Belief Regulated Dual Propagation Nets for Learning Action Effects on Groups of Articulated Objects

Ahmet E. Tekden[1], Aykut Erdem[2], Erkut Erdem[2], Mert Imre[1], M. Yunus Seker[1] and Emre Ugur[1]

***Abstract*— In recent years, graph neural networks have been successfully applied for learning the dynamics of complex and partially observable physical systems. However, their use in the robotics domain is, to date, still limited. In this paper, we introduce *Belief Regulated Dual Propagation Networks (BRDPN)*, a general purpose learnable physics engine, which enables a robot to predict the effects of its actions in scenes containing groups of articulated multi-part objects. Specifically, our framework extends the recently proposed propagation networks (PropNets) and consists of two complementary components, a *physics predictor* and a *belief regulator*. While the former predicts the future states of the object(s) manipulated by the robot, the latter constantly corrects the robots knowledge regarding the objects and their relations. Our results showed that after trained in a simulator, the robot could reliably predict the consequences of its actions in object trajectory level and exploit its own interaction experience to correct its belief about the state of the world, enabling better predictions in partially observable environments. Furthermore, the trained model was transferred to the real world and its capabilities were verified in correctly predicting trajectories of pushed interacting objects whose joint relations were initially unknown. We compared our BRDPN against the original PropNets and showed that BRDPN can perform consistently well even if the relations between the objects are not explicitly given but instead predicted from observations.**

I. INTRODUCTION

Predicting effects in complex robotic systems is a challenging problem, especially in the presence of varying numbers of objects and the rich and wide variety of interactions among these objects. When objects are linked with physical connections, this would also suggest some semantic connections between them, such as the motion of one object can propagate its motion onto another object, which might lead to a chain effect. To be able to model such systems accurately, data has to be represented in a way that it appropriately handles the encoding of multiple objects and their interactions with each other, and it should be robust to noise.

Recently, a great amount of effort has put on the prediction of the dynamics via graph networks (e.g. [1], [2]). These works can deal with varying number of objects and learn rich interaction dynamics among these objects. Some of these works have focused on unsupervised learning, while others were aimed at developing learnable physics engines. However, applying them to model robot-object interactions is not very straightforward as the active involvement of the robot was not taken into account and, moreover, uncertainty in perception was not explicitly addressed.

In this work, we propose *Belief Regulated Dual Propagation Network (BRDPN)* which takes the actions of the robot into account in predicting the next states[1]. It further continuously regulates its belief about the environment based on its interaction history to correct its future predictions. For belief regulation, extending the recently proposed propagation networks (PropNets) [3] that handle instantaneous effect propagation, we propose a temporal propagation network that takes history of the motion of each object to predict unknown object or relation properties. Our system is verified on a table-top push setup that has cylindrical objects and joint relations between them. Our setup includes varying number of objects that might be connected to each other with *rigid*, *revolute* or *prismatic joints*. The model definitions of these types of relations, including the PropNet$_n$ relation, is not provided to the robot. Furthermore, the relations between objects cannot be perceived by the robot. From its interaction experience in the simulator, it learns to predict relations between objects given observed object motions, and exploits this information to predict future object trajectories. Furthermore, it was transferred to and verified in real world experiments that included around 100 interactions. Our system was shown to outperform the original PropNets, both in simulation and real-world, when the relations between objects were not reliably provided to the system.

Our contribution to the state of the art is two-fold. First, we introduced a deep neural network based method for learning how to exploit the interaction experience of the robot to extract values of otherwise unknown state variables in partially observable environments. Second, we implemented a learning based effect prediction robotic framework that can handle multiple interacting objects that might have different types of connections, and we verified this framework both in simulated and real robot experiments.

II. RELATED WORK

The past years have seen considerable progress in modelling physics with probabilistic approaches. For instance, Battaglia et al. [4] proposed a Bayesian model called Intuitive Physics Engine and showed that physics of stacked cuboids

*This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 731761, IMAGINE; and supported by a TUBA GEBIP fellowship awarded to E. Erdem.

