *6*(4), 911–917.

Eckmann, J. P., & Ruelle, D. (1985, Jul). Ergodic theory of chaos and strange attractors. *Rev. Mod. Phys.*, *57*, 617–656. Retrieved from `https://link.aps.org/doi/10.1103/RevModPhys.57.617` doi: 10.1103/RevModPhys.57.617

Evensen, G. (2003, Nov 01). The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, *53*(4), 343-367. Retrieved from `https://doi.org/10.1007/s10236-003-0036-9` doi: 10.1007/s10236-003-0036-9

Farchi, A., Laloyaux, P., Bonavita, M., & Bocquet, M. (2021). Using machine learning to correct model error in data assimilation and forecast applications. *Quarterly Journal of the Royal Meteorological Society*, *147*(739), 3067-3084. Retrieved from `https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.4116` doi: https://doi.org/10.1002/qj.4116

Foias, C., Jolly, M. S., Kukavica, I., & Titi, E. S. (2001). The lorenz equation as a metaphor for the navier-stokes equations. *Discrete and Continuous Dynamical Systems*, *7*(2), 403-429. Retrieved from `https://www.aimsciences.org/article/id/69a3ea58-80e6-456f-974b-a5a90484700f` doi: 10.3934/dcds.2001.7.403

Ghil, M., & Malanotte-Rizzoli, P. (1991). Data assimilation in meteorology and oceanography. In R. Dmowska & B. Saltzman (Eds.), *Advances in geophysics* (Vol. 33, p. 141-266). Elsevier. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0065268708604422` doi: https://doi.org/10.1016/S0065-2687(08)60442-2

Glorot, X., & Bengio, Y. (2010, 13–15 May). Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh & M. Titterington (Eds.), *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (Vol. 9, pp. 249–256). Chia Laguna Resort, Sardinia, Italy: PMLR. Retrieved from `https://proceedings.mlr.press/v9/`

`glorot10a.html`

Guo, D., Shamai, S., & Verdu, S. (2005). Mutual information and minimum mean-square error in gaussian channels. *IEEE Transactions on Information Theory*, *51*(4), 1261-1282. doi: 10.1109/TIT.2005.844072

Hayden, K., Olson, E., & Titi, E. S. (2011). Discrete data assimilation in the lorenz and 2d navier–stokes equations. *Physica D: Nonlinear Phenomena*, *240*(18), 1416-1425. Retrieved from `https://www.sciencedirect.com/science/article/pii/S016727891100114X` doi: https://doi.org/10.1016/j.physd.2011.04.021

Hoteit, I., Luo, X., Bocquet, M., Kohl, A., & Ait-El-Fquih, B. (2018). Data assimilation in oceanography: Current status and new directions. *New frontiers in operational oceanography*, 465–512.

Hoteit, I., Pham, D.-T., Triantafyllou, G., & Korres, G. (2008). A new approximate solution of the optimal nonlinear filter for data assimilation in meteorology and oceanography. *Monthly Weather Review*, *136*(1), 317 - 334. Retrieved from `https://journals.ametsoc.org/view/journals/mwre/136/1/2007mwr1927.1.xml` doi: https://doi.org/10.1175/2007MWR1927.1

Houtekamer, P. L., & Mitchell, H. L. (1998). Data assimilation using an ensemble kalman filter technique. *Monthly Weather Review*, *126*(3), 796 - 811. Retrieved from `https://journals.ametsoc.org/view/journals/mwre/126/3/1520-0493_1998_126_0796_dauaek_2.0.co_2.xml` doi: https://doi.org/10.1175/1520-0493(1998)126⟨0796:DAUAEK⟩2.0.CO;2

Kalantarov, V. K., & Titi, E. S. (2018). Global stabilization of the navier-stokes-voight and the damped nonlinear wave equations by finite number of feedback controllers. *Discrete and Continuous Dynamical Systems - B*, *23*(3), 1325-1345. Retrieved from `https://www.aimsciences.org/article/id/b379f953-ac14-4309-8b5c-8d92d19c7b29` doi: 10.3934/dcdsb.2018153

Kalman, R. E. (1960, 03). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, *82*(1), 35-45. Retrieved from `https://doi.org/10.1115/1.3662552` doi: 10.1115/1.3662552

