Response Letter

(ID: TNNLS-2022-P-25352)

Dear Associate Editor, and Reviewers,

We would like to express our appreciation for your evaluation of our paper and for providing us with many useful and helpful comments and suggestions. We have seriously considered these comments and suggestions and carefully revised the manuscript accordingly. The details are explained below (response numbers are in one-by-one correspondence with the comments).

To Associate Editor and All Reviewers: Statement of Changes

All of the modifications have been highlighted and colored in blue in the revised paper. The major points are summarized as follows:

1. Revise the introduction and include relevant references.
2. Fine-tune hyperparameters to improve the performance of modeling and control.
3. Evaluate the computational complexity of DeSKO and make a comparison to baselines.
4. More discussions on the experimental results.
5. The appendix is uploaded to both the submission system and a GitHub repo (`https://github.com/hithmh/TNNLS-supplementary-materials`) for the reviewers' convenience.
6. Typos and inaccurate statements are revised.

0 To Associate Editor

This paper proposed a deep stochastic Koopman operator method to learn and control uncertain nonlinear dynamics. Overall, the paper is organized well. In summary, there are several concerns from reviewers:

Many thanks to the Associate Editor for handling our paper. In the revised version, all the concerns from the reviewers have been carefully addressed. For the convenience of your and the reviewers' checking, all the corrections and modifications are marked in blue color in the revised paper.

0-1. *More discussions and explanations are needed to improve the paper's quality.*

Many thanks for the AE's kind suggestion and summary. We have included more discussions on the experiment results and more evaluation tests to address the reviewers' concerns, details can be found in Responses 1-3, 1-5, 2-5, and 3-4 to 3-10.

0-2. *It's better to provide more detailed proof and additional experiments like computational complexity analysis and parameter sensitivity analysis.*

Thanks for the AE's advice. In the previous submission, the proof was in a supplementary material file. However, from my own experience of being a reviewer at TNNLS, a reviewer does not get access to these files, at least it was not obvious to me. Therefore I assume Reviewer 3 did not see this either. In this submission, we have uploaded the Appendix to both the submission system and GitHub for the reviewer's convenience. We've also mentioned this in Response 3-3 to Reviewer 3. The discussion on parameter sensitivity and computational complexity can be found in Responses 2-5 and 3-4.

In the revised manuscript, we have conducted more experiments to evaluate the computational complexity and parameter sensitivity as the AE and the reviewers suggested. Specific details can be found in Response 3-4 to Reviewer 3.

0-3. *The language expressions need to be carefully proofread.*

Thanks for the AE's advice, we have proofread the paper to improve the writing quality of the paper. We've addressed this in Response 2-7 to Reviewer 2.

Once again, we appreciate the Associate Editor's encouraging evaluations of our work.

1 To Reviewer #1

1-1. *This paper proposed a new data-driven framework for the modeling and control of nonlinear uncertain systems. The authors further developed the Koopman operator theory by introducing deep learning and variational inference techniques. Inferring the distribution of observables is a crucial contribution of this work because it enables the Koopman operators to describe stochastic systems. In the control part, the authors proved that DeSKO is stabilizing to the closed-loop system, even in presence of approximation errors. Simulation results show that DeSKO is applicable to many systems, and outperforms state-of-the-art baselines. The validation of the proposed approach on a real-world soft robot arm is plausible. I think the authors developed a novel learning control framework and provided solid proof and theoretical analysis. The simulation and real-world experiments are thorough and show a clear performance margin compared to existing methods. However, there are still some issues I would like the authors to address before the manuscript can be accepted. Following are my detailed questions.*

Thanks for the reviewer's general evaluation of this paper and the affirmative comments on the contribution of the proposed method.

1-2. *The discussion and comparison between DMD, EDMD, and DeepDMD on page 2 are distributed in two paragraphs, can you merge them and make them concise?*

Many thanks for the reviewer's advice, we have merged the discussion and made it more compact in Section I. Please find the revised discussion as follows.

Page 1

Recently, the Koopman operator theory has attracted a lot of attention. The Koopman operator can model complex nonlinear systems by linearly propagating observables in an infinite-dimensional functional space [15]. Based on this idea, practical algorithms, like dynamic mode decomposition (DMD) and extended dynamic mode decomposition (EDMD), were developed by selecting a finite set of parameterized linear and nonlinear observable functions and obtaining the Koopman operators through least square optimization [16], [17]. Deep-DMD exploits deep learning techniques to automate the design of observable functions [18]. Deep-DMD automatically learns a richer set of observable functions and can be more accurate than DMD or EDMD, even with fewer observables [3]. Thanks to the capacity of neural networks, Deep-DMD does not suffer from the curse of dimensionality; that is, the number of observables does not grow exponentially with the number of states. However, the methods discussed above focus on modeling nominal systems with clean data sets; they do not consider process noise or observation noise.