[1]Ahmet E. Tekden, Mert Imre, M. Yunus Seker and Emre Ugur are with Computer Engineering Department, Bogazici University, Turkey

[2]Aykut Erdem and Erkut Erdem are with Computer Engineering Department, Hacettepe University, Turkey

[1]Our source code and experimental data are available in the web page prepared for this paper: `https://fzaero.github.io/DPNet/`.

can be modeled with this model. Deisenroth et al. [5] suggested a probabilistic dynamic model that depends on Gaussian Processes and that is capable of predicting the next state of a robot given the current state and its actions. Recently, some researchers extended these works by using deep learning methods to model physics. Wu et al. [6] proposed a deep approach for finding the parameters of a simulation engine which predicts the future positions of the objects that slide on various tilted surfaces. Lerer et al. [7] trained a deep network to predict the stability of the block towers given their raw images obtained from a simulator. A specific topic of interest within modeling physics with deep learning is motion prediction from images, which has gained increasing attention over the last few years. The studies presented for this task either employ convolutional neural network (CNNs) or graph neural networks (GNNs).

Mottaghi et al. [8] trained a CNN for motion prediction on static images by casting this problem as a classification problem. Mottaghi et al. [9] employed CNNs to predict movements of objects in a static image when some external forces are applied to them. Fragkiadaki et al. [10] suggested a deep architecture in which the outputs of a CNN are used as inputs to Long Short Term Memory (LSTM) cells [11] to predict movements of balls in simulated environments.

A number of studies have examined the action-effect prediction in videos. Finn et al. [12] proposed a convolutional recurrent neural network [13] to predict the future image frames using only the current image frame and actions of the robot. Byravan et al. [14] presented an encoder-decoder like architecture to predict SE(3) motions of rigid bodies in depth data. However, the output images get blurry over time or their predictions tend to drift away from the actual data due to the accumulated errors, making it not straightforward to use for long-term predictions in robotics.

As deep structured models, GNNs allows learning useful representations of entities and relations among them, providing a reasoning tool for solving structured learning problems. Hence, it has found particularly wide use in physics prediction. Interaction network by Battaglia et al. [1] and Neural Physics Engine by Chang et al. [2] are the earliest examples to general purpose physic engines that depend on GNNs. These models do object-centric and relation-centric reasoning to predict movements of objects in a scene. Though they were successful in modeling dynamics of several systems such as n-body simulation and billiard balls, their models had certain shortcomings, especially when an object's movement has chain effects on other objects (*e.g.* a pushed object pushes other object(s) it is contacting with) or when the objects in motion have complex shapes. These shortcomings can be partly handled by including a message passing structure within GNNs as done in the recent works such as [3], [15], [16]. Watters et al. [17] and van Steenkiste et al. [18] proposed hybrid network models which encode object information directly from images via CNNs and which predict the next states of the objects with the use of GNNs. Our framework differs from these GNN approaches in that it learns to predict the relations between objects from the interaction history, consequences of the predicted relations in estimating their future states. Additionally, different from these approaches, our model is verified in a real robotic setup.

While the studies above focused on predicting the next states of the objects given relations among them, researchers [19], [20], [21], [22] also studied on estimating the joint relations between objects for real-time tracking and prediction of the articulated motions in challenging perceptual settings. These works however assume expert knowledge about the joint types and hard-code the corresponding transformation matrices [20], candidate template models [19], specific measurement models [21], [22] to detect kinematic structures. Our system assumes no prior knowledge about joint dynamics and the robot learns the dynamics of categories purely from observations. Therefore, learning dynamics of completely novel relation types is possible with our system. Exceptionally [19] learns articulation dynamics from data; however it was only realized on a single-pair of objects from a single articulation observation (garage door motion). Furthermore, these studies do not learn or predict how the pairs or chains of non-articulated touching objects would propagate the applied forces along the cluster/chain, whereas our system can predict the propagated effect on groups of touching non-articulated objects.

III. Proposed Model

In this section, we introduce the *Belief Regulated Dual Propagation Networks (BRDPN)* and explain how it extends the propagation network framework for articulated multi-part multi-object settings to allow the regulation of the beliefs about environment state variables. Belief regulation corresponds to regulating robot's belief about environment through extracting or updating the values of state variables. Fig. 1 shows a graphical illustration of our framework, which is composed of two main components: a *physics predictor* and a *belief regulator*. The physics predictor is based on propagation network and responsible for predicting future states of the manipulated objects. The belief regulation module is a propagation network with recurrent connections, which we call temporal propagation network. Belief regulation module is responsible from extracting/updating the knowledge of the robot about the environment through its observations of own-interaction experience. In the following, we give technical details of these models.