Kalnay, E. (2002). *Atmospheric modeling, data assimilation and predictability.* Cambridge University Press. doi: 10.1017/CBO9780511802270

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021, Jun 01). Physics-informed machine learning. *Nature Reviews Physics*, *3*(6), 422-440. Retrieved from `https://doi.org/10.1038/s42254-021-00314-5` doi: 10.1038/s42254-021-00314-5

Kingma, D. P., & Ba, J. (2017). *Adam: A method for stochastic optimization.*

Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, *32*(11), 1238-1274. Retrieved from `https://doi.org/10.1177/0278364913495721` doi: 10.1177/0278364913495721

Le Dimet, F.-X., & Talagrand, O. (1986). Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A: Dynamic Meteorology and Oceanography*, *38*(2), 97–110.

Lermusiaux, P. F. (2007). Adaptive modeling, adaptive data assimilation and adaptive sampling. *Physica D: Nonlinear Phenomena*, *230*(1), 172-196. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0167278907000589` (Data Assimilation) doi: https://doi.org/10.1016/j.physd.2007.02.014

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., … Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lorenc, A. C. (2003). The potential of the ensemble kalman filter for nwp—a comparison with 4d-var. *Quarterly Journal of the Royal Meteorological Society*, *129*(595), 3183-3203. Retrieved from `https://rmets.onlinelibrary.wiley`

`.com/doi/abs/10.1256/qj.02.132` doi: https://doi.org/10.1256/qj.02.132

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, *20*(2), 130 - 141. Retrieved from `https://journals.ametsoc.org/view/journals/atsc/20/2/1520-0469_1963_020_0130_dnf_2_0_co_2.xml` doi: https://doi.org/10.1175/1520-0469(1963)020⟨0130:DNF⟩2.0.CO;2

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... Kavukcuoglu, K. (2016, 20–22 Jun). Asynchronous methods for deep reinforcement learning. In M. F. Balcan & K. Q. Weinberger (Eds.), *Proceedings of the 33rd international conference on machine learning* (Vol. 48, pp. 1928–1937). New York, New York, USA: PMLR. Retrieved from `https://proceedings.mlr.press/v48/mniha16.html`

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... Hassabis, D. (2015, Feb 01). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529-533. Retrieved from `https://doi.org/10.1038/nature14236` doi: 10.1038/nature14236

Novati, G., de Laroussilhe, H. L., & Koumoutsakos, P. (2021, Jan 01). Automating turbulence modelling by multi-agent reinforcement learning. *Nature Machine Intelligence*, *3*(1), 87-96. Retrieved from `https://doi.org/10.1038/s42256-020-00272-0` doi: 10.1038/s42256-020-00272-0

Novati, G., & Koumoutsakos, P. (2019). Remember and forget for experience replay. In *Proceedings of the 36th international conference on machine learning* (pp. 1–10).

Ott, E., Hunt, B. R., Szunyogh, I., Zimin, A. V., Kostelich, E. J., Corazza, M., ... Yorke, J. A. (2004). A local ensemble kalman filter for atmospheric data assimilation. *Tellus A: Dynamic Meteorology and Oceanography*, *56*(5), 415-428. Retrieved from `https://doi.org/10.3402/tellusa.v56i5.14462` doi: 10.3402/tellusa.v56i5.14462

Pedatella, N. M., Raeder, K., Anderson, J. L., & Liu, H.-L. (2014). Ensemble data assimilation in the whole atmosphere community climate model. *Journal of Geophysical Research: Atmospheres*, *119*(16), 9793-9809. Retrieved from `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2014JD021776` doi: https://doi.org/10.1002/2014JD021776

Rabier, F. (2005). Overview of global data assimilation developments in numerical weather-prediction centres. *Quarterly Journal of the Royal Meteorological Society*, *131*(613), 3215-3233. Retrieved from `https://rmets.onlinelibrary.wiley.com/doi/abs/10.1256/qj.05.129` doi: https://doi.org/10.1256/qj.05.129

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-baselines3: Reliable reinforcement learning implementations. *The Journal of Machine Learning Research*, *22*(1), 12348–12355.