1-2. *In Section III.B, the authors assumed the probabilistic neural network to be Lipschitz continuous, I wonder if this is a trivial assumption. If not, how did you assure this assumption in practice?*

Thanks for the reviewer's valuable question. There are many ways to ensure the Lipschitz continuity of the NN, and one of the most straightforward approaches is to choose Lipschitz continuous activation functions, such as sigmoid or ELU. In this work, we chose ELU as the activation function, which is Lipschitz continuous and also shows a similar performance as ReLU. The design details had already been included in Appendix III on page 20.

1-3. *The data collection part in the SoPrA experiment is not clear to me. What are the two patterns of data collection mean and what were your purposes for designing them?*

Many thanks for the reviewer's advice, we definitely should elaborate more on the details of data collection. The details on the data collection pattern have been added to Appendix III as follows.

Page 19

In the SoPrA experiment, we introduced two patterns of random input generation: the sinusoidal input and the polyline input. In the sinusoidal input pattern,

$$a_t = \alpha \left[\sin(\omega_1 t), \sin(\omega_1 t + \frac{2\pi}{3}), \sin(\omega_1 t + \frac{4\pi}{3}), \sin(\omega_2 t), \sin(\omega_2 t + \frac{2\pi}{3}), \sin(\omega_2 t + \frac{4\pi}{3}),\right]^{\mathrm{T}} + \beta \mathbf{1},$$

where $\omega_{1,2}$, α, and β are randomized every 1000 steps to generate sinusoidal inputs with different magnitudes and frequencies. In the polyline input pattern, the input $a_t = p_c + \frac{p_t - p_c}{T} t$, where $p_c \in \mathbb{R}^6$ denotes the initial pressure, $p_t \in \mathbb{R}^6$ denotes a randomly generated target pressure and T is the randomly generated time interval.

1-4. *When the arm was holding an unknown object, the tracking error trajectories seem to be shaking quite a lot, can you explain the reasons for this? Is there a way to eliminate those tremors?*

Thanks for the reviewer's valuable question. We included the following discussion in the manuscript.

Page 14

The vibration in the tracking error is actually correlated to the soft nature of the SoPrA arm's material. The arm was built with silicone rubber and driven by pressurized air, which makes its actuation particularly prone to disturbances. Furthermore, the speed of the movements in this paper is far beyond the quasi-static actuation mode [30], [31] that most control approaches for soft robotic arm use. Thus, the vibration level becomes more apparent. Overall, the vibration in the state trajectory can be depressed by augmenting the body and material, as well as sending mild movement commands to the robot.

1-5. *The data points in Fig. 11 are overlapping each other, please finetune the picture, e.g. paint them with different colors to denote the third dimension.*

We appreciate the reviewer's valuable advice. We have revised Fig.11 to show a third dimension in 2D graph by using different colors.

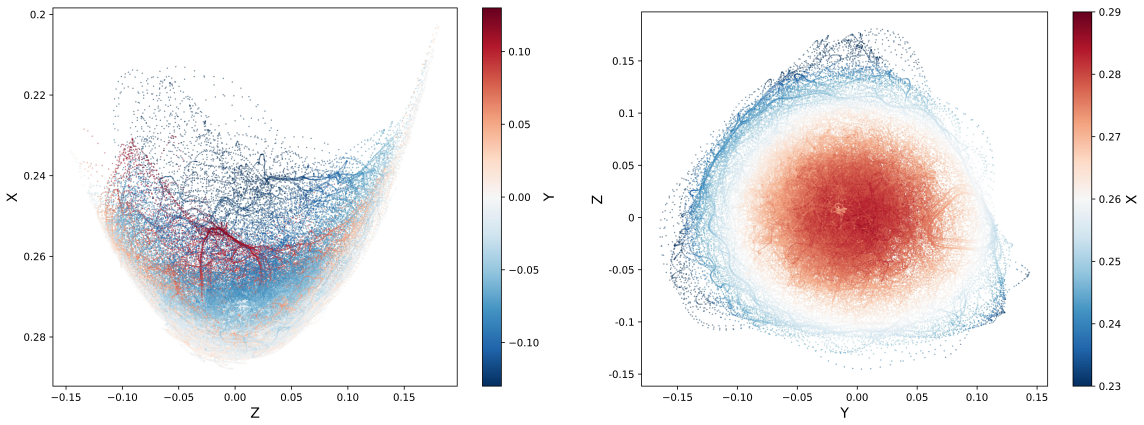


Figure 11: Visualization of the 200,000 data points on the X-Z and Y-Z plane.