Preliminaries: Assume that the robot is operating in a complex environment involving a set of multi-part objects O, we express the scene with a graph structure $G = \langle O, R \rangle$ where the nodes $O = \{o_i\}_{i=1:N^o}$ represent the set of objects (of cardinality N^o) and the edges $R = \{r_k\}_{k=1:N^r}$ represent the set of relations between them (of cardinality N^r). More formally, each node $o_i = \langle x_i, a_i^o \rangle$ stores object related information, where $x_i = \langle q_i, \dot{q}_i \rangle$ is the state of object i, consisting of its position q_i and velocity $\dot{q}_i$, and a_i^o denotes physical attributes such as its radius or mass. Each edge $r_k = \langle d_k, s_k, a_k^r \rangle$ encodes the relation between objects i and j with $d_k = q_i - q_j$ representing the displacement vector,

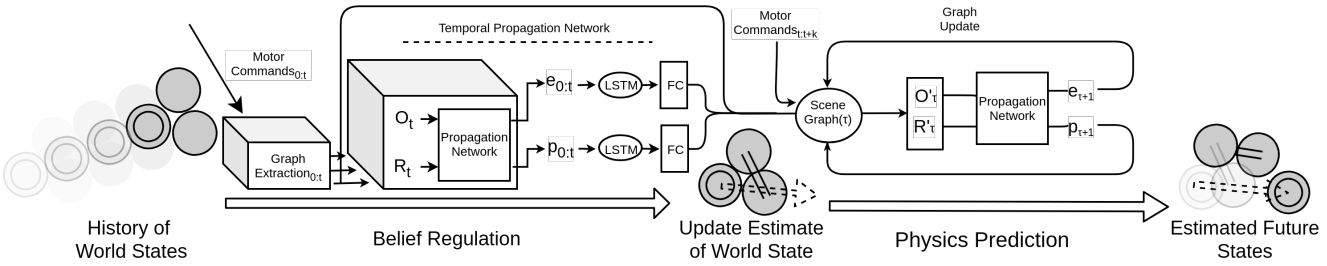


Fig. 1: *Belief Regulated Dual Propagation Networks*. System contains two parts: belief regulation module, and physics prediction module. Given previous world state values and motor commands, the belief regulation module is used to update the estimate of the state variables. Given the current estimate of the world state and planned motor commands, the physics prediction module predicts the sequence of future states expected to be observed.

$s_k = \dot{q}_i - \dot{q}_j$ denoting the velocity difference between them, and a^r_k representing attributes of relation k such as the type of the joints connecting objects i and j.

Physics Prediction: Propagation networks encode the states of the objects and the relations between them separately. This encoding is carried out by two encoders, one for the relations denoted by f^{enc}_R and one for the objects denoted by f^{enc}_O, defined as follows:

$$c^r_{k,t} = f^{enc}_R\left(r_{k,t}\right), \quad k = 1 \dots N^r \tag{1}$$

$$c^o_{i,t} = f^{enc}_O\left(o_{i,t}\right), \quad i = 1 \dots N^o \tag{2}$$

where $o_{i,t}$ and $r_{k,t}$ represent the object i and the relation k at time t, respectively.

To predict the next state of the system, these encoders are used in the subsequent propagation steps within two different propagator functions, f^l_R for relations and f^l_O for object, at the propagation step l, as follows:

$$e^l_{k,t} = f^l_R\left(c^r_{k,t}, p^{l-1}_{i,t}, p^{l-1}_{j,t}\right), \quad k = 1 \dots N^r \tag{3}$$

$$p^l_{i,t} = f^l_O\left(c^o_{i,t}, p^{l-1}_{i,t}, \sum_{k \in \mathcal{N}_i} e^{l-1}_{k,t}\right), \quad i = 1 \dots N^o \tag{4}$$

where $\mathcal{N}_i$ denotes the set of relations where object i is being a part of, and $e^l_{k,t}$ and $p^l_{i,t}$ represent the propagating effects from relation k and object i at propagation step l at time t, respectively. Here, the number of propagation steps can be decided depending on complexity of task. Through using the predicted states as inputs, it can chain the predictions and estimate the state of the objects at $t + T$. See [3] for more detailed description of this network.