Recht, B. (2019). A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, *2*(1), 253-279. Retrieved from `https://doi.org/10.1146/annurev-control-053018-023825` doi: 10.1146/annurev-control-053018-023825

Sallab, A. E., Abdou, M., Perot, E., & Yogamani, S. (2017, jan). Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, *29*(19), 70–76. Retrieved from `https://doi.org/10.2352%2Fissn.2470-1173.2017.19.avm-023` doi: 10.2352/issn.2470-1173.2017.19.avm-023

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal policy optimization algorithms.*

Seidler, J. (1971). Bounds on the mean-square error and the quality of domain decisions based on mutual information. *IEEE Transactions on Information The-*

ory, *17*(6), 655-665. doi: 10.1109/TIT.1971.1054717

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014, 22–24 Jun). Deterministic policy gradient algorithms. In E. P. Xing & T. Jebara (Eds.), *Proceedings of the 31st international conference on machine learning* (Vol. 32, pp. 387–395). Bejing, China: PMLR. Retrieved from `https://proceedings.mlr.press/v32/silver14.html`

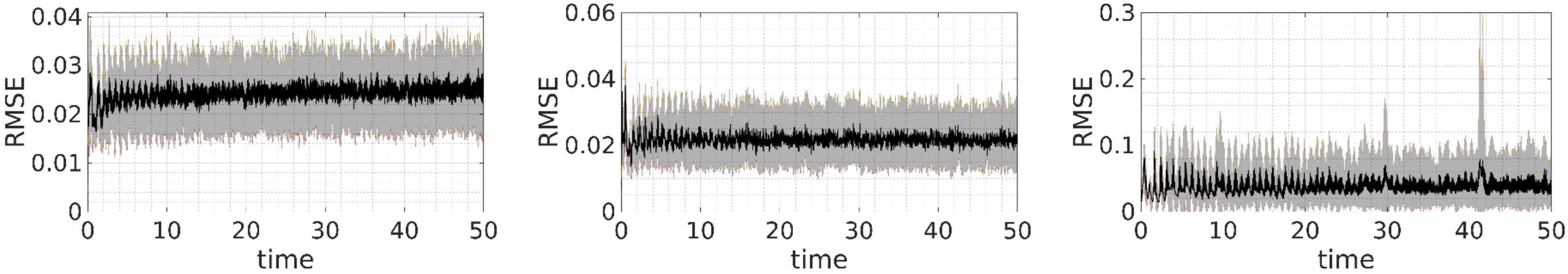
Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning, an introduction.* Bradford Books.

Talagrand, O., & Courtier, P. (1987). Variational assimilation of meteorological observations with the adjoint vorticity equation. i: Theory. *Quarterly Journal of the Royal Meteorological Society*, *113*(478), 1311-1328. Retrieved from `https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.49711347812` doi: https://doi.org/10.1002/qj.49711347812

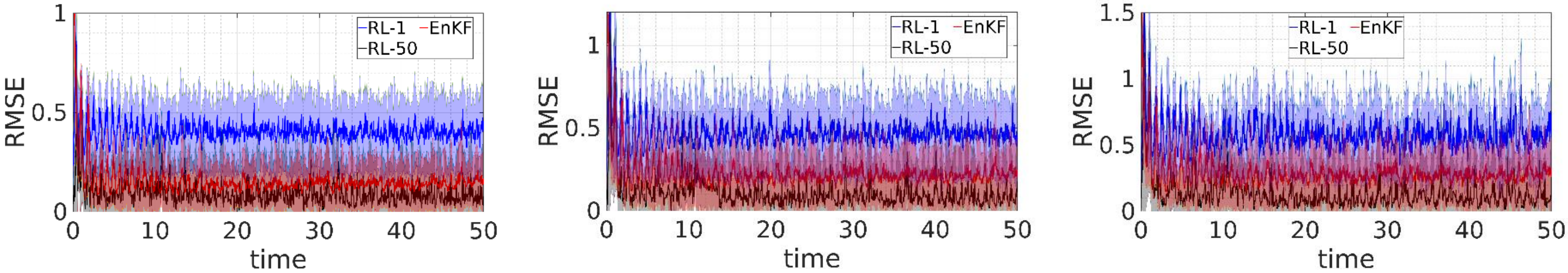
van Leeuwen, P. J. (2009). Particle filtering in geophysical systems. *Monthly Weather Review*, *137*(12), 4089 - 4114. Retrieved from `https://journals.ametsoc.org/view/journals/mwre/137/12/2009mwr2835.1.xml` doi: https://doi.org/10.1175/2009MWR2835.1

