Physics Informed Model Based Reinforcement Learning for Controlling Synchronization of Weakly Coupled Kuramoto System

Alif Bin Abdul Qayyum[*]
Texas A&M University
College Station, TX 77843
alifbinabdulqayyum@tamu.edu

A N M Nafiz Abeer[*]
Texas A&M University
College Station, TX 77843
nafiz.abeer@tamu.edu

Abstract

Kuramoto network as a representative of collective dynamics presents a challenging control task of affecting the synchronization of the interacting oscillators. As the dynamics become harder to estimate, making use of a learned model for controlling purposes is difficult. Learning through interactions with the environment enhanced by model-based reinforcement learning (MBRL) algorithms can alleviate the lack of sample efficiency involved with model-free reinforcement learning (MFRL) methods. Given prior knowledge of the underlying dynamics of the system, physics-informed MBRL can achieve even higher efficiency. In this study, we compare the performance of physics-informed MBRL, MBRL, and MFRL in synchronizing the Kuramoto network. We assess the scalability of these three reinforcement learning methods in a naturally chaotic or unsynchronized network.

1 Introduction

Control of a complex system of collective dynamics is often challenging, partly due to the difficulty and uncertainty involved in estimating the underlying dynamics. The model-free reinforcement learning (MFRL) methods happen to be successful in such applications but at the expense of poor sample efficiency. In fields like neuronal control where data collection is expensive, model-based reinforcement learning (MBRL) algorithms can be a compromise between sample efficiency and the collected rewards. Dyna-style MBRL [1] introduces the physics-informed framework [2] to reduce the gap between MBRL and MFRL. This physics-informed notion can be extended to many other established MBRL frameworks [3] which involve planning over a learned model of the environment that assists the learning process of optimal policy. In this work, we compared the performance of physics-informed MBRL (PiMBRL) with the MBRL and MFRL in synchronizing a system of oscillators, Kuramoto network [4]. Since the oscillators of Kuramoto network continuously interacts with each other, introducing synchronization in a naturally chaotic or unsynchronized network seems to be a challenging control task. The same control signals may push one oscillator towards synchronization while perturbing the harmony among others.

Our works are summarized as –

- We compared three RL approaches – MFRL, MBRL, and PiMBRL– in the task of increasing synchronization of an unsynchronized Kuramoto network.
- We performed experiments with two networks of different sizes to assess the robustness of these three approaches to the complexity of the system.

[*]These authors contributed equally to this work.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2 Problem Formulation

2.1 Kuramoto Model

A Kuramoto network of N oscillators can be represented by a system of coupled differential Equation (1). The interaction strength or coupling coefficients (K) between each pair of oscillators, number of oscillators (N), and the adjacency matrix $\mathbf{A}$ regulate the synchronization dynamics of $\{\theta_i\}$ of this system.

$$\frac{d\theta_i(t)}{dt} = \omega_i + \frac{K}{N}\sum_{j=1}^{N} A_{i,j}\sin(\theta_j(t) - \theta_i(t)), \quad i = 1, 2, \cdots, N \tag{1}$$

2.2 Enhancement of Synchronization as a Reinforcement Problem

In a Kuramoto network, the intrinsic frequencies ω_i of each oscillator push them toward independent oscillations. A smaller value for the coupling coefficient, K thus makes the whole system unsynchronized. To synchronize in such a weakly coupled unsynchronized system, we need to introduce control signals [5] that modify the phases of the oscillators in such a way that the whole system synchronizes. Gjata et al. [6] applied Hamiltonian control theory for desynchronizing a synchronized Kuramoto network. In their approach, the goal is to introduce perturbation signal $\phi_i(t)$ to each oscillator to disrupt the synchronized interaction of the network. We utilize their derived underlying system Equation (2) to increase the degree of synchronization.

