2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Comparison of Model-Based and Model-Free Reinforcement Learning for Real-World Dexterous Robotic Manipulation Tasks

David Valencia, John Jia, Raymond Li, Alex Hayashi, Megan Lecchi, Reuel Terezakis,
Trevor Gee, Minas Liarokapis, Bruce A. MacDonald, Henry Williams

***Abstract*— Model Free Reinforcement Learning (MFRL) has shown significant promise for learning dexterous robotic manipulation tasks, at least in simulation. However, the high number of samples, as well as the long training times, prevent MFRL from scaling to complex real-world tasks. Model-Based Reinforcement Learning (MBRL) emerges as a potential solution that, in theory, can improve the data efficiency of MFRL approaches. This could drastically reduce the training time of MFRL, and increase the application of RL for real-world robotic tasks. This article presents a study on the feasibility of using the state-of-the-art MBRL to improve the training time for two real-world dexterous manipulation tasks. The evaluation is conducted on a real low-cost robot gripper where the predictive model and the control policy are learned from scratch. The results indicate that MBRL is capable of learning accurate models of the world, but does not show clear improvements in learning the control policy in the real world as prior literature suggests should be expected.**

I. INTRODUCTION

Researchers have tried to emulate human skills and learning capabilities to create more effective robots for decades. Dexterous manipulation is desirable to enable effective interaction with objects in complex environments. Despite an extensive investigation, human-level manipulation by robots is still not yet achieved. Dexterous manipulation in real-world applications involves changing an object's position and/or orientation in complex and unstructured environments [5], [49]. Traditional solutions rely on detailed kinematics and physics models to generate complex motion plans to control the grippers [6]. The drawback is that these approaches cannot adapt to novel environments or refine their strategy while executing. They rely heavily on highly structured environments that are very sensitively calibrated [7]. These problems are further exacerbated as the degrees of freedom increase by adding more manipulators to the gripper. Therefore it is necessary to present new methodologies that facilitate rapid adaptation to environmental changes without the need for tedious manual interventions.

Reinforcement Learning (RL) has played an essential role in this effort. Unlike traditional AI methods, such as unsupervised and supervised learning, RL does not require pre-labelled data or an extensive training dataset to discover a solution for a new problem. Instead, it derives a suitable solution via interactions with the environment [52]. This way of learning has made RL popular in the robotics community and other areas, emerging as a potential solution for increasing the adaptability of robotics systems in diverse scenarios. Much research has been carried out in this area of AI, where several survey papers have been introduced that demonstrate the potential of RL [9], [31], [43], [12], [45]. Examples of the success of RL include autonomous soccer-playing robots [44], inverted helicopter flight controllers [36], or learning to play board games such as Go on a professional level [46].

RL has also been applied extensively to dexterous manipulation, bypassing the need for manually designed controllers or using fixed handcrafted models. Simulation and real-physical hardware proposals employing RL with many multi-fingered robotics hands have been presented for solving complex manipulation tasks [49]. However, the primary challenge of deploying RL at scale for physical systems is the significant training time required to learn the control policies.

MBRL has been proposed as a means of reducing the training time and improving the efficiency of the learning process [32]. The use and performance of MBRL to improve the learning time for robotic tasks have been explored in recent years [3], [25], [10], [34], [38], [39], however, the work is primarily limited to simulated tasks.

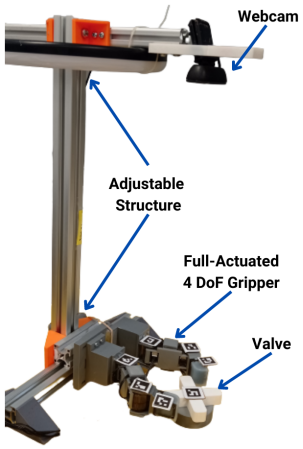


Fig. 1: The robotic testbed used for the experiments consists of an adjustable base structure, a fully-actuated robotic gripper, a webcam for recording the pose of the object and the joints, and a 3D printed valve-like object.

D. Valencia, J. Jia, R. Li, T. Gee, B. MacDonald and H. Williams are with the Centre for Automation and Robotic Engineering Science, The University of Auckland, New Zealand. Emails: {dval035,cjia881,rli948}@aucklanduni.ac.nz, {t.gee,b.macdonald,henry.williams}@auckland.ac.nz

A, Hayashi, M. Lecchi, R. Terezakis, and M. Liarokapis are with the New Dexterity Research Group, The University of Auckland, New Zealand. Emails: {ahay068,mlec922,rter148}@aucklanduni.ac.nz, minas.liarokapis@auckland.ac.nz

This paper presents a comprehensive comparative analysis of the state-of-the-art in MFRL and MBRL on a dexterous robotic manipulation task with a real robot. The complete learning procedure and testing experiments are carried out on an open-source robotic manipulation test-bed platform shown in Figure 1.