2 To Reviewer #2

2-1. *This paper presents a technique to perform data-driven modeling and control for robotic systems by applying the Koopman Operator Framework. To illustrate its utility the paper, the paper proves that the proposed closed-loop system framework generates uniformly ultimately bounded behavior. Finally, the paper evaluates the proposed framework and compares it to several state-of-the-art methods on a variety of applications both in simulation and in the real world. I found the paper well-written and the experimental section was quite convincing. Below are some of my suggestions for the manuscript:*

Thanks for the reviewer's general evaluation of this paper and the affirmative comments on the contribution of the proposed method.

2-2. *I think the figures in this paper are well-organized and informative. However, in Fig.2, the term "recursive propagation" is a bit confusing to me. I believe that this becomes clear later when reading the algorithm block, but I think the picture itself could be clearer by adding some content.*

Thanks for the reviewer's valuable advice. The 'recursive propagation' was referring to how the model does multi-steps forward prediction. According to the reviewer's suggestion, we have now removed this term to avoid confusion.

2-3. *The paper says "The parameters in (4) . . . " but it's not clear which parameters. All of them? A subset?*

Thanks for the reviewer's question and advice, the parameters we refer to are A, B, C, and θ. We have now explicitly noted this in the revised version in Section II B.

2-4. *I have a question regarding the proofs. In the value function equation (with no number) just after equation 18, what happened to the terms with $+q$ and $+p$? From there to the next inequality, they just seem to disappear. Can you explicitly state in the paper how that works? I also suggest the authors to index all the*

equations in the text no matter they were referenced or not, because that would be helpful for the reviewers to give comments.

We appreciate the reviewer's valuable question. We apologize for the confusion we made in the proof. In the revised manuscript, we have elaborated on the details of the derivation. And we agree that indexing all the equations would be helpful for review, and thus added the indexes in the revised version.

2-5. *In section IV E, (and the accompanying figure), the authors explain the effect of the entropy threshold on robustness. However, in the literature on robust control in general, one usually gives up aggressiveness or performance by making a system more robust. I wonder what trade-off the authors made in this work.*

Thanks a lot for the reviewer's insightful question. The entropy constraint in this paper is used as a technique to prevent potential overfitting caused by the neural network. The goal of the pipeline is to train an accurate enough model without overfitting, thus we chose a mild threshold of $\mathcal{H} = -20$. In our tests, this threshold can prevent overfitting without decreasing the modeling accuracy, thus producing satisfying control performance. In ablation evaluation, increasing this threshold can produce more robust control behavior in the presence of disturbances, and thus increase the chance of survival from falling over.

2-6. *The following recent works on learning-based control could be included: Reinforcement learning-based decentralized fault tolerant control for constrained interconnected nonlinear systems; Periodic event-triggered adaptive tracking control design for nonlinear discrete-time systems via reinforcement learning.*

We greatly appreciate the reviewer's advice. The discussion on the above recent works has been added to the Introduction in the revised manuscript.

Page 1

In addition to the model-based methods, reinforcement learning methods have also achieved impressive performance on the control of various kinds of nonlinear systems [13], [14].

2-7. *Some typos: In Remark 1, "To simply the notations" =¿ "To simplify the notations" In Fig.6, the label for GRN(Process Noise) should be (f).*

Many thanks for the reviewer's kind notification. We have fixed these typos in the revised manuscript.

3 To Reviewer #3

3-1. *A universal and robust control framework for the general class of uncertain nonlinear systems is proposed in this paper. A data-driven deep stochastic Koopman operator model (DeSKO) and robust learning control framework learns the uncertainty of dynamical systems by inferring a distribution of observables. A model predictive control framework with integral action to compensate for run-time parametric uncertainty. Moreover, simulations and experiments are performed to demonstrate the robustness of the proposed approach on the soft robotic arm. The following questions and suggestions should be clarified.*

Thanks for the reviewer's general evaluation of this paper and the affirmative comments on the contribution of the proposed method. We appreciate the questions and suggestions, and we address them all in the following.

3-2. Both Gaussian noise and uniform noise appear in this work, why?

We appreciate the reviewer's valuable question and we try to address this question as follows. The random noise considered in this work is subject to an unknown distribution p_w, as noted in (2). Thus in this paper, we used the most common noises, including Gaussian noise and uniform noise, to validate the effectiveness of our approach. In our approach, a multivariate Gaussian distribution parameterized by neural networks is used to approximate the unknown distribution of observables. Gaussian distribution is commonly employed in probabilistic machine learning due to its desirable properties, e.g. the existence of closed-form solutions, convenient to implement, and is thus suitable for various applications. This could be a most promising choice when one has no prior knowledge of the type of system or noises. As noted in Section II B, other types of parameterized distribution can be more appropriate when a particular type of system/noise is being studied, and we leave it for future exploration due to space limits. We have included this discussion in Section II B of the revised version as follows.