Belief Regulation: The success of physics prediction step highly depends on how accurate the environment is encoded in the graph structure. Here we refer to the term belief as the estimated world state and given previous states and motor commands, the role of the belief regulation module is to constant updates on this crucial part. As the main theoretical contribution in this paper, we propose a *temporal propagation network* architecture that augments a propagation network with a recurrent neural network (RNN) unit to regulate beliefs regarding object and relation information over time. More formally, it takes a sequence of a set of state variables during the action execution as input and by means of a secondary, special-purpose propagation network, it encodes these structured observations, which are then fed into an RNN cell to update the current world state, as follows:

$$r'_{k,t} = f^{blf}_O\left(e^L_{k,t}, r'_{k,t-1}\right), \quad k = 1 \dots N^r \tag{5}$$

$$o'_{i,t} = f^{blf}_R\left(p^L_{i,t}, o'_{i,t-1}\right), \quad i = 1 \dots N^o) \tag{6}$$

where L denotes the propagation step, and f^{blf}_O and f^{blf}_R denote the RNN-based encoder functions for objects and relations, respectively. Feeding these functions with the sequence of encoding vectors $r'_{k,t-1}$ and $o'_{i,t-1}$ allows the temporal propagation network to consider the overall history of object and relation states from the previous time-steps. Hence, it continuously updates its belief regarding objects and relations states ($o_{i,t}$ and $r_{k,t}$), and eventually minimize the difference between the effect predicted by our physics prediction module and reality.

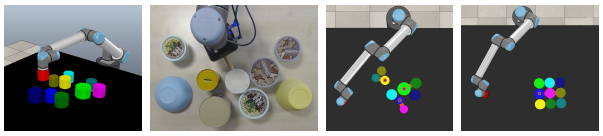
IV. Experimental Setup

We evaluate our model in simulation and on a real robot through a set of experiments. In the following, we give the details of the experimental setups designed to assess the generalization performance to the changing number of objects and time steps, transferability of our model to different object-relation distributions and to the real world setting.

A. Robotic Setup

Our simulation and real world experiments included a 6 degrees of freedom UR10 arm and a number of cylindrical objects placed on a table as shown in Fig 2a-b. The table-top settings were composed of objects of varying numbers and sizes. The objects might move independent of each other (no-joint) or connected to each other through three different joint relation types, namely *fixed*, *revolute* and *prismatic* joints. The robot learned effect prediction by self-exploration and observation in the V-REP physics based simulator with Bullet engine[2]. For this, the simulated robot exercised its push action on a set of objects by moving a cylindrical object that was attached to its end-effector. After training, the learned prediction capabilities were tested both in the simulated and the real world settings.

[2]www.coppeliarobotics.com/



(a) Simulation (b) Real World (c) Sparse (d) Dense

Fig. 2: Robotic setup. Table top scenes used in (a) simulation and (b) real world experiments. (c)-(d) Initial configurations used for training and testing.

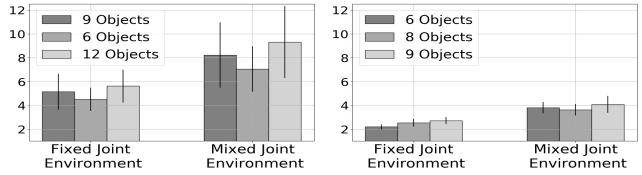
In our simulation experiments, we considered two different configurations for scene generation: a *sparse* configuration (Fig.2c) where objects were initially scattered randomly in the scene, and a *dense* one (Fig.2d) where the objects were initially grouped together. Sparse configuration was specifically designed to maximize the contact time between the end-effector and the objects, and to allow rich set of interactions. The robot chooses 8 different linear motions of 30cm, maximizing contact time with most diverse set of objects. While in the sparse configuration the objects were randomly scattered in the scene, in the dense configuration the objects were grouped together in a grid structure. Sparse configuration was used for training, and dense configuration for testing the generalization performance of the model on novel environments, i.e. on instances drawn from a completely different distribution of objects and relations.

A total of 900 different 9-objects scenes were used for training the model. Two different settings were used for testing. The sparse test set was composed of 50 9-objects, 25 6-objects and 25 12-objects scenes. The dense test set was composed of 50 6-objects, 50 8-objects and 50 9-objects scenes. For each scene above, the robot arm approached from four different random directions. Each object in these scenes had radii between 8 cm to 16 cm. In the evaluations, separate models were trained and tested on scenes where only fixed joints and mixed type of joints exist.