II. RELATED WORK

Even with the vast spectrum of "traditional engineering"[1] proposals indicating favourable results on dexterous manipulation in both simulations and real hardware [13], [1], [4], [33], [27], [26], the increased complexity in modelling modern robots, make dexterous manipulation impractical and computationally expensive to achieve with traditional or manually designed controllers [40]. RL presents a means of learning complex control policies without human-engineered solutions. RL falls into two categories: MFRL, and MBRL. The most significant difference between these two classes lies in whether or not a predictive model of the environment is known [47].

In MFRL, as its name implies, there is no explicitly defined predictive model. MFRL methods directly learn a policy by interacting with the environment and mapping from states to actions. MFRL algorithms and their applications have gained significant attention in research. While MFRL has shown significant progress, its efficiency cost is an essential point to consider. Real-world samples are costly to obtain due to hardware vulnerability and time restrictions; the more fragile a robotic system, the more valuable data-efficient methods should be [11]. MFRL is expensive in terms of sample complexity needing many interactions to learn a task; this may limit its application primarily to simulated domains. Although some MFRL approaches have demonstrated a significant reduction in the number of samples needed in the training process [15], there are still several limitations associated with hardware constraints, exploration strategies or the cost of obtaining new samples, making MFRL replication difficult in real-world domains.

MBRL, on the other hand, uses a predictive model (known or learned) to learn a global policy or value function to act in the environment [32], reducing the number of real-world interactions required and consequently learning faster than MFRL. However, there is a significant challenge for MBRL, model accuracy. The learned model might not be accurate enough; consequently, the control or policy learning is performed against an imperfect model [25]. Current literature defines this problem as model-bias [30]. This could be a bottleneck deteriorating the performance of the policy, creating a limitation on MBRL methods to perform worse or converge to less optimal solutions than their model-free counterparts [48], [20]. Table I summarises the primary pros and cons of MBRL and MFRL.

A. Model-Free Dexterous Manipulation Approaches

Numerous proposals have been presented utilizing MFRL for dexterous robotic manipulation tasks. Work has been explored in simulation [28], [40], [2] and real-world [16], [50], [53], [18], with real-world performance trailing the complexity of the simulated tasks.

The MFRL method presented in [40] uses a 9-DoF simulated robot (6-DoF for joints of an arm and 3-DoF for a gripper) to pick up lego bricks and stack them vertically. The authors solve the task in 10M transitions by extending DDPG[28] and parallelising the learning process among multiple machines (16 simulated robots). Although impressive, this work requires extensive training time across numerous robot platforms and has only been conducted in simulation.

A real low-cost multi-fingered hand is presented in [53], where MFRL is scaled up to learn a combination of manipulation tasks (rotating an object, box flipping and door opening) using a low-dimension vector state representation. Seven to 16 hours of constant training time is required to solve the tasks. The work is impressive for having been conducted in the real world; however, the training time is relatively long given the low complexity of the tasks at hand compared to similar work in simulation.

Attempts to bypass training in the real world by transferring from simulation to the real world have been attempted [51], [2]. [2] presents a transfer learning technique where a multi-finger robot hand is trained entirely in simulation to solve an in-hand object reorientation task; the policy is transferred later to the physical robot. Even when the task was completed in both scenarios, the result indicates that the simulation performed better than the real robot. More than 50 hours and an expensive (16 GPU) setup were required limiting the practicality of scaling this approach. Furthermore, training in simulation may create a "simulation to reality gap" where the trained model does not perform as well in real life as it does in simulation due to the limitations in the complexity of the simulated environment [22]. Without significantly more complex simulations, the success of transferring learning is unlikely to resolve the problem entirely.

Outside of the number of interactions required to learn the tasks is the time that is required to reset the training process. [17] propose a reset-free reinforcement learning algorithm where failing one task starts training on the next. The authors argue that the combinations of multi-task running sequentially help to solve the tasks intrinsically and reduce the time required to train in the process. More than 60 hours of real-time was needed to reach a 70% success rate on four separate but linked tasks. This reset method is clever, but such linked reset conditions are not always feasible.