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., … Silver, D. (2019, Nov 01). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, *575*(7782), 350-354. Retrieved from `https://doi.org/10.1038/s41586-019-1724-z` doi: 10.1038/s41586-019-1724-z

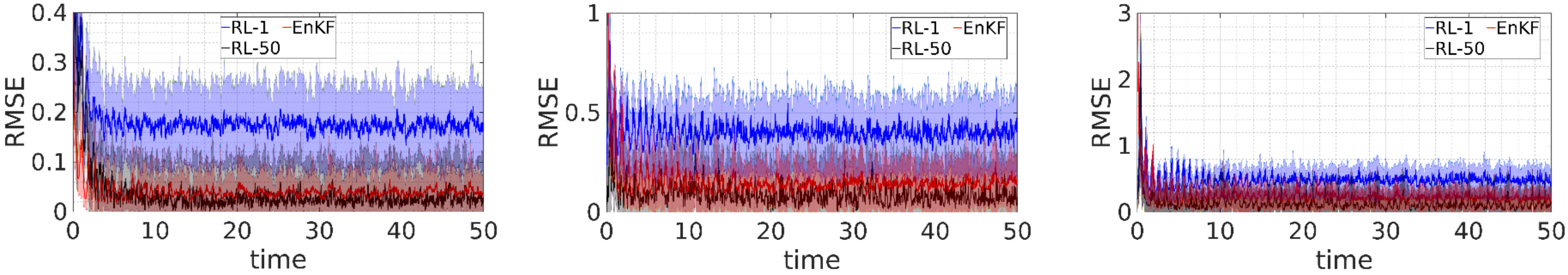
Figure.



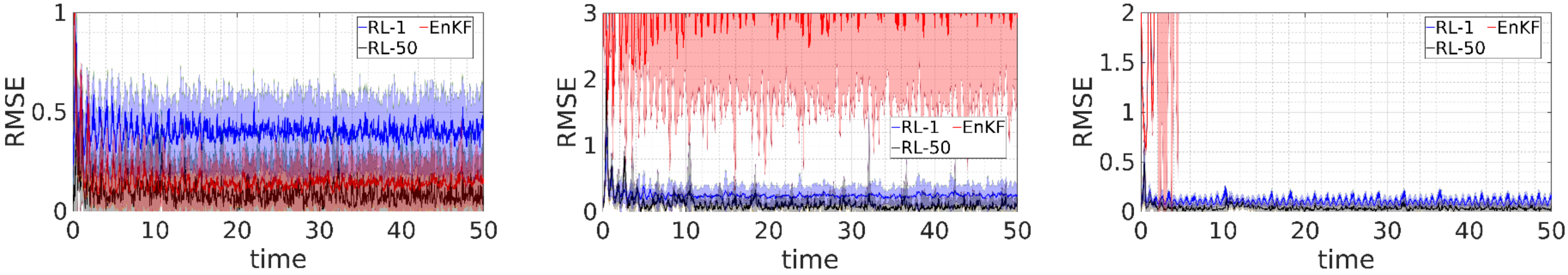
(a) (N/A, 5, $I_{d,\,3x3}$) (b) (N/A, 50, $I_{d,\,3x3}$) (c) (N/A, 100, $I_{d,\,3x3}$)



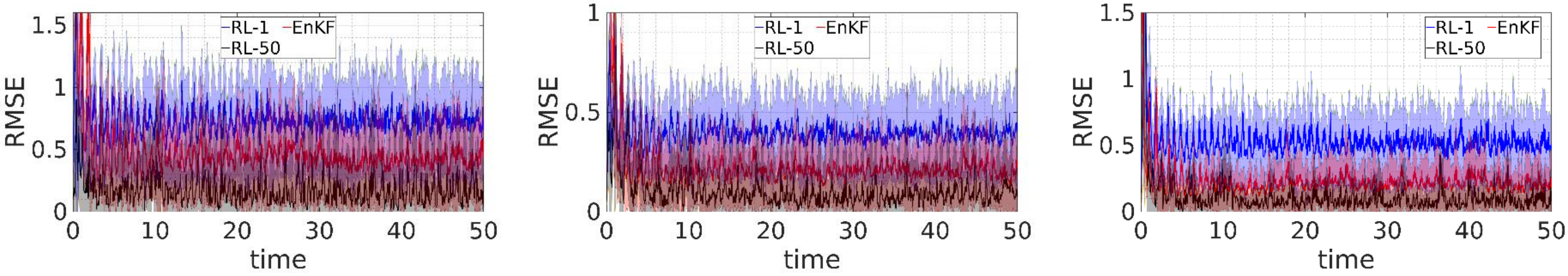
(d) ($\mathcal{N}(0,1)$, 50, $I_{d,\,3x3}$) (e) ($\mathcal{N}(0,2)$, 50, $I_{d,\,3x3}$) (f) ($\mathcal{N}(0,3)$, 50, $I_{d,\,3x3}$)



