Belief Regulated Dual Propagation Nets for Learning Action Effects on Articulated Multi-Part Objects

A Preprint

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

September 9, 2019

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 articulated multi-part multi-objects. Specifically, our framework extends the recently proposed propagation networks 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.

1 INTRODUCTION

In robotics, planning, and control of a robotic system play a crucial role in the execution. To enable planning and improve its capacity, prediction mechanisms are utilized for anticipating the impact of an action or the motion in advance. In our previous work, we showed that learned effect predictions in sub-symbolic [1] and symbolic [2] spaces enabled the robots to make multi-step plans to achieve given goals. However, those predictions provided only approximate estimations about the next states as the effects were represented with categorical variables. Lower-level effect prediction that takes into account real-valued state variables and temporal information, on the other hand, would provide the possibility to mentally simulate any action on any object.

Predicting effects in complex physical 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 video prediction frameworks and graph networks [3, 4, 5, 6, 7]. Some of these works have focused on unsupervised learning, while others were aimed at developing learnable physics engines. On the subject of effect prediction, recent works commonly employ graph

[*] *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

neural networks. Mostly, the domains explored in these studies are not conditioned on actions, because they do not have any external agent that acts inside of the system. The training data were collected passively and most of the training iterations establish coherent distributions with the previous trials. However, for robotic systems that have an external agent whose actions change the system, the learning model should be able to predict action-conditioned effects. In this study, we focus our attention on this mostly neglected issue and investigate the use of graph neural networks for modeling action-conditioned effect prediction. Furthermore, in the robotics domain, we need to deal with uncertainty in sensorimotor data. With this work, we contribute to the state of the art through including the robot actions in the learning and prediction loop and introducing a temporal propagation network that learns to extract unobservable state variables from the interaction experience of the robot. Finally, to the best of our knowledge, this is the first time these deep prediction capabilities are verified in real robot settings where the robot can predict action effects on multiple articulated objects in low-level trajectory level.

To understand and learn the dynamics of systems containing explicitly selected actions (*e.g.* a robot hand manipulating objects in an unstructured environment), it is essential to have a policy to generate sufficiently diverse and large training data with a variety of state/action outcomes. Sparsity in the data may cause the model to output inaccurate predictions so that the errors can accumulate to a point that the model ends up with a physically impossible state of the environment over the course of multi-step predictions. These errors can get more catastrophic when the configuration of the scene is quite different from those seen during training.

In addition to the aforementioned issues, there could be errors caused by the missing or incorrect knowledge about the world (*i.e.* unknown object properties or relation attributes, or the discrepancy between the actual motion of the objects and the belief of the agent on how the objects should actually move). Ideally, the robot should be able to correct these errors in time as it observes the effects of its own actions. For example, when the robot is about to contact an object in order to push it, if the motion of the object is observed as it is pushed before the contact, then the robot should be able to correct the perception error in tracking and/or geometry information about the object as the action continues. As another example, while pushing an object, if the robot observes the movement of another object which it has no contact with, the robot should be able to infer that these two objects might be physically related. So in general, the robot should improve its effect prediction performance by correcting its beliefs about the dynamic system using the past observations of the objects' motion information up to the present time.

In our work, we propose *Belief Regulated Dual Propagation Network (BRDPN)* which can regulate its belief about the environment by making use of observed object motions to correct its future predictions. For belief regulation, extending the recently proposed propagation networks [8] 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. We showed that it can effectively predict joint types between objects given observed object motions, and use this information to predict future object trajectories of position and velocity.

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.