B. Implementation Details

Our physic prediction module takes object position, velocity and object radius as object features, and joint relation type between objects as relation features. Specifically, object encoder takes object radius and velocity, and is implemented with a MLP with 3 hidden layers of 150 neurons. The relation encoder takes position and velocity differences between objects at the receiver and sender end of the edges, along with their radius and joint relation type, and is implemented with a MLP with 1 hidden layer of 100 neurons. While our relation propagator is a MLP with 2 hidden layers of 150 neurons, our object propagator is a MLP with one hidden layer of 100 neurons. During training, at each epoch, we validated our physics prediction module on the validation set containing instances from the sparse configuration, and selected the model that has the lowest mean squared error (MSE) over 200 time-step trajectory roll-outs.

Our belief regulation module uses the sequence of positions, velocities and radii of the objects, and predicts joint types between each pairs of objects. Output of the relation propagator is connected to LSTM with 100 hidden neurons. This LSTM is then connected to fully connected layer to predict joint type between objects. This network was trained using sequences with 100 time steps. During training, we optimized this network with the loss coming from the predicted joint types between the time steps 50 and 100 to make sure that our model can generalize to the changing number of time steps, while not over-fitting to the position information coming from the single time steps.



(a) Sparse configuration (b) Dense configuration

Fig. 3: Error(cm) for object positions over (a) 200 time-step trajectory roll-outs for sparse configuration, and (b) 50 time-step trajectory roll-outs for dense configuration.

V. Results

For quantitative analysis, we compared our method with PropNets with alternative (hard-coded) relation assignments: As a strong baseline, **PropNet**$_{gt}$ uses ground-truth relations. **PropNet**$_{f}$ assumes all pairs of contacting objects have fixed relations between them. **PropNet**$_{n}$ assumes no joints between objects. Furthermore, to analyze the influence of temporal data in predicting relations within our model, we also report results with **1-step BRDPN** that predicts object relations using only the observation from the previous step.

A. Quantitative Analysis of Separate Modules in Simulation

First, the physics prediction module is evaluated given ground-truth relation information. Fig. 3 presents the performance on the test set for different object configurations. Each bar provides the mean error averaged over differences between predicted and observed trajectories. We carry out the evaluation for both the sparse and dense configuration settings in *fixed-joint* and *mixed-joint* environments separately. As shown, around 7 and 3 cm mean error is observed in sparse and dense object configurations. Furthermore, we observed that the error drops significantly (to 4 and 2 cm) in case only fixed joints are included. Given the average motion (including zero motion in many cases) in objects is $40cm$ sparse and $18cm$ in dense configuration, these results show that the model achieves high prediction performance if it uses the ground-truth relations; and with increasing complexity of object relations, learning becomes more challenging.

Next, the performance of our belief regulation prediction module is evaluated on the sparse test set. As shown in Fig. 4, the accuracy is already very high from the instant when the robot makes its first contact in *fixed-joint* environments. The accuracy increases in *mixed-joint* environments to over 98% as well, with the accumulated observations from the interactions of the robot.

We performed a number of experiments in the dense configuration as well. However, we observed that directly

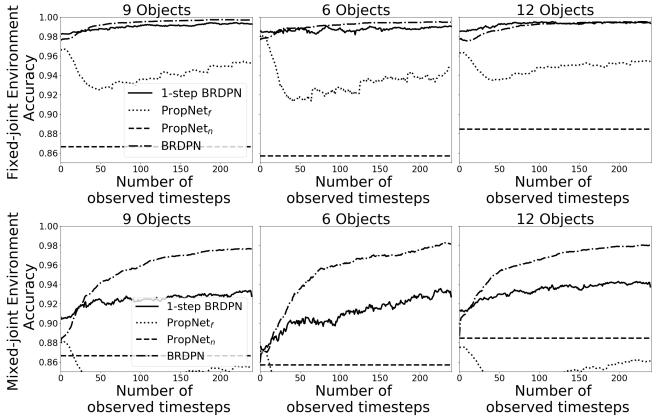


Fig. 4: Relation prediction accuracies (sparse configuration).