(g) ($\mathcal{N}(0,1)$, 5, $I_{d,\,3x3}$) (h) ($\mathcal{N}(0,1)$, 50, $I_{d,\,3x3}$) (i) ($\mathcal{N}(0,1)$, 100, $I_{d,\,3x3}$)

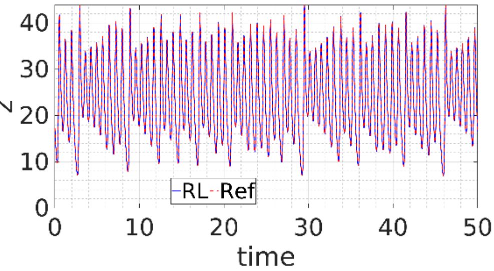


(j) ($\mathcal{N}(0,1)$, 50, $I_{d,\,3x3}$) (k) ($\mathcal{L}(0,1)$, 50, $I_{d,\,3x3}$) (l) ($\mathcal{U}(0,1)$, 50, $I_{d,\,3x3}$)

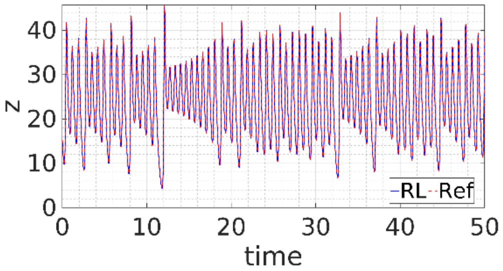


(m) ($\mathcal{N}(0,1)$, 50, *diag*(1,0,0)) (n) ($\mathcal{N}(0,1)$, 50, *diag*(1,1,0)) (o) ($\mathcal{N}(0,1)$, 50, *diag*(1,0,1))

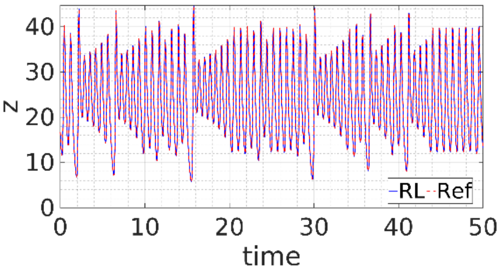
Figure.



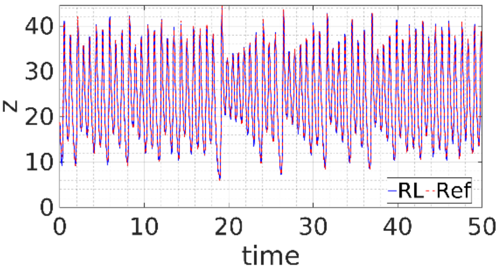
(a) (N/A, 5, $I_{d,\,3x3}$)



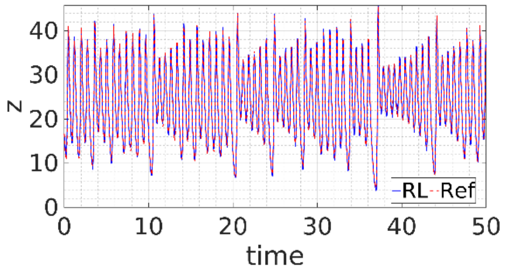
(b) (N/A, 50, $I_{d,\,3x3}$)



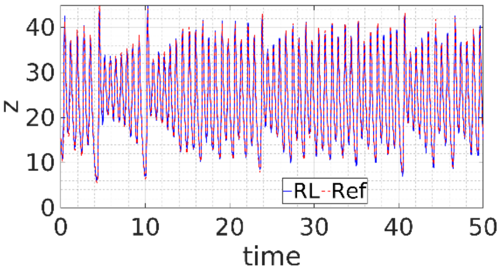
(c) (N/A, 100, $I_{d,\,3x3}$)