2 Related Work

The past years have seen considerable progress in modelling physics with probabilistic approaches. For instance, Battaglia et al. [9] proposed a Bayesian model called Intuitive Physics Engine and showed that physics of stacked cuboids can be modeled with this model. Deisenroth et al. [10] 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. [11] 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.[12] 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. [13] trained a CNN for motion prediction on static images by casting this problem as a classification problem. Mottaghi et al. [14] employed CNNs to predict the movements of objects in a static image when some external forces are applied to them. Fragkiadaki et al. [15] suggested a deep model architecture in which the outputs

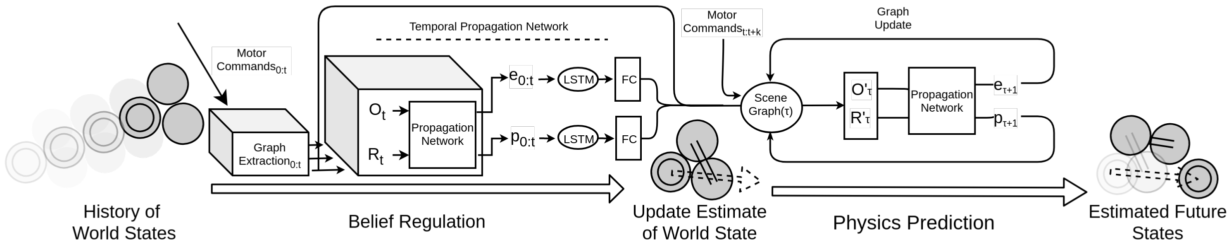


Figure 1: *Belief Regulated Dual Propagation Networks*. System contains two parts: belief regulation module, and physics prediction module. Given previous world state values and motor commands, the belief regulation module is used to update the estimate of the state variables. Given the current estimate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

of a CNN are used as inputs to Long Short Term Memory (LSTM) cells [16] to predict movements of balls in simulated environments.

Finn et al. [5] proposed a convolutional recurrent neural networks [17] based model to predict the future image frames using only the current image frame and actions of the robot. Byravan et al. [6] presented an encoder-decoder like architecture to predict SE(3) motions of rigid bodies in depth data.

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. [3] and Neural Physics Engine by Chang et al. [4] 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 [8, 18, 19]. Watters et al. [20] and van Steenkiste et al. [21] 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.

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

Physics Prediction In this paragraph we summarize the propagation network that is used in physics prediction step. See [8] for more detailed description of this network.

Propagation networks encode the objects and the relations between them with the functions f_O and f_R, respectively, and employs them to predict the state of the objects at time t. Through using the predicted states as inputs, it can chain the predictions and estimate the state of the objects at $t + T$. It can cope with instantaneous propagation of effects by using mechanisms that allow passing messages in graphical models. The shared information regarding objects and relations are first encoded before using them in the propagation steps. This encoding is carried out by two encoders, one for the relations denoted by f_R^{enc} and one for the objects denoted by f_O^{enc}, defined as follows:

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

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

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_R^l for relations and f_O^l for object, at the propagation step l, as follows:

$$e_{k,t}^l = f_R^l\left(c_{k,t}^r, p_{i,t}^{l-1}, p_{j,t}^{l-1}\right), \quad k = 1 \dots N^r \tag{3}$$

$$p_{i,t}^l = f_O^l\left(c_{i,t}^o, p_{i,t}^{l-1}, \sum_{k \in \mathcal{N}_i} e_{k,t}^{l-1}\right), \quad i = 1 \dots N^o \tag{4}$$

where $\mathcal{N}_i$ denotes the set of relations where object i is being a part of, and $e_{k,t}^l$ and $p_{i,t}^l$ 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.

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:

$$r_{k,t}' = f_O^{blf}\left(e_{k,t}^L, r_{k,t-1}'\right), \quad k = 1 \dots N^r \tag{5}$$

$$o_{i,t}' = f_R^{blf}\left(p_{i,t}^L, o_{i,t-1}'\right), \quad i = 1 \dots N^o) \tag{6}$$

where L denotes the propagation step, and f_O^{blf} and f_R^{blf} 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.

4 Experimental Results

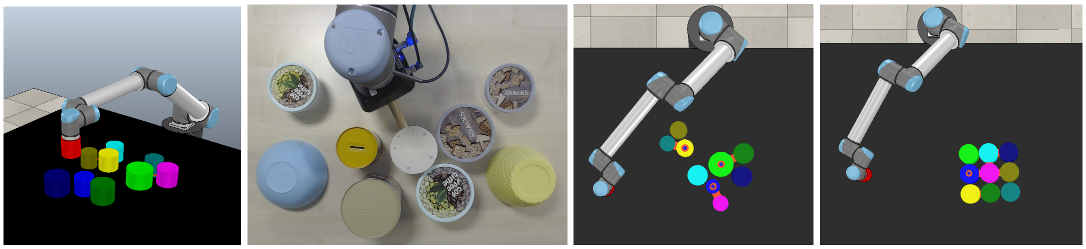
4.1 Experimental Setup

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 2. The robot learned effect prediction by self-exploration and observation in the V-REP[4] physics based simulator with Bullet engine. 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.