The proposals cited above show that learning policies without modelling the robot system still dominate RL. In other words, MFRL is still a prevalent option in robotics applications because of its simplicity in implementation. However, the sampling efficiency remains daunting. When dealing with complex real-world physical systems in unstructured environments, several MFRL algorithms could not be used since they might be prohibitively costly or even impracticable in terms of time and resource usage to acquire data for policy training. Training a real robot to learn a task

[1]Human Engineered solutions

TABLE I: Model-Base versus Model-Free, advantages and disadvantages

	MBRL	MFRL
Advantages	• Small the number of necessary interactions • Ability to generated new information from a specific task to learn a new task • Minimize the risk of damage and wear on the robot	• Easy and cheap implementation • Does not explicitly require a model of the system
Disadvantages	• Model-bias (Heavy dependence on model accuracy) • A perfect probabilistic model is infeasible to obtain. • Computationally Expensive	• Large number of interactions needed • High risk of damaging the robot • Mostly useful only in simulated domains

with MFRL methods may require a large number of time steps, the equivalent of several hours or days of training in real-time [21], which is unsuitable for most applications.

B. Model-Based Dexterous Manipulation Approaches

MBRL approaches can be considerably more data efficient [47] but have not yet been scaled up to the same complex level as MFRL. Nevertheless, the related literature also reveals applications using MBRL for dexterity applications.

[41] presents SOIL, an imitation learning proposal for dexterous manipulation using a state-only method. A 24-DoF multi-finger hand is simulated and trained to solve object relocation, in-hand manipulation, door opening, and tool use tasks. An inverse dynamic model is learned using self-supervision and human demonstration experiences to infer the action. The policy (NPG method) and the inverse model are trained jointly. Even when a comparative analysis with related proposals is presented, there is no evidence to show if this proposal can be better (in terms of sample efficiency) than the latest MFRL algorithms. Another MBRL approach combining SAC and Model-Predictive Control (MPC) is presented in [37]. The training process is divided into two phases. First, the learning phase updates the policy along with a dynamics model that predicts the next state. The second stage finds the best actions employing the learned model and imaginary trajectories generated by the MPC. These two phases are trained individually. Two simulated-only tasks (valve-turning and cube manipulation) are solved and compared against state-of-the-art MFRL approaches; the results indicate a slight superiority of this MBRL proposal.

Employing a real and a simulated pneumatically-actuated 24-DoF hand, an MBRL proposal is presented in [24]. The authors used a 100-dimensional continuous state space (including the positions and velocities of the joints) with local linear models to learn a time-varying linear-Gaussian controller. The actions correspond to the pneumatic valve's input voltage. Two tasks are implemented; reaching a target pose and manipulating a freely-moving cylindrical object equipped with active infrared markers. The environment reset, i.e. repositioning the cylinder, has to be manual. Not much information about the training process, reward function or comparative analysis in terms of data sample needed is mentioned. Similarly, [35] presents a method using the same 24-DoF Shadow Hand for rotating two spheres using PDDM-MBRL. The goal is to rotate the two spheres around the robot's palm. This is a remarkably complex task since two objects must share the same workspace without being dropped. A parameterised gaussian distribution with deep neural networks is used for dynamic model learning. The authors solved this task without simulation or prior knowledge after two hours of real-world training. The evaluation is thorough and demonstrates improved performance over MFRL and related MBRL methods; in some cases solving tasks other approaches are unable to. However, the control policy utilised Model Predictive Control and was not a learned control policy [29]. This work demonstrates the ability of MBRL to learn the dynamics of the real world and is thus potentially suitable for use with MFRL approaches.

Even when the related literature shows an advance in the MBRL area where complex tasks have been solved, most of these have been in simulated environments or expensive setups that are difficult to scale in terms of cost. There is also a general lack of a direct analysis against MFRL proposals with real robots.

III. Contributions

The previous literary review is just a sample of the evolution in dexterity in robotics using RL (see [23], [49], and [19] for an in-depth overview), where the predominance of MFRL in simulations is mainly evident. With few exceptions, the majority of these implementations are unlikely to be transferred to real-world scenarios due to the long training cycles that MFRL demands and the sparse information available in the real world compared to simulated environments. Concerning the implementation using real robots, the literature shows that results are promising; however, the use of sophisticated multi-fingered robotic hands (often fragile and expensive) coupled with complex video tracking systems and data sensing issues still keeps dexterity manipulation with RL at a laboratory level. Using RL to solve complex tasks in real psychical systems is essential. The solution, however, should be: inexpensive in terms of sample complexity, with low-cost setups allowing easy replicability but robust enough to work in non-laboratory conditions.