Environment	$\mathcal{M}(\{\omega_i\}, K, \mathbf{A})$
State	$\mathbf{s}_{t_k} := \begin{bmatrix} \{\phi_i(t_{k-1})\} \\ \{\theta_i(t_{k-1+j/N_s})\}_{j=1,\cdots,N_s} \end{bmatrix}$
Action	$\mathbf{a}_{t_k} := [\{\phi_i(t_k)\}]$
Reward	$r_{t_k} := R(\{\theta_i(t_{k+1})\})$

Table 1: Problem formulation.

$$\frac{d\theta_i(t)}{dt} = \omega_i + \frac{K}{N}\sum_{j=1}^{N} A_{i,j}\sin(\theta_j(t) - \theta_i(t)) + \frac{d\phi_i(t)}{dt}, \quad i = 1, 2, \cdots, N \tag{2}$$

Equation 2 dictates how the network will respond to the introduction of the control signals $\{\phi_i\}$. The details of the simulation of this environment, $\mathcal{M}(\{\omega_i\}, K, \mathbf{A})$ are discussed in Appendix A.2.

At a certain time point t_k, the environment has knowledge about a set of angular positions, $\{\theta_i(t_k)\}$, and immediate past control signal $\{\phi_i(t_{k-1})\}$. When the learner takes action $\{\phi_i(t_k)\}$, $\mathcal{M}(\{\omega_i\}, K, \mathbf{A})$ moves to a new set of $\{\theta_i(t_{k+1})\}$ and the learner gets the reward $r_{t_k} = R(\{\theta_i(t_{k+1})\})$. $R(\cdot)$ needs to have a higher value when the network becomes more synchronized, and the order parameter (equation 3) of the Kuramoto model is a natural candidate for this.

$$R := R(\{\theta_j\}) = \frac{1}{N}\left|\sum_{j=1}^{N} e^{i\theta_j}\right| \in [0, 1] \tag{3}$$

To formulate the control problem in a reinforcement learning framework, we treat it as a continuing task[7]. Thus, the agent's goal is to learn some policy $\pi_{\boldsymbol{\theta}}$ to maximize the discounted reward $\sum_{k=0}^{\infty}\gamma^{t_k} r_{t_k}$. In our work, we consider $\{t_k\}$ are evenly spaced within $[0, T]$, where T is the maximum duration of the task. Here $\boldsymbol{\theta}$ represents the learnable parameters of the policy (actor) network. To take an action $\{\phi_i(t_k)\}$, we allow the policy network to utilize $\{\theta_i(t_{k-1+j/N_s})\}_{j=1,\cdots,N_s}$ along with past action, $\{\phi_i(t_{k-1})\}$. N_s is the length of angular states, i.e. the number of immediate oscillator phases including $\{\theta_i(t_k)\}$. So the state representation for our problem is the concatenation of $N_s \times N$ oscillator phases and N previous control actions. We restrict the spaces of $\{\theta_i\}$ to $[0, 2\pi]$ and $\{\phi_i\}$ to $[0, \pi]$. The latter choice is made to facilitate the learning process of the actor-critic-based learning algorithm we used in our work. Table 1 shows the summary of the RL problem formulation.

3 Methodology

We consider a Kuramoto network with N oscillator, with a very low coupling coefficient, K which causes lower synchronicity among oscillators. To learn the optimal control policy for increasing synchronization, we pursue the model free and model based learning algorithms.

3.1 Model Free Reinforcement Learning (MFRL)

As a baseline approach, we have considered the twin delayed DDPG (TD3)[8] for policy optimization with the usage of $\sin$ instead of $\tanh$ activation function at the output layer of the actor network. With $\tanh$, the predicted action values were often close to the maximum or minimum limit of action values resulting in no improvement to the uncontrolled scenario.