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. Fixed joints stuck two objects together as one complex rigid object. Revolute joints allowed rotational motion for one of the objects around the vertical axis of the other one but fixed the distance between the objects. Lastly, the prismatic joints fixed two objects to each other with respect to the rotation but allowed the distance change between the two objects. These joints types and their expected outcomes are illustrated in Table 1 in supplementary material.

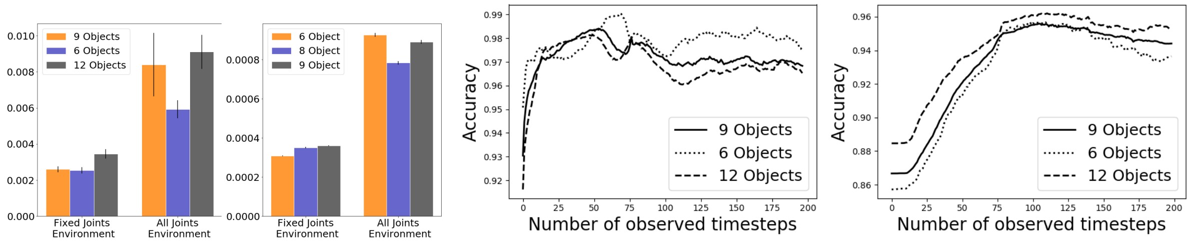
In our experiments, we considered two different configurations for scene generation: a sparse configuration (Fig.2c) where objects were initially scattered randomly in the scene and a dense one (Fig.2d) where the objects were initially

[4] www.coppeliarobotics.com/



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

Figure 2: On left, table top scenes used in simulation and in the real world experiments.On right, initial configurations used for training and testing



(a) Sparse configuration (b) Dense configuration (c) Accuracy on fixed joints environment (d) Accuracy on all joints environment

Figure 3: Mean Square errors 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. Accuracies acquired in sparse configuration by belief regulator (c) fixed joints environment and (d) all joints environment.

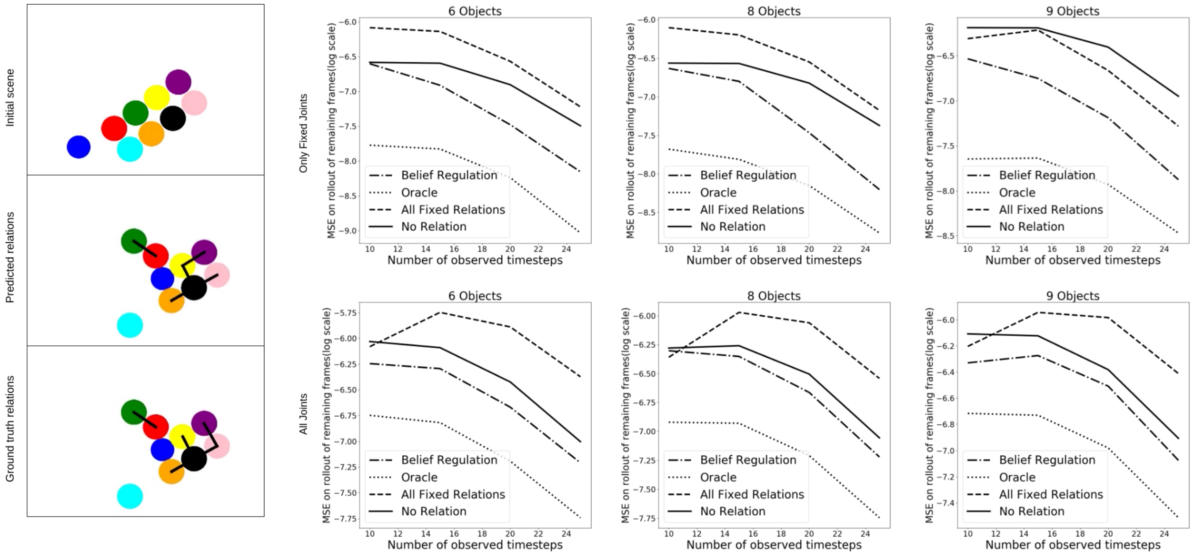
grouped together. Sparse configuration was specifically designed to maximize contact time between the end-effector and the objects, and to allow rich set of interactions. While in the sparse configuration the objects were randomly scattered in the scene, dense configuration mostly included the objects that were grouped together in a grid structure. Hence, it was specifically designed in order to test the generalization performance of the proposed model in novel environments by letting the instances drawn from a completely different distribution of objects and relations. More information on setups can be found on supplementary material.