There is an evident disconnect between the challenges and limitations solved in simulation versus in real-world dexterity applications. MBRL has been studied and tested in multiple applications in simulations and expensive environments and has been presented as a possible solution to overcome MFRL's limitations. But are the current MBRL proposals the solution that will help scale MFRL in the real world? This article tries to answer this question by comparing the state-of-the-art MFRL and MBRL under the same training conditions using a low-cost open-source robot gripper.

IV. METHODOLOGY

Our proposal could be seen as an extension of the original Dyna algorithm presented in [47], where a dynamic model is learned to generate new samples and update a policy using RL methods. However, using this idea with MBRL is not necessarily straightforward. Deep MBRL algorithms tend to overfit and be particularly unstable during training if insufficient data samples exist. Likewise, the non-deterministic essence of real-world scenarios makes it necessary to use more suitable learning techniques to decrease model-bias issues and have reliable predictive models of high-dimensional real-world systems that traditional deterministic models may struggle with.

Inspired by [20], [10] and [25], we present an MBRL algorithm that uses an ensemble of probabilistic networks to learn a predictive model. In this work, however, we combine the ensemble of probabilistic neural networks with a Mixture of Gaussian Distributions [8]. This simple but effective combination helps to improve the model predictions and generated better sample experiences overcoming the non-deterministic nature of the real world, and maintaining model uncertainty.

Figure 2 illustrates the structure of the model prediction. A state and action pair (collected from the environment) is passed as input to an ensemble of probabilistic neural networks. Each ensemble member outputs a set of n parametrised Gaussian distributions defined by $f = (S_{t+1}|S_t, a_t) = N(\mu(S_t, a_t), \sigma(S_t, a_t))$. Then a Gaussian mixture function is generated for each model. We average the output of each mixture to generate a prediction. A negative log-likelyhood is used as a loss function.

Our proposal shares multiple aspects already mentioned in [20], [10]; we simplify these proposals in some aspects. For instance, the authors in [20] implemented branched rollouts with k-steps predictions. However, we believe this may produce issues with the learning policy, especially in initial states where the predictive model is not yet accurate. Therefore, we use a single-step prediction. During the training phase, the policy model is trained on experiences from the environment for the first w episodes (set to 10% of the total episodes) and then from the experiences generated from the predictive model. This helps the training, especially in the early stages when the world model predictions are less accurate.

Furthermore, related literature [10], [42] uses the predictive model for horizon planning and predictive control strategies, whereas we use the predictive model for policy learning using the data generated by the model. Algorithm 1 shows the methodology flow in this proposal. We employ TD3 [14] as our policy update method. The value of each hyperparameter, the source-code, hardware specifications, and a video demonstrating the learning process can be found here[2]

[2]https://cares.blogs.auckland.ac.nz/research/reinforcementlearning/mbrl/

Algorithm 1 Model-Based Reinforcement Algorithm

```
Require: Data-sets D_env, D_model, Initialize policy π_φ and predictive model p_θ
for T Episodes do
    for H Steps do
        Take action in Env according to initial policy π_φ
        Add experience to D_env
        for M Rollouts do
            Sample S_t uniformly from D_env
            Using π_φ and p_θ generate single step rollouts
            Add experience to D_model
        end for
        for G gradient updates do
            if Episode ≤ w then
                Update policy parameters on env data:
                φ ← φ − ∇_φ J_π(φ, D_env)
            else
                Update policy parameters on model data:
                φ ← φ − ∇_φ J_π(φ, D_model)
            end if
        end for
    end for
end for
```

V. EXPERIMENTS

The dexterous tasks presented involve training the 4-DoF robot gripper to manipulate a real object. Two tasks have been devised to test RL control. The first task aims to demonstrate the ability to translate an object (cube), while the second task tries to demonstrate the ability to rotate an object (valve).

The task is to learn to manipulate a cube to a given point in a 2D area. The translation task is considered solved if the gripper moves the cube to within 10 millimetres of the goal point. The goal point is randomly chosen among all possible values in a specified area that the robot can physically reach (see Figure 3a). Considering the agent has no previous knowledge of the task or the environment, this task is challenging to solve with a real robot in a continuous action space since the robot must learn to locate, hold, and move the cube. After each time step, the cube is reset to an initial position.

The second task requires turning a valve-like object (resembling a gas valve) to an arbitrary target orientation (see 3b). This task is significantly demanding to learn since it involves visual perception (valve's current and desired orientation) and physical coordination (where the robot must learn to coordinate the two arms' movements to avoid the rotational movement's cancellation). Also, a complex finger gait is required to make continuous rotations but stop at the desired angle. The desired location is changed randomly for each episode, whereas the valve orientation is not reset. This adds an extra level of complexity to the task since the policy must learn to perceive the current valve orientation and rotate it to any possible position between $0-360$ degrees.