Page 3

The multivariate Gaussian distribution is commonly employed in probabilistic machine learning to model uncertainty due to its many desirable properties and its suitability for various applications [25]. While a Gaussian distribution is effective in our experiments, other forms of parameterized distributions could be more appropriate for specific systems.

3-3. The proof of uniformly ultimately bounded is suggested to be provided.

We appreciate the reviewer's kind suggestion. In the previous submission, we included the proof in Appendix I of the appendix file and uploaded it to the system separately due to the page limit on the main file. However, **we didn't realize that this file may not be visible to the reviewers and we sincerely apologize for that**. In this submission, we will upload the file to a GitHub repository in case you still do

not get access: https://github.com/hithmh/TNNLS-supplementary-materials. For the reviewer's convenience, we also post the proofs in the following.

Page 16

Proof of Proposition 1

Proof. The stability proof is composed of two parts: the stability of the nominal system, and the stability of the error system between the actual system (12) and the nominal system (8). First, we prove the stability of the nominal system (8). By solving the MPC problem (9)-(10), we can obtain the optimal control sequence

$$\{c^*_{t|t}, c^*_{t+1|t}, \ldots, c^*_{t+H-1|t}\} \tag{23}$$

and the resulting optimal state trajectory

$$\{\hat{\mu}^*_{t|t}, \hat{\mu}^*_{t+1|t}, \ldots, \hat{\mu}^*_{t+H-1|t}, \hat{\mu}^*_{t+H|t}\} \tag{24}$$

at instant t. By appending the control signal that is produced by the feedback controller $K\hat{\mu}^*_{t+H|t}$ to (23), a suboptimal solution at the next time step $t+1$ is given by

$$\{c^*_{t|t}, c^*_{t+1|t}, \ldots, c^*_{t+H-1|t}, K\hat{\mu}^*_{t+H|t}\} \tag{25}$$

and

$$\{\hat{\mu}^*_{t|t}, \hat{\mu}^*_{t+1|t}, \ldots, \hat{\mu}^*_{t+H-1|t}, \hat{\mu}^*_{t+H|t}, A_K\hat{\mu}^*_{t+H|t}\}, \tag{26}$$

where $A_K := A + BK$ denotes the closed-loop transition matrix. Based on this suboptimal solution, we can prove that the optimal value function $V^*(\hat{\mu}_t)$ decreases along the trajectory. Since (25) and (26) are suboptimal, one has

$$\begin{aligned} V(\hat{\mu}_{t+1}) = &\sum_{k=1}^{H-1} q(\hat{\mu}^*_{t+k|t}, c^*_{t+k|t}) + q(\hat{\mu}^*_{t+H|t}, K\hat{\mu}^*_{t+H|t}) && (27)\\ &+p(A_K\hat{\mu}^*_{t+H|t}) && (28)\\ =&V^*(\hat{\mu}_t) + q(\hat{\mu}^*_{t+H|t}, K\hat{\mu}^*_{t+H|t}) + p(A_K\hat{\mu}^*_{t+H|t}) && (29)\\ &- q(\hat{\mu}^*_{t|t}, c^*_{t|t}) - p(\hat{\mu}^*_{t+H|t}), && (30) \end{aligned}$$

where $q(\hat{\mu}_t, c_t) = \|C\hat{\mu}_t\|^2_Q + \|c_t\|^2_R$ denotes the stage cost and $p(\hat{\mu}_t) = \|C\hat{\mu}_t\|^2_P$ denotes the terminal cost. Note that K is a stabilizing state feedback controller as designed by (15) and (16), thus it follows that

$$q(\hat{\mu}^*_{t+H|t}, K\hat{\mu}^*_{t+H|t}) + p(A_K\hat{\mu}^*_{t+H|t}) - p(\hat{\mu}^*_{t+H|t}) \leq 0 \tag{31}$$

Take into account the optimal value function $V(\hat{\mu}_{t+1}) < V^*(\hat{\mu}_{t+1})$, thus

$$V^*(\hat{\mu}_{t+1}) - V^*(\hat{\mu}_t) \leq -q(\hat{\mu}^*_{t|t}, c^*_{t|t}), \tag{32}$$

and the optimal value function $V^*(\cdot)$ is a valid Lyapunov function. Therefore, the expectation of the nominal state $\mathbb{E}\hat{x}_t = C\hat{\mu}_t$ converges to zero as $t \to \infty$; that is, the nominal state is mean square stable.