4.2 Evaluation Protocol

In training the proposed belief regulated dual propagation networks system and its subcomponents physics prediction and belief regulation modules, we used a training set that includes 900 scenes that consist of only 9 objects, and that is generated under the sparse configuration. Our validation set includes 200 scenes of which 150 of them consist of 9 objects, 25 of them 6 objects, and 25 of them 12 objects, all initially placed under the sparse configuration. For testing, we consider two settings from the sparse and the dense object configurations of objects. Under the sparse configuration, we use 100 scenes of which 50 of them consist of 9 objects, 25 of them 6 objects, and 25 of them 12 objects. Under the dense configuration where we test the generalization performance of our model, we use 150 scenes of which 50 of them have initially a group of 2×3 objects, 50 of them a group of 2×4 objects, and 50 of them a group of 3×3 objects. In evaluations, we used separate networks trained on scenes containing only fixed joint and all joint relations.

4.3 Effect prediction with full state information

Fig. 3 presents the performance of our physics prediction module on our 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 with some objects connected with only fixed and with any type of joints. As shown, around 1 cm mean error is observed in both sparse and dense object configurations. Furthermore, we observed that the error drops significantly (to 0.2 cm) in case only fixed joints are included. These results show that that with increasing complexity of physical object relations, learning becomes more challenging.



(a)

(b)

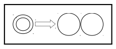
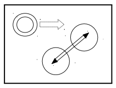
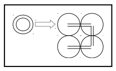
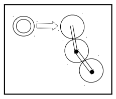
Figure 4: (a) The top figure shows the initial configuration where the end-effector of the robot is shown in blue color. The middle and bottom figures provide the joint predictions after interacting with the objects, and ground-truth joint information, respectively. (b) Mean square errors of the complete system after the given number of timesteps are observed.

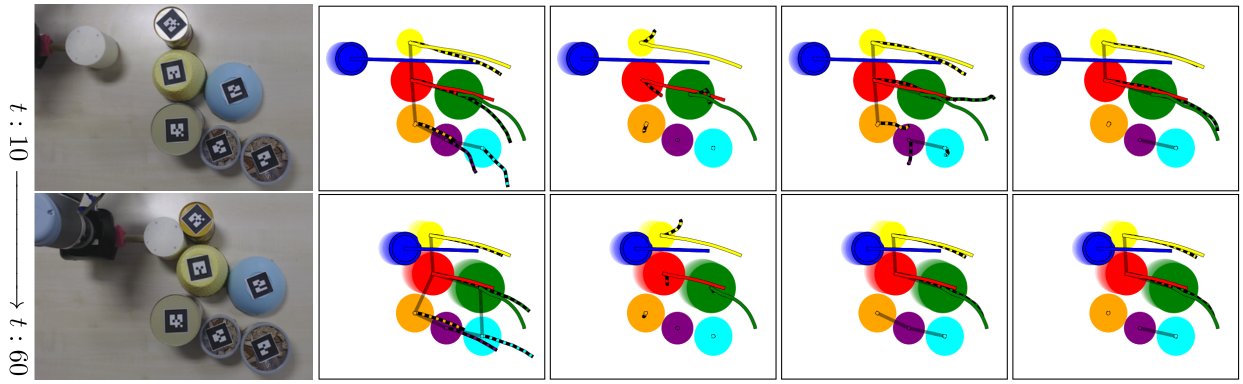
4.4 Effect predictions with partial state information