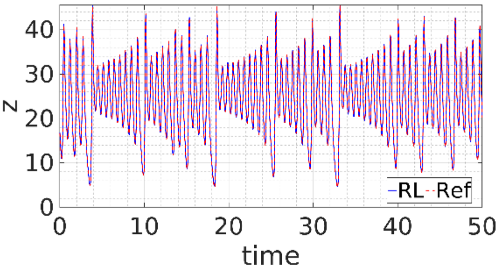
(d) ($\mathcal{N}(0,1)$, 50, $I_{d,\,3x3}$)



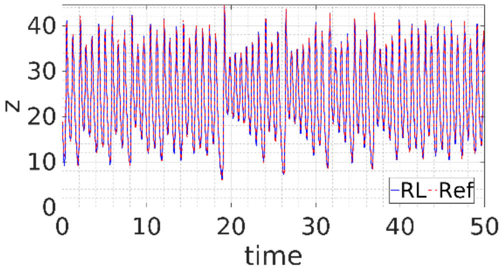
(e) ($\mathcal{N}(0,2)$, 50, $I_{d,\,3x3}$)



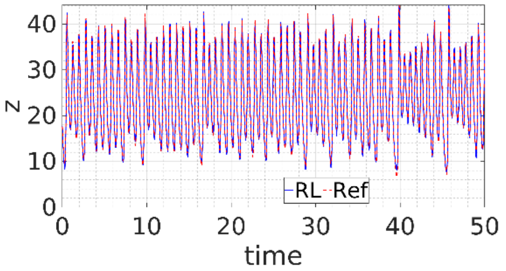
(f) ($\mathcal{N}(0,3)$, 50, $I_{d,\,3x3}$)



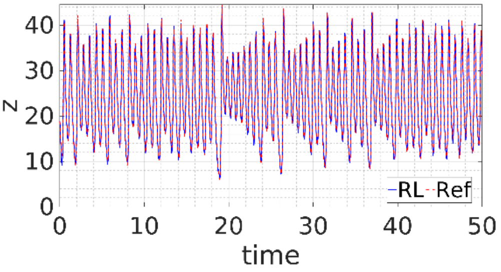
(g) ($\mathcal{N}(0,1)$, 5, $I_{d,\,3x3}$)



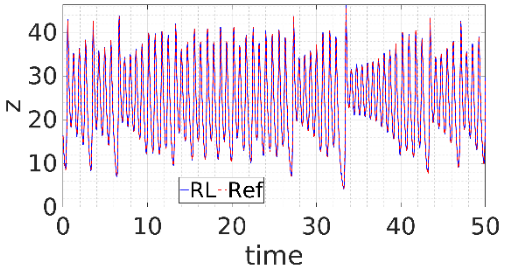
(h) ($\mathcal{N}(0,1)$, 50, $I_{d,\,3x3}$)



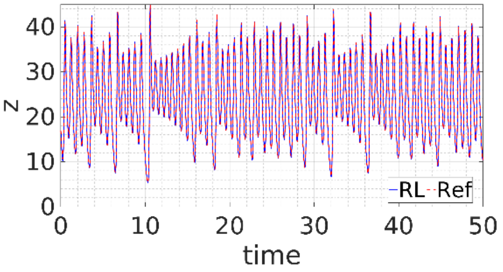
(i) ($\mathcal{N}(0,1)$, 100, $I_{d,\,3x3}$)



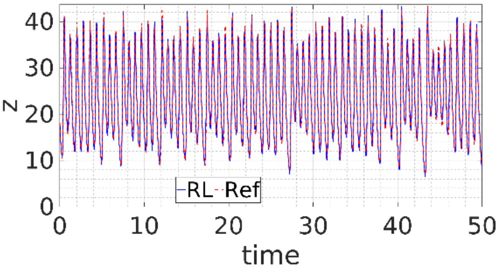
(j) ($\mathcal{N}(0,1)$, 50, $I_{d,\,3x3}$)