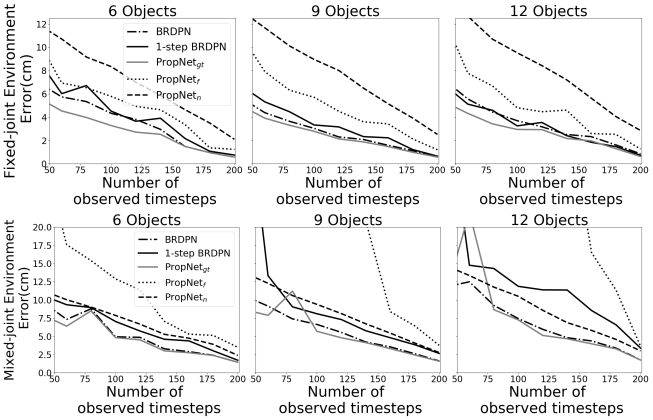
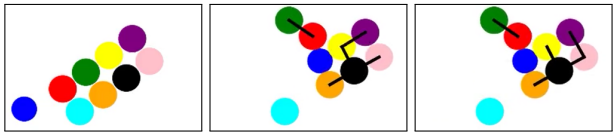


Fig. 6: Error of the BRDPN in sparse configuration.



(a) Initial Scene (b) Predicted Relations (c) Ground Truth Relations

Fig. 5: (a) The end effector of the robot (shown in blue color) moves towards the object group in the simulator. The predicted joints are provided in (b) and the real ones (not visible to the robot) are shown in (b) with the black lines. Even though the joints were not correctly identified, the inference on joint relations was a plausible one given interaction experience of the robot.

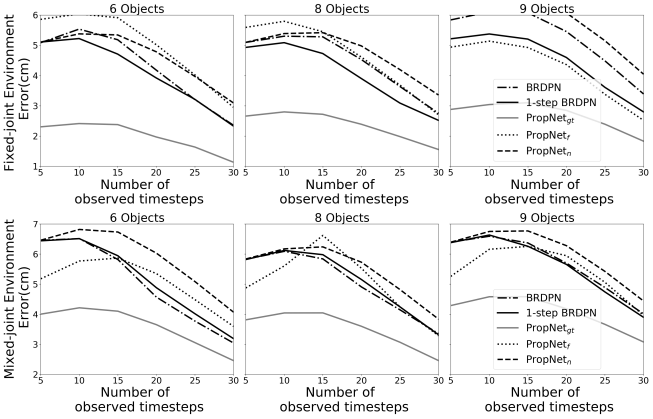


Fig. 7: Error of the BRDPN in dense configuration.

comparing real and predicted relations in this configuration might be misleading as different sets of joints that connect the objects in the same grid might generate identical effects in response to the robot interactions. The system might suffer from ambiguities in predicting joint relations from such interaction experience. For example, a group of objects that form a rigid body through different set of connections would behave same in response to the push action. Fig. 5 provides a snapshot of such a case where the robot started interaction with 8 objects placed on a grid. In this case, even if the joint relations were incorrectly predicted for the sub-group of 5 objects, this was actually a plausible inference that enabled the system to make correct prediction about the object trajectories from that moment. While incorrect state predictions might not affect the effect prediction performance of the system in this particular extreme example, we might need intelligent exploration strategies that enable the robot to collect more reliable information in other ambiguous cases.

B. Quantitative Analysis of the BRDPN in Simulation

In this section, the complete system is evaluated at different time-points during interactions. The belief regulation module predicts the relations between objects using the observations upto the corresponding time-points. Given the states of the objects, the robot actions, and the predicted relations between pairs of objects, the physics prediction module finds the trajectories of the objects that are expected to be observed for the rest of the motion. The results are presented in Fig. 6 and Fig. 7 where the errors on the remaining trajectories are computed with the predictions of the system at the reported time steps. These results indicate that even if the relations are unknown, the proposed belief regulation improves the effect prediction performance of the system with more interaction experience. While BRDPN performs better for the sparse dataset, 1-step BRDPN performs similar to or better than BRDPN in dense configurations probably because the model was optimized for temporal information coming from sparse environments. Note that BRDPN outperforms the PropNets variants that do not utilize the ground-truth relations.

C. Real World Experiments

In this section, we provide the results obtained in the real world. For this, the prediction model trained in the simulator was directly transferred to the real world. A mallet that was grasped by the 3-finger gripper of the UR10 robot was used to push objects The cylindrical objects on the setup can be seen in the the Fig. 2. Only one type of joint, namely fixed joint was used in this setup. Fixed joint relations are accomplished by placing customized card-boards under the specified objects, making all the group move together. A top-down oriented RGB camera with 1920×1080 pixels resolution was placed above the scene, ARTags were placed