In this setup, at time t, belief regulation module uses scenes observed in the last 80 time-steps as input, and predicts the relations between nodes. 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. The system was evaluated in the dense configuration. While reporting the quantitative results, we considered the following alternative relation assignment strategies along with the proposed belief regulation approach: **Oracle** which assigns ground truth relations and used as a strong baseline, **All Fixed** which assumes all pairs of contacting objects have fixed joints between them and **No Joint** which assumes there are no joints between objects.

Our results can be seen in Fig. 4b, where the error is computed by predictions of the system at the reported time points. For example, in order to compute the error at time point 16, the system predicts the relations using the scenes from first 15 time points, and predicts the trajectories of all objects from 16 to 50. The prediction error for each object is computed by averaging the euclidean distances between corresponding points on the predicted and real trajectories of the corresponding objects, and these errors are averaged to obtain the final error at time point 16. The results are reported for 6, 8, and 9 objects in scenes with fixed-joints only and in all scenes. These results indicate that the belief regulation increases the performance of system when the relations between objects are not provided. This increase becomes more significant with the increasing experience of the robot through its interaction. The more it interacts, the system collects more observation from the environment, better predicting the joint types and therefore future trajectories of the objects. Note that, the effect predictions performed by the predicted relations were always worse than the predictions performed by the correct relations due to the errors made in relation predictions. This was an expected result as the robot makes effect predictions some of the objects that it had no interaction experience

Table 1: Interpretation of results in a number of ambiguous setups.

Example setup	Outcome Analysis
	An effect of horizontal translation of the group is generated when objects are connected with fixed or prismatic joints.
	A similar effect on the top object is generated when the objects are connected with a revolute joint, or they are not connected at all.
	The same effect of rigid body motion of the group is generated by joints that connect the objects in different ways.
	The same effect is observed in case the prismatic and revolute joints swap.



(a) Demo (b) All Fixed Relations (c) No Relations (d) Predicted Relations (e) Oracle

Figure 5: A scene from the real worlds experiments and the predictions on the remaining trajectories of the objects with predicted joint relations and alternative relation assignment approaches at specific time-steps are provided. Solid and dotted lines indicate the ground truth and predicted trajectories.

before, and therefore could not possibly make reliable predictions on their relations with other objects. Still, the belief propagation performed better compared to the other models that did not update their beliefs on object relations from interaction history.

Note that the system might suffer from ambiguities in predicting joint relations from its interaction experience. For example, a group of objects that form a rigid body through different set of connections would behave same in response to push action. Fig. 4a provides a snapshot that was observed during experiments where the robot started interaction with 8 objects placed on a grid. second row shows the joints predicted by our system and third row shows the ground truth. 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 did 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 more ambiguous cases such as the ones shown in Table 1.

4.5 Real World Verification

In this section, we provide the results obtained in the real world. For this, the prediction model trained in the simulator is directly transformed 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 so that the movement of an object causes the whole group to move together while keeping the relative positions constant.A top-down oriented RGB camera with 1920×1080 resolution was placed above the scene, ARTags were placed on the objects and ar_track_alvar[5] software was used to track the objects during robot action executions.

Our prediction system was tested in two different environments. In the first environment, 6 objects were placed as a group as shown in Fig. 5, where some objects were connected to each other with the underlying cardboards. The top left 3 objects and the bottom right pair were connected to each other in this particular case. A more challenging configuration was formed in the second environment, where 7 objects were placed in two separate groups (Fig. 6).