3.2 Model-based Reinforcement Learning (MBRL)

Model-based approach tries to reduce the number of interactions between the agent and the environment, using a simulator or fictitious environment. In our work, we apply the TD3 algorithm through interactions with the environment and we use those interactions data from the "real" replay buffer to learn the underlying model of the environment. Specifically, the environment of Kuramoto network, $\mathcal{M}(\{\omega_i\}, K, \mathbf{A})$ is modeled as $\mathcal{M}_F(\{\hat{\omega}_i\}, \hat{K}, \hat{\mathbf{A}})$, where $\hat{\omega}_i, \hat{K}, \hat{\mathbf{A}}$ are learned by minimizing the data loss, $L_D = \frac{1}{n_b}\|\mathbf{s}_{t+1} - \hat{\mathbf{s}}_{t+1}\|^2$, between predicted and true next state for batches of transition pair $(\mathbf{s_t}, \mathbf{a_t}, \mathbf{s_{t+1}}, r_t)$ from real replay buffer, $\mathscr{D}^r$. $\hat{\mathbf{s}}_{\mathbf{t+1}}$ is the next state prediction when we initialize the state of $\mathcal{M}_F(\{\hat{\omega}_i\}, \hat{K}, \hat{\mathbf{A}})$ as $\mathbf{s}_t$ and apply $\mathbf{a}_t$ on this model of the environment. Once we have very small data loss, we use the learned model, $\mathcal{M}_F$ as a parallel source of interaction data along with the real environment. For one interaction with the real environment, we perform r_M steps in the simulator $\mathcal{M}_F$. Data collected from the latter interactions are stored in another experience replay buffer, $\mathscr{D}^f$. Since samples in $\mathscr{D}^f$ are collected in a parallel manner, the policy optimization has the opportunity of seeing a large amount of data with a lower number of interactions with the real environment. Faster convergence to the optimal policy is expected as long as the learned model $\mathcal{M}_F$ is reliable. Algorithm 1 shows the pseudocode for model based learning used in our work.

3.3 Physics-informed Model-based Reinforcement Learning (PiMBRL)

Using a simulator can hinder the agent's learning by introducing modeling errors that mislead the policy optimization with inconsistent environmental behavior. To incorporate the prior knowledge about the environment into the modeling of the simulator, we add another loss, $L_r = \frac{1}{n_b}\left\|\dot{\mathbf{s}}_{t+1} - \frac{d\hat{\mathbf{s}}_{t+1}}{dt}\right\|^2$ which is the residual loss between the true gradients and the estimated gradients for the states. The true gradients $\dot{\mathbf{s}}_{t+1}$ is collected along with $(\mathbf{s_t}, \mathbf{a_t}, \mathbf{s_{t+1}}, r_t)$. In Algorithm 1 these additional information is included in $\mathbf{d}_{t+1}$. The predicted gradients are obtained through the simulator, $\mathcal{M}_F$. The objective is to minimize the total loss of $2L_D + 5L_r$. The relative weight between the residual loss and data loss is chosen from the common practice of physics-informed neural network (PINN)[2]. When both L_D and L_r are decreased beyond a very small threshold (λ), we begin collecting the data from the learned model, $\mathcal{M}_F$. The rest of the framework is the same as the vanilla MBRL.

4 Results

For a fully connected N oscillators system, we initialize the $\mathcal{M}(\{\omega_i\}, K, \mathbf{A})$ with evenly spaced intrinsic frequencies and choose a very small value for K, so that $\mathcal{M}$ has a lower degree of synchronization without any external control signals. We considered Kuramoto networks with $N = 5, 10$.

4.1 Enhancement of Sample Efficiency

To see whether the physics prior helps the PiMBRL over MBRL, we first use $N = 5$ oscillators. In the first 2000 training steps, we select actions by taking random samples uniformly within $[0, \pi]$. Following this exploration policy, we begin executing actions according to the actor network of TD3 algorithm. For all three approaches, the agents are trained for 50000 iterations. In every 500^{th} training iteration, the RL agents are evaluated by collecting the average reward obtained over $n_{eval} = 5000$ steps in a separate environment. Figure 1 shows the smoothed evaluation reward for three approaches with different values of fictitious to real data usage ratio, r_M. In all values of r_M, the two model-based approaches show faster convergence to higher evaluation rewards. If we look closely towards the early steps, physics-informed model-based reinforcement learning shows slightly faster improvement compared to the vanilla MBRL. This is most prominent for $r_M = 20$. However, MBRL quickly closes the gap in this smaller network of 5 oscillators.