Second, let's consider the dynamics of the error system $e_t := \mu_t - \hat{\mu}_t$, which is defined by the difference between the (12) and the (8). By substituting the controller (13) into the error system, it follows that

$$e_{t+1} = (A + BK)e_t + g(w_t). \tag{33}$$

Iterate the dynamics of the error system (34) from the initial time instance 1 to t with $e_{t+1} = A_K g(w_t) + A_K^2 g(w_{t-1}) + \cdots + A_K^t g(w_1) + A_K^t e_1$. According to Algorithm 1, the initial instance e_1 equals zero, as $\hat{\mu}_1 = \mu_\theta(x_1)$. Then, the L_2 norm of the error state is given by

$$\begin{aligned}
\|e_{t+1}\| &= \|A_K g(w_t) + A_K^2 g(w_{t-1}) + \cdots + A_K^t g(w_1)\| && (34)\\
&\leq \|A_K g(w_t)\| + \|A_K^2 g(w_{t-1})\| + \cdots + \|A_K^t g(w_1)\| && (35)\\
&\leq \beta\|g(w_t)\| + \beta^2\|g(w_{t-1})\| + \cdots + \beta^t\|g(w_1)\|. && (36)
\end{aligned}$$

Taking the expectation over the random noise w_t, and using the fact that the random noise signal is independently distributed at different time instances, it follows that

$$\begin{aligned}
\mathbb{E}\|e_{t+1}\| \leq& \beta\mathbb{E}_{w_t}\|g(w_t)\| + \beta^2\mathbb{E}_{w_{t-1}}\|g(w_{t-1})\| + \dots && (37)\\
&+ \beta^t\mathbb{E}_{w_1}\|g(w_1)\|. && (38)
\end{aligned}$$

According to Assumption 1, we can further infer that

$$\begin{aligned}
\mathbb{E}\|e_{t+1}\| \leq& \beta L\mathbb{E}_{w_t}\|w_t\| + \beta^2 L\mathbb{E}_{w_{t-1}}\|w_{t-1}\| + \dots && (39)\\
&+ \beta^t L\mathbb{E}_{w_1}\|w_1\| && (40)\\
\leq& \beta Lb + \beta^2 Lb + \cdots + \beta^t Lb && (41)\\
=& \frac{(\beta - \beta^t)Lb}{1-\beta}, && (42)
\end{aligned}$$

where the second inequality is a direct result of Assumption 2. As $t \to \infty$, the expectation of the error state norm is bounded by $\frac{\beta Lb}{1-\beta}$. The state of the original system is given by $\mathbb{E}x_t = C(\hat{\mu}_t + \mathbb{E}e_t)$, thus the effect of the error state upon the original state is bounded by $\frac{\beta\sigma Lb}{1-\beta}$, where $\sigma := \|C\|$. Because the nominal state is mean square stable and the error between the actual and nominal state is bounded, the system (8) is proven to be uniformly ultimately bounded. □

Proof of Proposition 2

Proof. As the nominal system remains the same as in Proposition 1, the proof for the mean square stability of the nominal system is identical as well. We focus on proving the uniform ultimate boundedness of the error system.

In the presence of the approximation residuals, the dynamic of the error system $e_t := \mu_t - \hat{\mu}_t$is given as follows,

$$e_{t+1} = (A + BK)e_t + g(w_t) + \epsilon_t \tag{43}$$

Iterating the above equation (33) from the initial time instance 1 to t, one has

$$\begin{aligned} e_{t+1} =& A_K^t e_1 + \underbrace{A_K\epsilon_t + A_K^2\epsilon_{t-1} + \cdots + A_K^t\epsilon_1}_{\sum_1^t A_K^k\epsilon_k} \\ &+ \underbrace{A_K g(w_t) + A_K^2 g(w_{t-1}) + \cdots + A_K^t g(w_1)}_{\sum_1^t A_K^k g(w_k)}. \end{aligned} \tag{44}$$

According to Algorithm 1, the initial instance e_1 equals zero, as $\hat{\mu}_1 = \mu_\theta(x_1)$. Then the L_2 norm of the error state is given by

$$\|e_{t+1}\| = \|\sum_1^t A_K^k\epsilon_k + \sum_1^t A_K^k g(w_k)\| \tag{45}$$

$$\leq \sum_1^t \|A_K^k\epsilon_k\| + \sum_1^t \|A_K^k g(w_k)\| \tag{46}$$

$$\leq \sum_1^t \beta^k\|\epsilon_k\| + \sum_1^t \beta^k\|g(w_k)\|. \tag{47}$$

Taking the expectation over the random noise w_t, it follows that

$$\mathbb{E}\|e_{t+1}\| \leq \sum_1^t \beta^k\|\epsilon_k\| + \sum_1^t \beta^k\|g(w_k)\| \tag{48}$$

$$\leq \sum_1^t \beta^k\gamma + \sum_1^t \beta^k Lb \tag{49}$$

$$= \frac{(\beta - \beta^t)(Lb + \gamma)}{1 - \beta}, \tag{50}$$

where the second inequality is a direct result of Assumption 2 and Assumption 3. As $t \to \infty$, the expectation of the error state norm is bounded by $\frac{\beta Lb+\gamma}{1-\beta}$. Because the nominal state is mean square stable and the error between the actual and nominal state is bounded by a constant $\frac{\beta(Lb+\gamma)}{1-\beta}$, the system (4) is proven to be uniformly ultimately bounded. □

3-4. What are the computational complexities of the proposed DeSKO method and the three baseline methods, i.e., DKO, MLP and SAC?