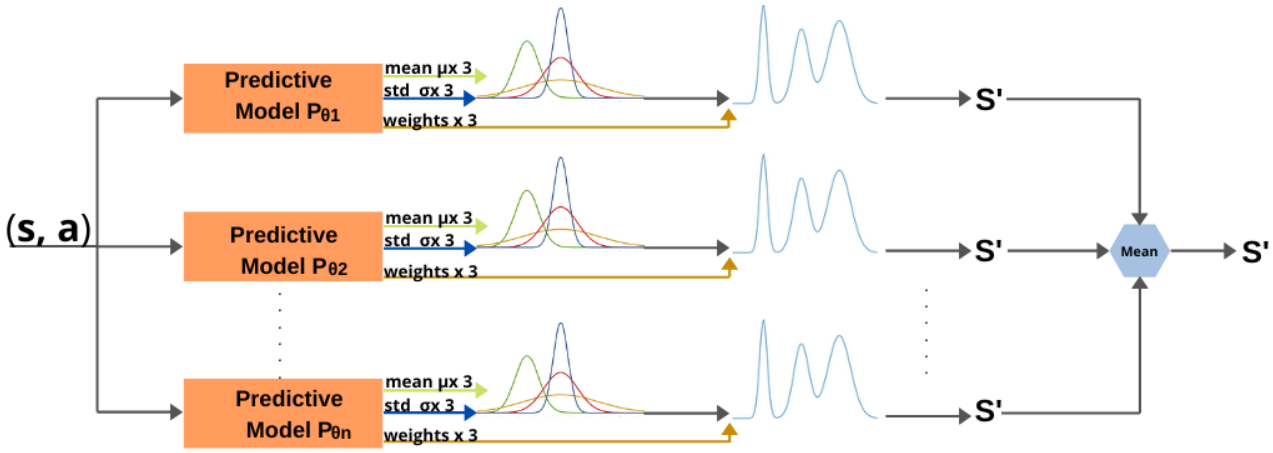
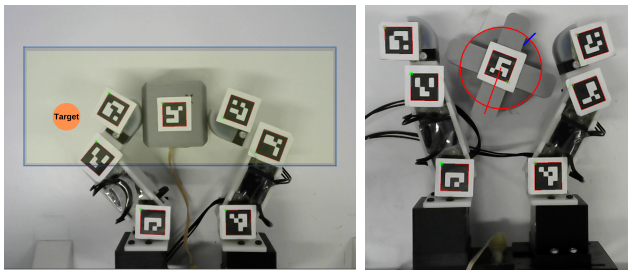


Fig. 2: Deep Model-Based Reinforcement Learning algorithm that uses an ensemble of probabilistic networks to learn a predictive model.

Previous related implementations rotate a valve to a set of fixed positions [18]. The state-space representation for both tasks is a 1D vector that includes the joint position, object position, object orientation and desired goal location/angle. The relative position of each joint, as well as the position of the object, is determined by employing individual Aruco markers and the low-cost webcam.

We use a continuous-action space representation where the action space controls the joint angles. The action space outputs a four-element vector with the desired position of each servo. This representation allows all four servos to move simultaneously at each time step. The action space has no direct influence on the manipulated object. A dense reward function is used for both tasks. The reward function is the negative $L2 - norm$ distance between the current cube's position and the goal point or the absolute difference between the valve's current location and the desired angle. If the task is completed, an additional reward of +500 is added.



(a) Translation Task (b) Rotation Task

Fig. 3: Setup of the two robotic dexterous manipulation tasks

VI. Results

One of the main problems with MBRL is the dependency on the model. As mentioned above and based on the theory that supports the idea of MBRL, the predictions have to be reasonably precise. Table II and III show our results on the model predictions of the state space representation. This was measured using data samples outside of the training set, gathered by running the algorithm through 100 random steps of the task. The results show that the model is able to accurately predict the next state for each task.