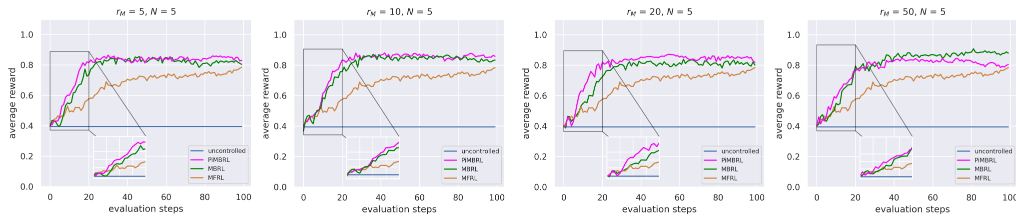

Figure 1: Evaluation reward for $r_M = 5, 10, 20, 50$ for 5 oscillator system. During the training of the agent, in every 500^{th} iterations the learned agent is applied to the environment, and the average of cumulative undiscounted rewards is shown here as the average reward.

4.2 Impact of Larger Network

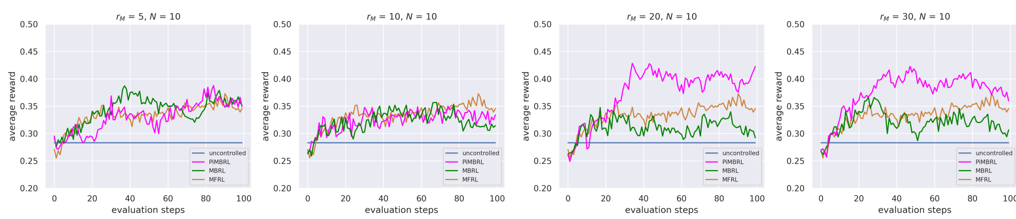

Figure 2: Evaluation reward for $r_M = 5, 10, 20, 30$ for 10 oscillator system.

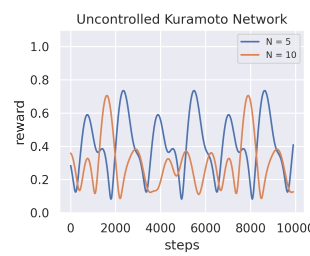

Figure 3: The reward for uncontrolled networks with $N = 5, 10$

As we increase the size of the network to $N = 10$, the dynamics of the Kuramoto network become more involved. Figure 3 shows the reward, i.e. order parameter of the uncontrolled networks for $N = 5$ and 10. The larger network has more concentration toward smaller rewards which makes the task of RL agent more challenging compared to $N = 5$.
We repeat the same experimental procedure as $N = 5$, except for evaluation after every 500^{th} training iteration, we run the agents for $n_{eval} = 6000$ steps to include all the major transitions of the uncontrolled case. For $r_M = 5, 10$, all three approaches show similar convergence rates (Figure 2). However, the MBRL shows a small level of instability which is more pronounced for higher r_M. As MBRL only optimizes for data loss, in this slightly larger network the learned model fails to capture the underlying dynamics of the Kuramoto network. And with a high usage rate of generated data from that unreliable model, the policy learning algorithm gets confused with the erroneous data. On the other hand, the PiMBRL shows significant improvement for $r_M = 20, 30$ as the data generated by its learned model is more accurate representation of the dynamics. Figure 4 in Appendix A.4 shows the learned agent's performance controlling the networks when $r_M = 20$ for three learning approaches.

5 Conclusion

Finding the control signal for increasing synchronization in an unsynchronized Kuramoto network is challenging due to mutual interactions among the oscillators. In our work, we apply the reinforcement algorithms to assess how reliable they are for different sizes of networks. Our empirical results demonstrate that the physics prior does not add significant improvement over the MBRL for a smaller network. With an increased complexity of a larger network, PiMBRL shows better performance over MFRL whereas the MBRL approach deteriorates because of the inaccuracy in modeling the environment. Leveraging a generative model [9] along with the prior knowledge about the environment can be explored next to highlight the importance of the model-based approach.

References

[1] Xin-Yang Liu and Jian-Xun Wang. Physics-informed dyna-style model-based deep reinforcement learning for dynamic control. *Proceedings of the Royal Society A*, 477(2255):20210618, 2021.

[2] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[3] Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey, Mar 2022. URL https://arxiv.org/abs/2006.16712.