We greatly appreciate the reviewer's important advice. We agree that the computational complexity of the method is an important aspect of performance. Thus we have evaluated the computational complexity of DeSKO and baseline methods on the Cartpole example and summarized the results in Section IV.D.

According to the test results, DeSKO takes 0.49ms for a 16 steps forward state prediction rollout, while MLP takes 1.33ms to do this. DKO is lightly faster than DeSKO because it has fewer parameters. In terms of control, SAC takes 0.26ms to calculate the control input, while DeSKO and DKO take 1.61ms and 1.56ms, respectively. This is because SAC generates the control input through one-step neural network mapping, while DeSKO and DKO rely on MPC and optimization to determine the control action. Nonetheless, both SAC, DeSKO, and DKO can all achieve a control frequency over 500 Hz and thus are applicable to real-time control. The following contents are added to the manuscript.

Page 8

Computational Efficiency

The computation efficiency/complexity of the algorithms is also an important aspect of the performance aspect. The computation efficiency of DeSKO in prediction and control on the Cartpole example was also evaluated and compared to baselines. We had the algorithms to predict future states (16 steps forward) in the test set and calculated their average time consumption. This test is not applicable to SAC because SAC is model-free. During the control evaluation, the algorithms' time consumption for each step was also recorded and averaged. All of the experiments were conducted on a computer with a 12th Gen Intel Core i7-12700 processor. As shown in Table 1, DeSKO and DKO performed similarly in terms of computation efficiency. MLP is the most computationally expensive method due to the iteration of neural network forward propagation. In terms of control, SAC showed the highest computation efficiency compared to the Koopman-based methods, because no optimization is needed at runtime.

Table 1: Computation time of different methods in milliseconds (ms)

Tasks	DeSKO	DKO	MLP	SAC
Forward prediction	0.49	0.42	1.33	N/A
Control	1.61	1.56	N/A	0.26

3-5. Please provide the values of the parameters used in the simulation and experiments.

Thanks for the reviewer's comment. The setup of the experiment and values of parameters were also included in the Appendix file, and was somehow not shown to the reviewer. In this submission, the experiment setup and parameter values are also included in the Appendix due to space limit, and we have uploaded the file to a GitHub repository: `https://github.com/hithmh/TNNLS-supplementary-materials`.

3-6. Please explain why DeSKO has more burrs and a wider confidence interval compared with the baseline method in modeling evaluation.

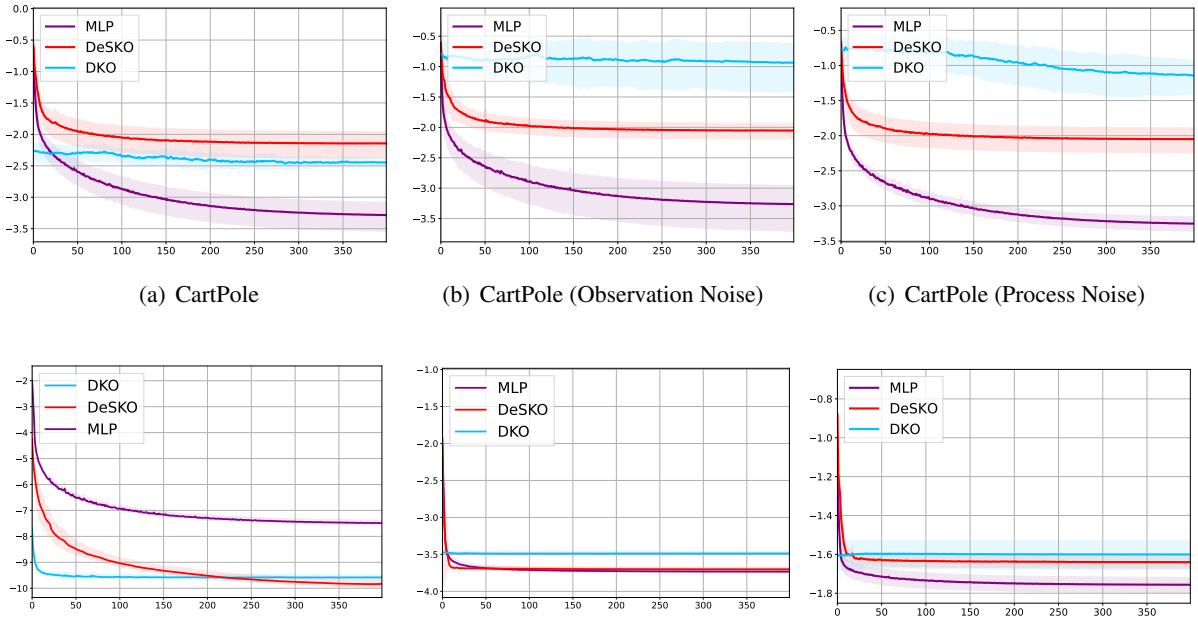
Thanks for the reviewer's question. We believe that the reviewer was majorly referring to the Fig.4 (d) and (e), and Fig.5 (c) and (d). In the previous evaluation, we used the same setting of hyperparameters