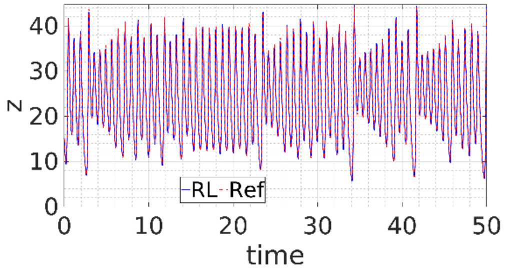
(k) ($\mathcal{L}(0,1)$, 50, $I_{d,\,3x3}$)



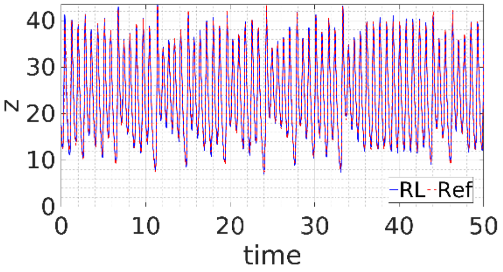
(l) ($\mathcal{U}(0,1)$, 50, $I_{d,\,3x3}$)



(m) ($\mathcal{N}(0,1)$, 50, *diag*(1,0,0))

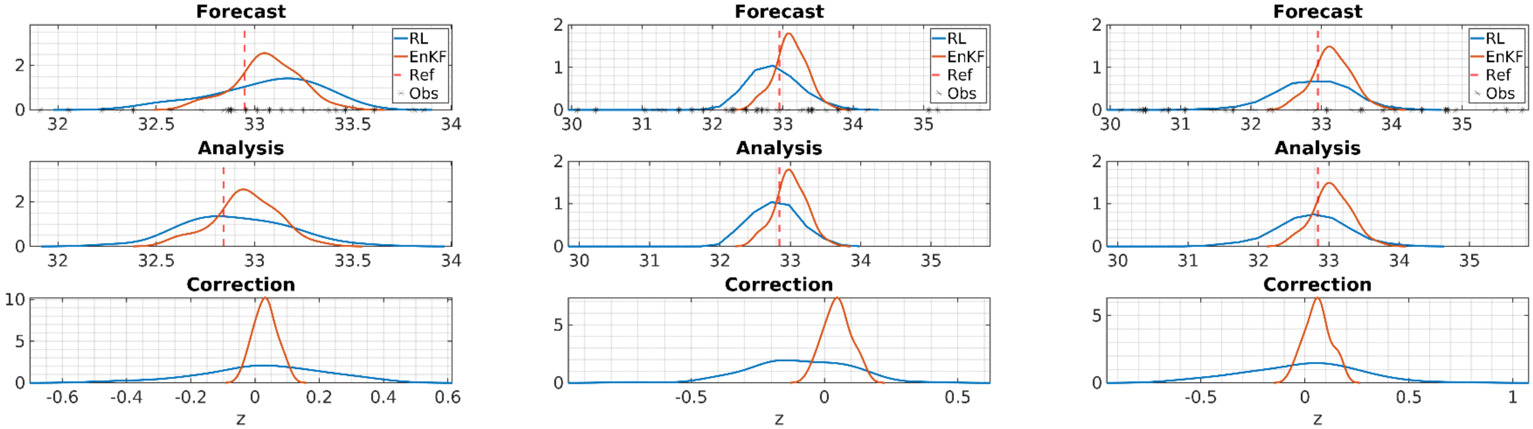


(n) ($\mathcal{N}(0,1)$, 50, *diag*(1,1,0))