[4] Yoshiki Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. *Lect. Notes Phys.*, 39:420–422, 1975. doi: 10.1007/BFb0013365.

[5] Jordan Snyder, Anatoly Zlotnik, and Aric Hagberg. Stability of entrainment of a continuum of coupled oscillators. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10):103108, 2017.

[6] Oltiana Gjata, Malbor Asllani, Luigi Barletti, and Timoteo Carletti. Using hamiltonian control to desynchronize kuramoto oscillators. *Physical Review E*, 95(2):022209, 2017.

[7] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[8] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[9] Cong Lu, Philip Ball, Yee Whye Teh, and Jack Parker-Holder. Synthetic experience replay. *Advances in Neural Information Processing Systems*, 36, 2023.

[10] BA Mitchell and Linda R Petzold. Control of neural systems at multiple scales using model-free, deep reinforcement learning. *Scientific reports*, 8(1):1–12, 2018.

[11] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

A Appendix

A.1 Related Work

For introducing synchronization into a weakly coupled Kuramoto network Snyder et al. [5] studies an experimental setting of applying a periodic excitation signal to focus each individual oscillation towards a predefined pattern. In the Hamiltonian analysis of [6], the control signal to perturb an already synchronized Kuramoto network is analyzed. Based on their formulation, Mitchell and Petzold [10] tries to apply reinforcement learning ideas to design the control signal by interacting with Kuramoto network. Specifically, they applied Deep Deterministic Policy Gradients (DDPG)[11] to learn the policies for the control signals in model free fashion.
The idea of using prior physics information in model-based reinforcement learning is first proposed by Liu and Wang [1]. Their proof of concept experiments shows significant improvement by PiMBRL over MBRL. However, the dimension of those problems is not large enough to see how well the performance transfers for large-scale networks.

A.2 Simulation of Dynamics of Kuramoto Network with Control Signal

We have created an environment object for the Kuramoto network with external control signals. Given an action i.e. the control signals, the system of coupled differential equations in Equation 2 is solved via numerical method (Euler's method) to approximate the change in oscillator phases due to rate of change in the control signal. As the chosen step size is small enough, we have not to used higher-order methods. In the following paragraph, we provide the exact steps we followed to simulate the Kuramoto network, $\mathcal{M}(\{\omega_i\}, K, \mathbf{A})$.
First, the step size for Euler's method, h is set as $\frac{\Delta t_s}{N_s}$ with $\Delta t_s = 0.01$. We assume to have $\{\phi_i\}$ at some discrete time points $\{t_k\}$. To approximate the $\dot{\phi}_i(t)$, we use the frame-skipping-like technique [12] commonly used in playing Atari games. Specifically, when the agent executes an action $\{\phi_i(t_k)\}$, we approximate $\left\{\dot{\phi}_i(t)\right\}$ as in equation 4.

$$\frac{d\phi_i(t)}{dt} \approx \frac{\phi_i(t_k) - \phi_i(t_{k-1})}{\Delta t_s} \quad \forall t \in \left\{t_{k+1-j/N_s}\right\}_{j=1,\cdots,N_s} \tag{4}$$

Next, we apply the following recursive Euler's forward step to find the $\left\{\theta_i(t_{k+j/N_s})\right\}_{j=1,\cdots,N_s}$

$$\begin{aligned}\frac{\theta_i\left(t_{k+(n+1)/N_s}\right) - \theta_i\left(t_{k+n/N_s}\right)}{h} &= \omega_i + \frac{K}{N}\sum_{j=1}^{N} A_{i,j}\sin(\theta_j\left(t_{k+n/N_s}\right) - \theta_i\left(t_{k+n/N_s}\right)) \\ &+ \frac{\phi_i(t_k) - \phi_i(t_{k-1})}{\Delta t_s} \quad \text{for } n = 0, 1, \cdots, N_s - 1\end{aligned} \tag{5}$$

From equation 5, we get the $\{\theta_i(t_{k+1})\}$ caused by $\{\phi_i(t_k)\}$, then the reward for the agent is determined by equation 3.
For an uncontrolled Kuramoto network, we omit the approximated $\dot{\phi}_i(t)$ (third term in right-hand side of equation 5), and the resultant $\{\theta_i(t)\}$ are used to measure the degree of synchronization for the uncontrolled case.