(including learning rates, latent space dimension, target entropy, etc) in training DeSKO models for different simulated systems. After checking the training logs of these tasks, we suspect that the burrs and wide confidence interval could be caused by the inappropriate setting of the learning rate (10^{-3}) and target entropy ($\mathcal{H} = -20$). Large learning rates would cause the optimization to take large update steps, and thus produce burrs. And if the target entropy is too large, the model would be more stochastic than necessary and thus generate a larger confidence interval.

Based on these insights, we fine-tuned the learning rate and target entropy for these environments and the results are shown in Fig.4 (d) and (e), and Fig.5 (c) and (d) in the revised manuscript. As shown in the figures, the training curves are more smooth and the confidence intervals are reduced. We have now included the above discussion in Section IV.B. The specific hyperparameter can be found in Appendix III.

Page 8

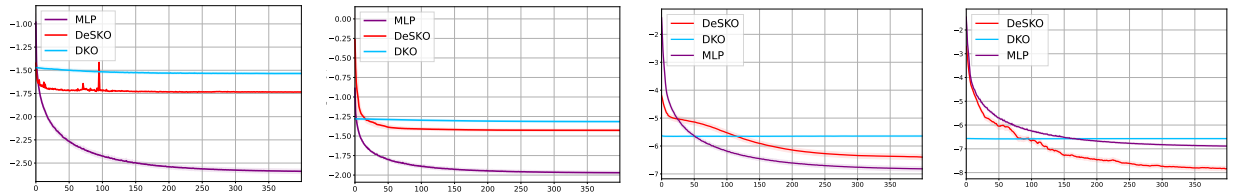
During testing, we also found that the training process of DeSKO can be unsteady with burrs and large confidence intervals. This is usually caused by the inappropriate setting of the learning rate and target entropy. Large learning rates could cause the optimization to take large update steps and thus produce burrs. And if the target entropy is too large, the model would be more stochastic than necessary and thus generate a larger confidence interval.



(a) CartPole (b) CartPole (Observation Noise) (c) CartPole (Process Noise)

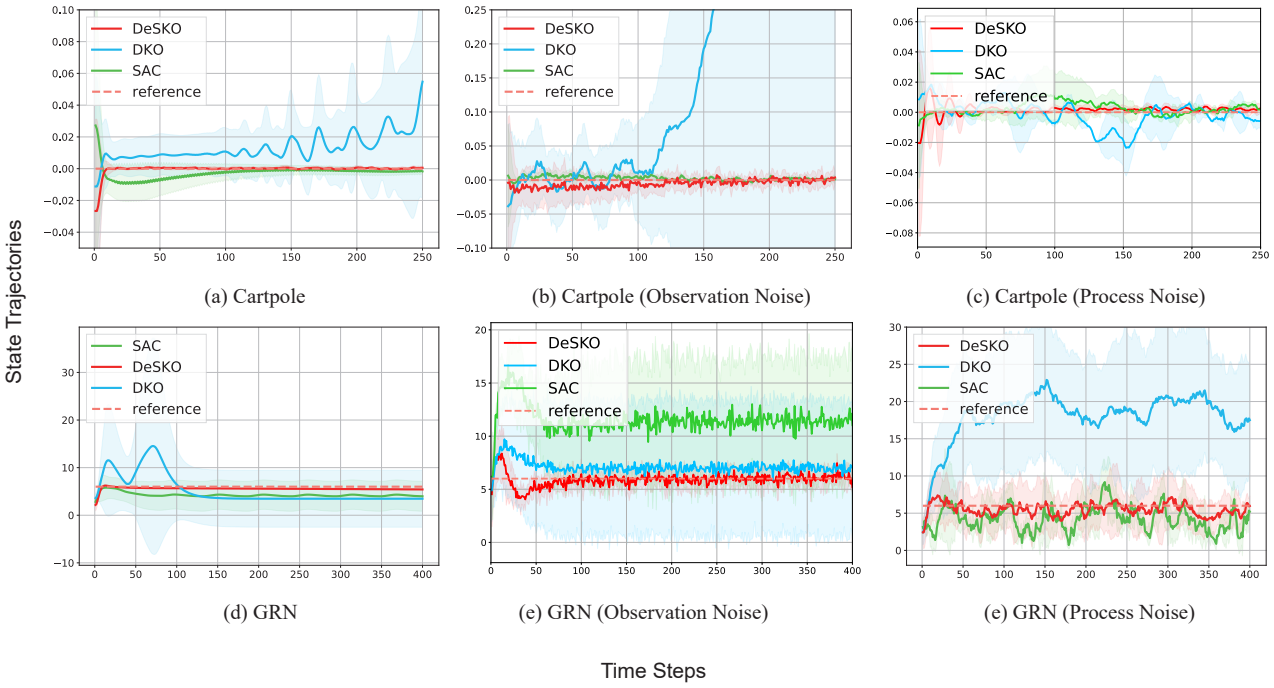
(d) GRN (e) GRN (Observation Noise) (f) GRN (Process Noise)