TABLE II: World Model state accuracy for the translation task (millimetres)

Element	Real Value Sample	Prediction Value Sample	Average Error on 100 predictions
Joint 1 Finger 1	(14.13, 57.88)	(13.95, 57.49)	(0.02, 0.04)
Joint 1 Finger 2	(-47.71, 58.51)	(-48.44, 58.92)	(0.07, 0.1)
Joint 2 Finger 1	(49.64, 5.90)	(48.79, 2.65)	(0.97, 2.39)
Joint 2 Finger 2	(-84.37, 5.76)	(-79.76, 6.28)	(1.87, 0.75)
Fingertip Finger1	(68.24, -6.23)	(67.42, -5.02)	(0.9. 1.45)
Fingertip Finger2	(-88.63, -15.85)	(-84.89, -13.85)	(2.74, 1.85)
Cube Position	(18.26 -14.09)	(18.01, -13.3)	(1.07, 2.12)

The average reward for the MFRL and MBRL approaches during the learning process for the translation and rotation tasks are shown in Figures 4 and 5 where the superiority of MFRL against MBRL is evident. The expected improvement in the training iterations is not evident in the training results. MBRL follows a similar trend to the MFRL on the translation task but diverges as training continues while failing to improve the reward at all on the rotation task. The success rate for each task is shown in TableIV and also shows that the performance of MFRL exceeds the MBRL approach, with the MBRL failing to even learn the rotation task. These results are unexpected given the accuracy of the learn model in Tables II and III.

VII. DISCUSSION

The results we have obtained demonstrate that MFRL is capable of sufficiently learning the translation and rotation task. The MFRL training is completed in approximately

TABLE III: World Model state accuracy for the valve task (millimetres)

Element	Real Value Sample	Prediction Value Sample	Average Error on 100 predictions
Joint 1 Finger 1	(14.23, 58.43)	(13.79, 56.12)	(0.13, 0.09)
Joint 1 Finger 2	(-48.05, 59.03)	(-48.23, 57.98)	(0.15, 0.32)
Joint 2 Finger 1	(89.95, 15.37)	(87.23, 12.78)	(0.82, 1.23)
Joint 2 Finger 2	(-29.04, -12.36)	(-30.97, -10.74)	(1.25, 2.23)
Fingertip Finger1	(45.26, 17.23)	(44.45, 16.85)	(1.98, 2.36)
Fingertip Finger2	(56.21, 85.23)	(56.01, 84.06)	(2.87, 3.98)
Valve Orientation	(-45.23°)	(-42.02°)	(3.41°)

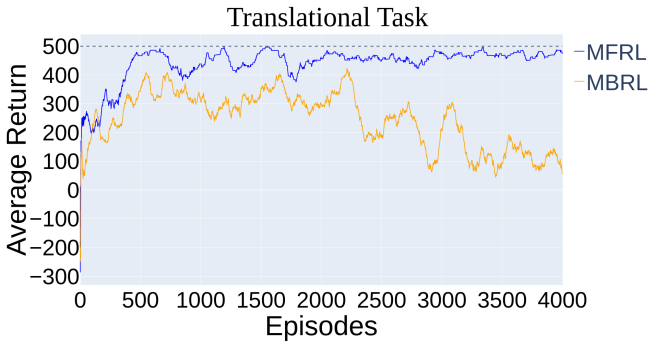


Fig. 4: Average return plotted against the number of episodes for the translational task

five hours for the translation task and 20 hours for the rotation task. However, the introduction of the MBRL to provide artificial experiences to reduce training time has not been observed even with results indicating an accurate world model. The results presented by prior literature show a clear superiority of MBRL against state-of-the-art MFRL methods; the number of interactions, data samples and training time is significantly reduced [10]. However, it is essential to mention that the experiments are primarily developed in simulated environments with optimal conditions, constant data samples, and straightforward testing setups. Working in real-world systems, factors such as noise in the input signals, friction, vibrations, modifications in the control signals, wear of mechanical parts, or even slight variations in the voltage input, add challenges for MBRL to learn. Although work in [35] demonstrated the ability to learn a model of a complex real-world task, the control policy was not dependent on it for any form of training.

Our opinion about the results we obtained is mainly associated with the exploration stage. The exploration stage plays a fundamental role in producing samples for the MFRL algorithm to learn from and producing samples to train the world model. If this stage is not broad enough and does not cover a wide spectrum of experiences (including

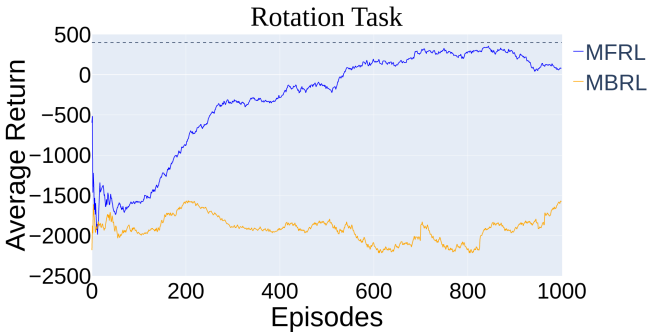


Fig. 5: Average return plotted against the number of episodes for the rotational task