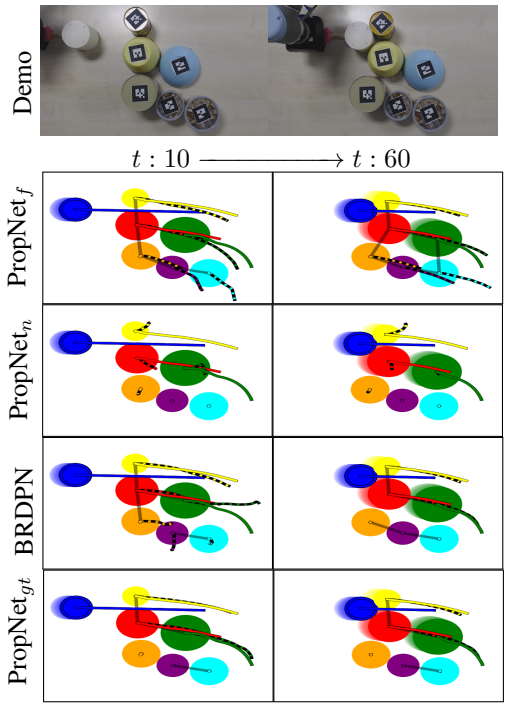


Fig. 8: The first real-world interaction example. The relation assignments/predictions, the real and the predicted trajectories are shown with black, solid colored and dashed colored lines, respectively.

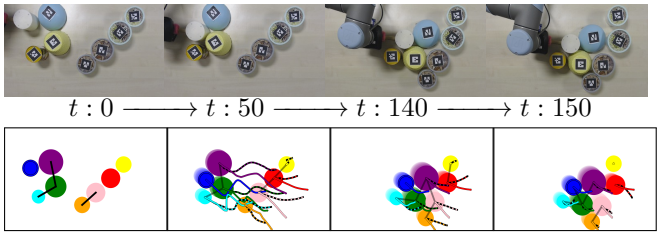


Fig. 9: The second real-world interaction example.

on the objects for tracking.

First we present the results qualitatively over two example scenes. In the first example scene, 6 objects were placed as a group as shown in Fig. 8, where top left 3 objects and the bottom right pair were connected to each other. A straight push motion was executed by the robot and the object positions at time steps 10 and 60 were provided. Solid and dashed lines show the real and predicted trajectories. As shown, given ground-truth joint information, the model made almost perfect trajectory predictions. When the ground-truth relations are not provided, as in PropNet$_n$ and PropNet$_f$, the model either predict that all objects are pushed aside or all contacting ones move together. Finally, when the relations are predicted, first the model predicts trajectories similar to PropNet$_f$ case, but after seeing the independent motions of upper three object group, it corrects the joint relations, and predicts the correct trajectory successfully. In the second example scene, a more challenging configuration was used, where 7 objects were placed in two separate groups and objects in each group are attached to each other (Fig. 9). The

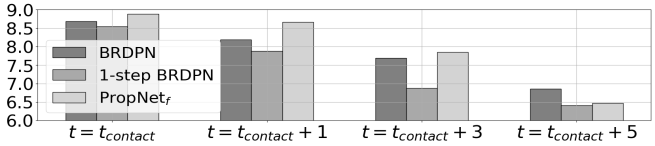


Fig. 10: Average error (in cm) in the real world.

end-effector made a zigzag motion towards the objects. The relation prediction on the first group (the one closer to the robot) was correct at $t:50$, since the robot had sufficient interaction with these objects. The indirect contact to the second group via the first group took place slightly before $t:140$ and the robot correctly inferred that not all the objects in the second group had fixed relations. The prediction of the first group remained correct, but the robot made incorrect predictions in two cases: it incorrectly inferred that the first and second group was connected, and that the the top-right pair was also connected. With further interaction, these incorrect inferences were corrected at $t:150$.

Finally, we evaluate our model quantitatively with large number of interactions. We generated 102 different setups that include 2 to 5 objects with 1 to 3 connections. One of the 5 different predefined straight motions of 30 to 60 cm was applied towards these objects that were placed in different locations which results in objects moving 19.5 cm on average. Our model achieved an average error of 6.6 cm in predicting their final positions. Although in some cases incorrect effect predictions caused failures in predicting the movement direction of interacted objects, our model performed well considering the average diameter of 12 cm of the objects and our direct transfer strategy from the simulation. Fig. 10 provides a more detailed analysis of the results focusing to the time-point when the first contact with the objects occur. As shown, the prediction error of 1-step BRDPN quickly drops compared to the model that assigns fixed-joint to objects whose distances are smaller than 2.5 cm. Probably after the objects physically separated from each other, PropNet$_f$ does not consider those objects to be attached to each other and also start making predictions with similar accuracy. Note that PropNet$_n$ significantly underperformed and was not included in the figure, and ground-truth-relation model generated higher performance consistently, obtaining around 4 cm error at the end.