A.3 Hyperparameters

Table 2: Values of different hyperparamters

Parameter and Value	
N_s	10
T	50000
γ	0.99
n_M	5000
λ	1e−8 ($N = 5$), 1e−6 ($N = 10$)
n_f	10
n_b	128
actor l_r	5e−4 ($N = 5$),1e−3 ($N = 10$)
critic l_r	1e−3 ($N = 5$),2e−3 ($N = 10$)
exploration policy duration	2000

Algorithm 1 Model-Based Reinforcement Learning (MBRL and PiMBRL) in Our Work

Start with randomly initialized actor network $\pi_{\boldsymbol{\theta}}(\mathbf{s})$, critic network(s) $q_{\boldsymbol{\phi}}(\mathbf{s}, \mathbf{a})$, fictitious model, $\mathcal{M}_F$, and replay buffers $\mathscr{D}^r$, $\mathscr{D}^f$ for real and fictitious environments.
Collect $(\mathbf{s}_0, \mathbf{d}_0)$ from the real environment, where $\mathbf{d}_0$ may contain additional information like episode termination signal, gradients of states etc.
for $i = 0, \cdots, T$ **do** ▷ T : maximum episode length
    Take action $\mathbf{a}_i = \pi_{\boldsymbol{\theta}}(\mathbf{s}_i)$ in the real environment.
    Add $(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}_{i+1}, r_i, \mathbf{d}_i, \mathbf{d}_{i+1})$ to the real buffer $\mathscr{D}^r$;
    if real buffer $\mathscr{D}^r$ has at least n_M samples **then** ▷ n_M: starting iteration for learning $\mathcal{M}_F$
        Update the fictitious model, $\mathcal{M}_F$ using the data loss L_D (MBRL) or combination of L_D and residual loss L_r (PiMBRL) on the batches from $\mathscr{D}^r$;
    end if
    if fictitious model meets the accuracy threshold ($L_D < \lambda$ or/and $L_r < \lambda$) **then**
        Reset the $\mathcal{M}_F$
        for $j = 1, \cdots, r_M$ **do** ▷ r_M : fictitious to real data usage ratio
            Collect current $(\mathbf{s}_j, \mathbf{d}_j)$ from $\mathcal{M}_F$
            Take action $\mathbf{a}_j = \pi_{\boldsymbol{\theta}}(\mathbf{s}_j)$ in the model $\mathcal{M}_F$;
            Add $(\mathbf{s}_j, \mathbf{a}_j, \mathbf{s}_{j+1}, r_j, \mathbf{d}_j, \mathbf{d}_{j+1})$ to $\mathscr{D}^f$;
        end for
    end if
    if $i \equiv 0 \mod n_f$ **then** ▷ n_f : agent update frequency
        if real buffer $\mathscr{D}^r$ has at least n_b samples **then**
            Update policy parameters $\boldsymbol{\theta}$ and value parameters $\boldsymbol{\phi}$ using sampled batches from $\mathscr{D}^r$
        end if
        if fictitious buffer $\mathscr{D}^f$ has at least n_b samples **then**
            Update policy parameters $\boldsymbol{\theta}$ and value parameters $\boldsymbol{\phi}$ using sampled batches from $\mathscr{D}^f$
        end if
    end if
end for