TABLE IV: Success Rate of MBRL vs MFRL over 100 testing episodes

Task	MFRL	MBRL
Translational Task	99	71
Rotation Task	80	0

experiences that give high rewards), the MFRL approaches will overfit to a subset of the problem space. The inaccuracies in the data generated by the world model cause divergence in the control policy. Similarly, the world model does not become accurate and useful for producing experiences until it has sufficiently explored the environment. This creates a contradiction against the main idea behind MBRL, which is to reduce the number of physical interactions during training.

The MBRL approach fails to learn the valve task while still producing reasonable performance on the translation task. Conceptually, the translation task will provide diverse experiences through random exploration, indirectly pushing the cube around. In contrast, it is unlikely the gripper through random exploration will reliably rotate the valve to provide a full range of experiences to learn from. This limitation in experience degrades the overall performance of the MFRL as the MBRL lacks sufficient data. Therefore it is important to fnd better exploration methods or means of incorporating world model data into the learning process.

VIII. CONCLUSIONS

MBRL has shown incredible results in recent literature, and its capability to bring RL into the real world is promising; however, there is still a large gap to overcome for transitioning to the real world. Our results indicate that some current proposals may not be feasible for training MBRL methods on real-world tasks as smoothly as MFRL. Consequently, further developments must be included to enable RL algorithms to solve more complex real-world applications with adequate training time and the number of data samples.

ACKNOWLEDGEMENTS

This research was partially supported by the New Zealand Ministry for Business, Innovation and Employment (MBIE) on contract UOAX1810.

REFERENCES

[1] Sheldon Andrews and Paul G Kry. Goal directed multi-finger manipulation: Control policies and analysis. *Computers & Graphics*, 37(7):830–839, 2013.

[2] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[3] Christopher G Atkeson and Juan Carlos Santamaria. A comparison of direct and model-based reinforcement learning. In *Proceedings of international conference on robotics and automation*, volume 4, pages 3557–3564. IEEE, 1997.

[4] Yunfei Bai and C Karen Liu. Dexterous manipulation using both palm and fingers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1560–1565. IEEE, 2014.

[5] Antonio Bicchi. On the closure properties of robotic grasping. *The International Journal of Robotics Research*, 14(4):319–334, 1995.

[6] Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: A review. In *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, volume 1, pages 348–353. IEEE, 2000.

[7] Aude Billard and Danica Kragic. Trends and challenges in robot manipulation. *Science*, 364(6446):eaat8414, 2019.

[8] Christopher M Bishop. Mixture density networks. 1994.

[9] Sinan Çalışır and Meltem Kurt Pehlivanoğlu. Model-free reinforcement learning algorithms: A survey. In *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE, 2019.

[10] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

[11] Marc Peter Deisenroth, Carl Edward Rasmussen, and Dieter Fox. Learning to control a low-cost manipulator using data-efficient reinforcement learning. In *Robotics: Science and Systems VII*, volume 7, pages 57–64, 2011.

[12] Vektor Dewanto, George Dunn, Ali Eshragh, Marcus Gallagher, and Fred Roosta. Average-reward model-free reinforcement learning: a systematic review and literature mapping. *arXiv preprint arXiv:2010.08920*, 2020.

[13] Mehmet R Dogar and Siddhartha S Srinivasa. Push-grasping with dexterous hands: Mechanics and a method. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2123–2130. IEEE, 2010.

[14] 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.

[15] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3389–3396. IEEE, 2017.

[16] Abhishek Gupta, Clemens Eppner, Sergey Levine, and Pieter Abbeel. Learning dexterous manipulation for a soft robotic hand from human demonstrations. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3786–3793. IEEE, 2016.

[17] Abhishek Gupta, Justin Yu, Tony Z Zhao, Vikash Kumar, Aaron Rovinsky, Kelvin Xu, Thomas Devlin, and Sergey Levine. Reset-free reinforcement learning via multi-task learning: Learning dexterous manipulation behaviors without human intervention. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6664–6671. IEEE, 2021.

[18] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[19] Jiang Hua, Liangcai Zeng, Gongfa Li, and Zhaojie Ju. Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning. *Sensors*, 21(4):1278, 2021.

[20] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *arXiv preprint arXiv:1906.08253*, 2019.