VI. Conclusion

We presented *Belief Regulated Dual Propagation Networks (BRDPN)*, a general purpose learnable physics engine that also continuously updates the estimated world state through observing the consequences of its own interactions. We demonstrated our network in setups containing articulated multi-part multi-objects settings. In these settings, we validated our network and its modules on several test cases. While our system was validated in both simulation and real world robotic experiments, we discussed that intelligent exploration strategies that resolve inference problem in ambiguous situations are necessary. In the future, we aim to study on generating goal directed action trajectories that balance the trade-off between exploration and exploitation.

REFERENCES

[1] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, *et al.*, "Interaction networks for learning about objects, relations and physics," in *Advances in neural information processing systems*, 2016, pp. 4502–4510.

[2] M. B. Chang, T. Ullman, A. Torralba, and J. B. Tenenbaum, "A compositional object-based approach to learning physical dynamics," in *International Conference on Learning Representations*, 2017.

[3] Y. Li, J. Wu, J.-Y. Zhu, J. B. Tenenbaum, A. Torralba, and R. Tedrake, "Propagation networks for model-based control under partial observation," in *International Conference on Robotics and Automation*, 2019.

[4] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum, "Simulation as an engine of physical scene understanding," *Proceedings of the National Academy of Sciences*, vol. 110, no. 45, pp. 18 327–18 332, 2013.

[5] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning (International Conference on Machine Learning)*, 2011, pp. 465–472.

[6] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum, "Galileo: Perceiving physical object properties by integrating a physics engine with deep learning," in *Advances in neural information processing systems*, 2015, pp. 127–135.

[7] A. Lerer, S. Gross, and R. Fergus, "Learning physical intuition of block towers by example," in *International Conference on Machine Learning*, 2016.

[8] R. Mottaghi, H. Bagherinezhad, M. Rastegari, and A. Farhadi, "Newtonian scene understanding: Unfolding the dynamics of objects in static images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3521–3529.

[9] R. Mottaghi, M. Rastegari, A. Gupta, and A. Farhadi, ""what happens if..." learning to predict the effect of forces in images," in *European Conference on Computer Vision*. Springer, 2016, pp. 269–285.

[10] K. Fragkiadaki, P. Agrawal, S. Levine, and J. Malik, "Learning visual predictive models of physics for playing billiards," in *International Conference on Learning Representations*, 2016.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *Advances in neural information processing systems*, 2016, pp. 64–72.

[13] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.

[14] A. Byravan and D. Fox, "SE3-Nets: Learning rigid body motion using deep neural networks," in *International Conference on Robotics and Automation*, 2017, pp. 173–180.

[15] D. Mrowca, C. Zhuang, E. Wang, N. Haber, L. F. Fei-Fei, J. Tenenbaum, and D. L. Yamins, "Flexible neural representation for physics prediction," in *Advances in neural information processing systems*, 2018, pp. 8813–8824.

[16] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," in *International Conference on Learning Representations*, 2019.

[17] N. Watters, D. Zoran, T. Weber, P. Battaglia, R. Pascanu, and A. Tacchetti, "Visual interaction networks: Learning a physics simulator from video," in *Advances in neural information processing systems*, 2017, pp. 4539–4547.

[18] S. van Steenkiste, M. Chang, K. Greff, and J. Schmidhuber, "Relational neural expectation maximization: Unsupervised discovery of objects and their interactions," in *International Conference on Learning Representations*, 2018.

[19] J. Sturm, V. Pradeep, C. Stachniss, C. Plagemann, K. Konolige, and W. Burgard, "Learning kinematic models for articulated objects," in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[20] T. Schmidt, R. A. Newcombe, and D. Fox, "Dart: Dense articulated real-time tracking." in *Robotics: Science and Systems*, vol. 2, no. 1. Berkeley, CA, 2014.

[21] R. Martín-Martín, S. Höfer, and O. Brock, "An integrated approach to visual perception of articulated objects," in *International Conference on Robotics and Automation*. IEEE, 2016, pp. 5091–5097.

[22] R. Martín-Martín and O. Brock, "Coupled recursive estimation for online interactive perception of articulated objects," *The International Journal of Robotics Research*, 2019.