A.4 Performance of Learned Agent

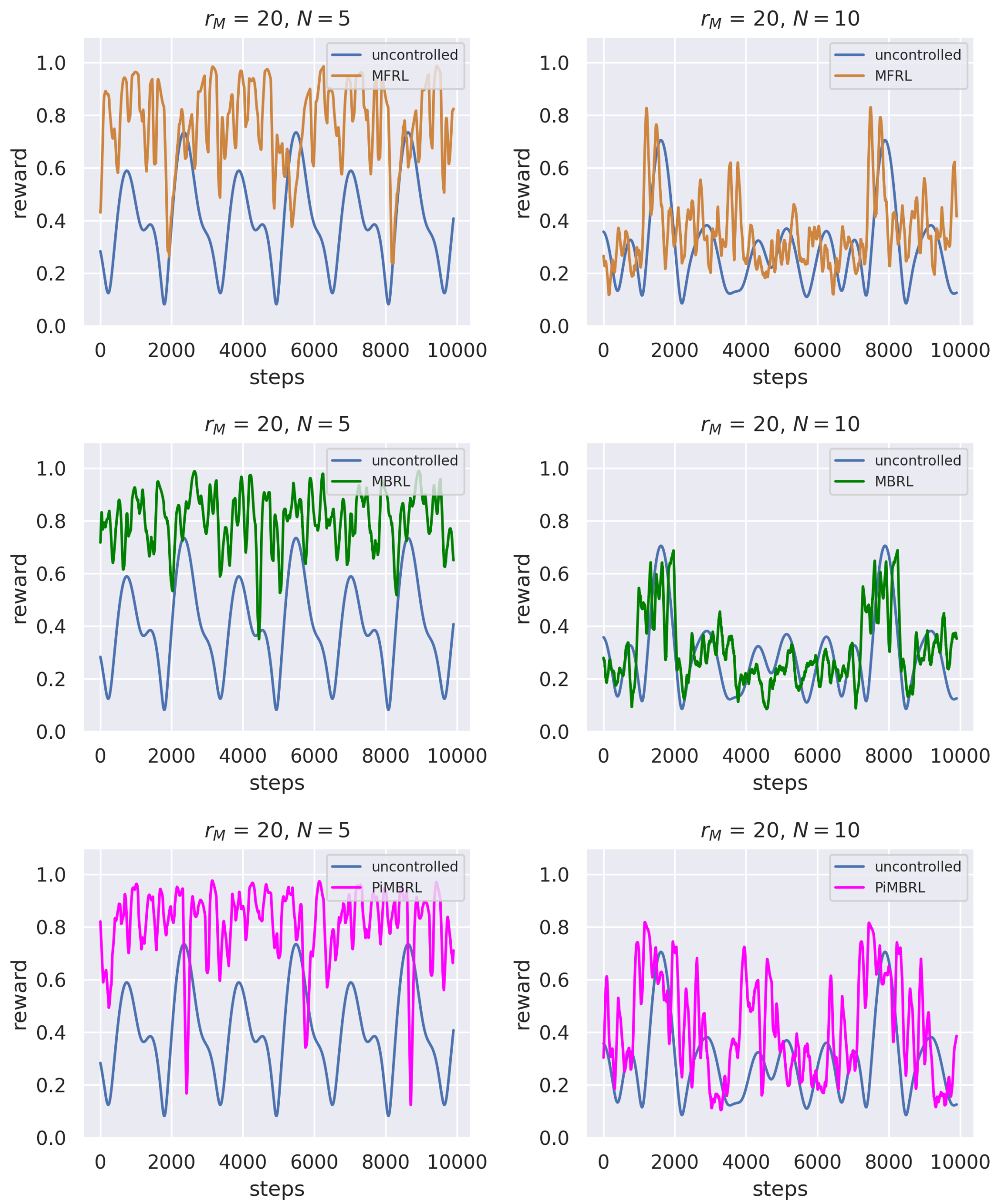

Figure 4: Performance of the learned agents for MFRL, MBRL, and PiMBRL along with an uncontrolled oscillators Networks for $N = 5, 10$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: This paper claims that physics informed model-based reinforcement learning achieves superior performance in controlling a weakly coupled Kuramoto oscillator network compared to its model-based and model-free counterparts.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Empirical results demonstrated in this work shows the limitations of our proposed approach in some of the problem scenarios.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Background theory has been discussed in detail (both in the main paper and in the appendix).

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper discloses the methods used with all the necessary hyper-parameters.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release the code and data upon acceptance of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The rewards shown in the evaluation phase have been time averaged, indicating the statistical signigicance of the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provide sufficient information on the computer resources needed to reproduce the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Controlling of a weakly coupled system has several applications ranging from industrial control system to medical research. We do not know of any negative societal impact of this work.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All sources have been cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Details of the model has been discussed. Code implementations will be released upon acceptance.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.