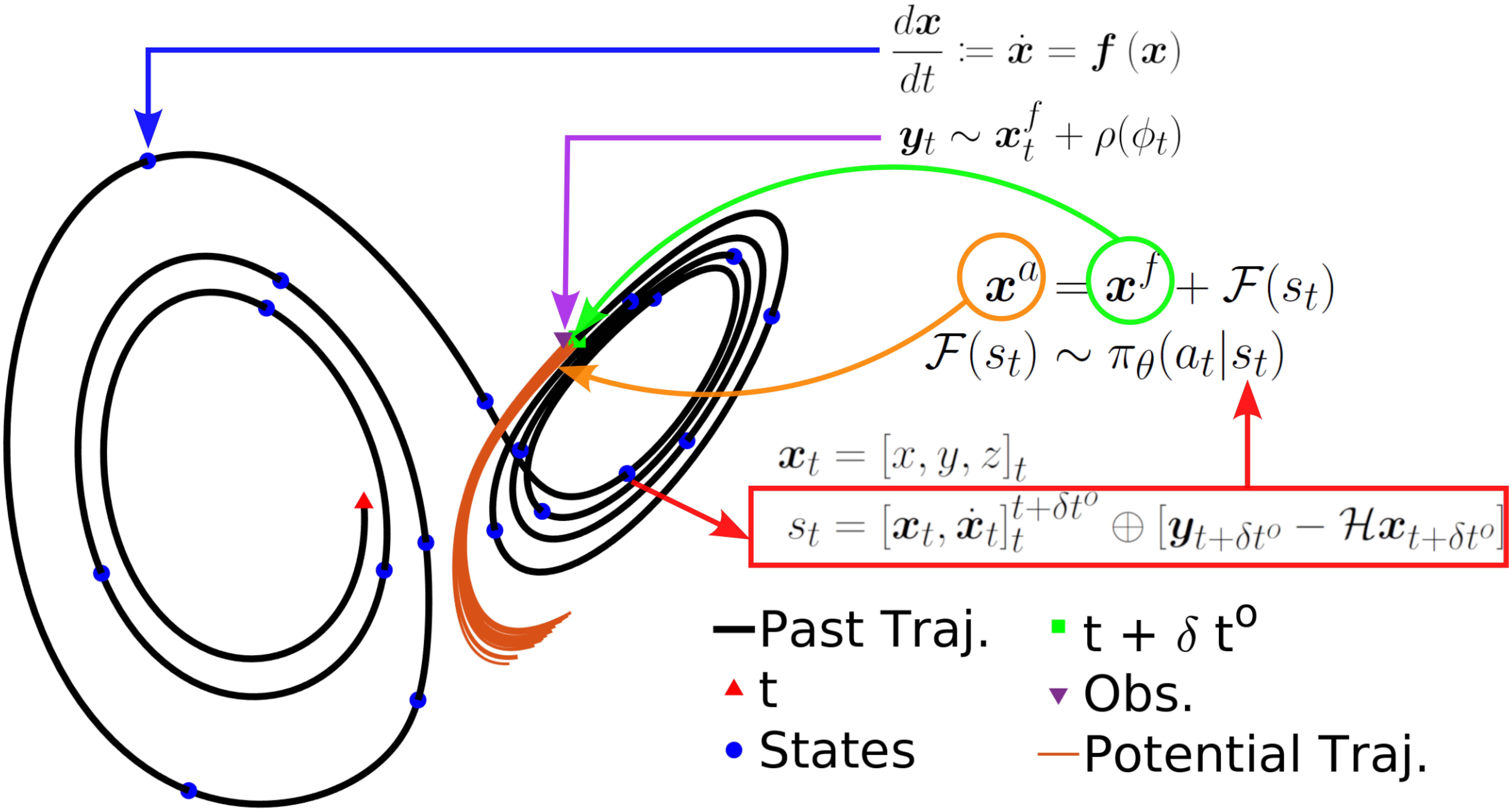
(o) ($\mathcal{N}(0,1)$, 50, *diag*(1,0,1))

Figure.



(a) ($\mathcal{N}(0,1)$, 50, $\mathrm{I}_{\mathrm{d,\ 3x3}}$) (b) ($\mathcal{N}(0,2)$, 50, $\mathrm{I}_{\mathrm{d,\ 3x3}}$) (c) ($\mathcal{N}(0,3)$, 50, $\mathrm{I}_{\mathrm{d,\ 3x3}}$)

Figure.


$\frac{d\boldsymbol{x}}{dt} := \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$
$\boldsymbol{y}_t \sim \boldsymbol{x}_t^f + \rho(\phi_t)$
$\boldsymbol{x}^a = \boldsymbol{x}^f + \mathcal{F}(s_t)$
$\mathcal{F}(s_t) \sim \pi_\theta(a_t|s_t)$
$\boldsymbol{x}_t = [x, y, z]_t$
$s_t = [\boldsymbol{x}_t, \dot{\boldsymbol{x}}_t]_t^{t+\delta t^o} \oplus [\boldsymbol{y}_{t+\delta t^o} - \mathcal{H}\boldsymbol{x}_{t+\delta t^o}]$
Past Traj.
t
States
$t + \delta t^o$
Obs.
Potential Traj.