Figure 4: Cumulative prediction error on the validation set for CartPole and GRN. The Y-axis indicates the cumulative mean-squared prediction error in log space over 16 time steps, and the X-axis indicates the training time-steps. The shaded region shows the confidence interval (one standard deviation) over 10 random seeds.



(a) Halfcheetah (b) Halfcheetah (Observation Noise) (c) SoPrA (d) SoPrA (Observation Noise)

Figure 5: Cumulative prediction error on the validation set for Halfcheetah and SoPrA. The Y-axis indicates the cumulative mean-squared prediction error in log space over 16 time steps, and the X-axis indicates the training time-steps. The shaded region shows the confidence interval (one standard deviation) over 10 random seeds.



(a) Cartpole (b) Cartpole (Observation Noise) (c) Cartpole (Process Noise)

(d) GRN (e) GRN (Observation Noise) (e) GRN (Process Noise)

Figure 6: The state trajectories and reference signals of different systems. The y-axis indicates the average state trajectory, and the x-axis indicates the time steps in each episode. The shaded region shows the confidence interval (one standard deviation) over ten random seeds. We only show the dimensions in which a reference signal was tracked.

3-7. Why does the DeSKO method in the control evaluation have steady-state errors in Fig.6(c) and Fig.6(e)?

We appreciate the reviewer's valuable question. The steady-state errors were majorly caused by the inappropriate setting of control parameters, specifically the prediction horizon and weighting matrix. According to the reviewer's suggestion, we have fine-tuned these parameters by extending the prediction horizon of MPC and decreasing the values in the weighting matrix to increase the steady-state performance.

The improved results are shown in Fig.6 (c) and (e).

3-8. When grasping different objects, please describe the universal model.

Thanks for the reviewer's question. In this work, we train a deep stochastic Koopman operator shown in Fig. 2 to model the dynamics of the SoPrA arm and we apply control as shown in Fig. 3. The model parameters as used for Fig. 2 can be found in the Github repo `https://github.com/hithmh/TNNLS-supplementary-materials`. We have now indicated this GitHub location in the supplementary material.

3-9. When grasping different objects, please describe the process of interacting with the environment.

According to the reviewer's suggestion, we have described the interaction process in more detail as follows in Section V.E.

Page 12

The experiment consisted of two scenarios: 1) cube picking and 2) common objects picking. In Cube picking scenario, the process is summarized as follows:

(a) The system initializes and the 3D-printed cube is put at one of the test positions. Marker points were attached to the cube so the motion capture system can identify its position.

(b) Move the gripper to the detected cube position with set-point tracking MPC.

(c) Actuate the gripper to pick up the cube.

(d) Move to the dropping position and release the cube.

Key frames of the process have been marked in Fig 18.

In the common objects picking scenario, the process is summarized as follows:

(a) The system initializes and one of the objects is put at a certain test position.

(b) Move the gripper to the test position with set-point tracking MPC.

(c) Actuate the gripper to pick up the object.

(d) Move to the dropping position and release the object.

We have also recorded a video that shows a clear picture of the process and can be found in the supplementary materials.

3-10. Please describe the success rate of grasping.

Thanks for the reviewer's suggestion, and we have included the discussion on the success rate of grasping as follows in Section V.E.

Page 14

The success rate of grasping is affected by the precision of control, as well as the design of the motion planning module and the gripper structure. In this work, we focus on achieving precise control with the proposed framework. Based on the size of the gripper and objects, we regard the steady state tracking error under 2 cm at the picking instant (4s) as a successful grasping. Thus in the cube picking scenario, the success rate of MPC is 22.2% and MPC with the integral controller is 66.7%. In the common objects grasping scenario, MPC would fail to grasp all of the objects, while MPC with integral control achieves 100% success rate.

Final Remarks

Finally, we would like to thank all the reviewers again for reviewing our paper. Thanks to the help of the reviewers, we can further improve the paper comprehensively. We wish the revised paper meets the criteria to be published in IEEE Transactions on Neural Networks and Learning Systems.