[21] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

[22] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[23] Kilian Kleeberger, Richard Bormann, Werner Kraus, and Marco F Huber. A survey on learning-based robotic grasping. *Current Robotics Reports*, 1(4):239–249, 2020.

[24] Vikash Kumar, Emanuel Todorov, and Sergey Levine. Optimal control with learned local models: Application to dexterous manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 378–383. IEEE, 2016.

[25] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. *arXiv preprint arXiv:1802.10592*, 2018.

[26] Minas Liarokapis and Aaron M Dollar. Combining analytical modeling and learning to simplify dexterous manipulation with adaptive robot hands. *IEEE Transactions on Automation Science and Engineering*, 16(3):1361–1372, 2018.

[27] Minas V Liarokapis and Aaron M Dollar. Learning task-specific models for dexterous, in-hand manipulation with simple, adaptive robot hands. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2534–2541. IEEE, 2016.

[28] 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.

[29] Yuan Lin, John McPhee, and Nasser L Azad. Comparison of deep reinforcement learning and model predictive control for adaptive cruise control. *IEEE Transactions on Intelligent Vehicles*, 6(2):221–231, 2020.

[30] 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.

[31] Yongshuai Liu, Avishai Halev, and Xin Liu. Policy learning with constraints in model-free reinforcement learning: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021.

[32] Thomas M Moerland, Joost Broekens, and Catholijn M Jonker. Model-based reinforcement learning: A survey. *arXiv preprint arXiv:2006.16712*, 2020.

[33] Igor Mordatch, Zoran Popović, and Emanuel Todorov. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 137–144, 2012.

[34] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7559–7566. IEEE, 2018.

[35] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112. PMLR, 2020.

[36] Andrew Y Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger, and Eric Liang. Autonomous inverted helicopter flight via reinforcement learning. In *Experimental robotics IX*, pages 363–372. Springer, 2006.

[37] Muhammad Omer, Rami Ahmed, Benjamin Rosman, and Sharief F Babikir. Model predictive-actor critic reinforcement learning for dexterous manipulation. In *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pages 1–6. IEEE, 2021.

[38] Aske Plaat, Walter Kosters, and Mike Preuss. Model-based deep reinforcement learning for high-dimensional problems, a survey. *arXiv preprint arXiv:2008.05598*, 2020.

[39] Aske Plaat, Walter Kosters, and Mike Preuss. High-accuracy model-based reinforcement learning, a survey. *arXiv preprint arXiv:2107.08241*, 2021.

[40] Ivaylo Popov, Nicolas Heess, Timothy Lillicrap, Roland Hafner, Gabriel Barth-Maron, Matej Vecerik, Thomas Lampe, Yuval Tassa, Tom Erez, and Martin Riedmiller. Data-efficient deep reinforcement

learning for dexterous manipulation. *arXiv preprint arXiv:1704.03073*, 2017.

[41] Ilija Radosavovic, Xiaolong Wang, Lerrel Pinto, and Jitendra Malik. State-only imitation learning for dexterous manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7865–7871. IEEE, 2021.

[42] Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. Epopt: Learning robust neural network policies using model ensembles. *arXiv preprint arXiv:1610.01283*, 2016.

[43] Jorge Ramírez, Wen Yu, and Adolfo Perrusquía. Model-free reinforcement learning from expert demonstrations: a survey. *Artificial Intelligence Review*, pages 1–29, 2021.

[44] Martin Riedmiller, Thomas Gabel, Roland Hafner, and Sascha Lange. Reinforcement learning for robot soccer. *Autonomous Robots*, 27(1):55–73, 2009.

[45] Yoshiharu Sato. Model-free reinforcement learning for financial portfolios: a brief survey. *arXiv preprint arXiv:1904.04973*, 2019.

[46] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

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

[48] Thomas George Thuruthel, Egidio Falotico, Federico Renda, and Cecilia Laschi. Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators. *IEEE Transactions on Robotics*, 35(1):124–134, 2018.

[49] Chunmiao Yu and Peng Wang. Dexterous manipulation for multi-fingered robotic hands with reinforcement learning: A review. *Frontiers in Neurorobotics*, 16, 2022.

[50] Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4238–4245. IEEE, 2018.

[51] Kaichena Zhang, Farzad Niroui, Maurizio Ficocelli, and Goldie Nejat. Robot navigation of environments with unknown rough terrain using deep reinforcement learning. In *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–7. IEEE, 2018.

[52] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744, 2020.

[53] Henry Zhu, Abhishek Gupta, Aravind Rajeswaran, Sergey Levine, and Vikash Kumar. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3651–3657. IEEE, 2019.