Fig. 5 demonstrates the prediction results of our system where joint information was either provided with relation assignment strategies described in Section 4.4 or predicted through belief regulation network. A straight push motion was executed by the robot and the object positions at times 10 and 60 were provided. The actual trajectories observed until the end of the execution were provided with solid lines. The expected object trajectories predicted by the models at time steps 10 and 60 were shown with dashed lines. Note that the colors of the objects and lines match each other for the convenience of the reader. Additionally, the transparency around the objects show the previously followed trajectories and the blue circle corresponds to the gripper of the robot. The first observation is that given correct joint information, the model made almost perfect predictions on the trajectories of objects. When no joints strategy was followed for the effect prediction, the model predicts that objects are pushed aside. When all fixed strategy was

[5] http://wiki.ros.org/ar_track_alvar

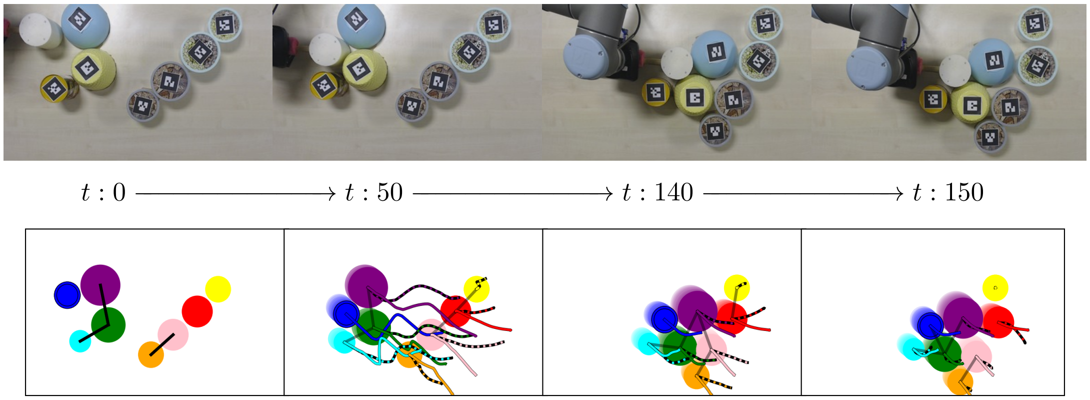


Figure 6: A scene from the real worlds experiments (on top) and its corresponding illustration of the predictions on the joint relations between the objects and the remaining trajectories of them at given time-steps as provided. Solid and dotted lines indicate the ground truth and predicted trajectories.

followed, the model predicts some motions on objects that are stationary. Finally, when belief propagation was used, initially, the model predicts trajectories similar to all fixed strategy, but after seeing the independent motions of upper three object group, it corrects the joint relations, and predicts correct trajectory successfully.

In experiment 2, the three objects that were initially placed next to the end effector of the robot, and the two objects that were placed at the bottom right were attached with fixed joints Fig. 6. Also see the connecting lines in the ground truth illustration at the bottom row. The 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 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 $t : 150$.

In order to further evaluate the performance of our method in various real-world conditions, we generated 15 different setups that include 4 to 6 objects where 1 to 3 of them were attached to each other. A straight motion of $40cm$ was applied towards these objects that were placed in different locations. Prediction error was computed by computing the distance between the predicted and actual final positions of the objects. As a result, while the objects moved $18cm$ in average, our model achieved an average error of $6.33cm$ in predicting their final positions. Although in some cases incorrect joint predictions caused failures in predicting the movement direction of interacted objects, our model performed well considering the average diameter of $12cm$ of the objects and our direct transfer strategy from the simulation.

5 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 future, we aim to study on generating goal directed action trajectories that balance the trade-off between exploration and exploitation.

References

[1] Emre Ugur, Erhan Oztop, and Erol Sahin. Goal emulation and planning in perceptual space using learned affordances. *Robotics and Autonomous Systems*, 59(7-8):580–595, 2011.

[2] Emre Ugur and Justus Piater. Bottom-up learning of object categories, action effects and logical rules: From continuous manipulative exploration to symbolic planning. In *ICRA*, pages 2627–2633. IEEE, 2015.

[3] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *NIPS*.

[4] Michael B Chang, Tomer Ullman, Antonio Torralba, and Joshua B Tenenbaum. A compositional object-based approach to learning physical dynamics. In *ICLR*, 2017.

[5] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *NIPS*.

[6] Arunkumar Byravan and Dieter Fox. SE3-Nets: Learning rigid body motion using deep neural networks. In *ICRA*, pages 173–180, 2017.

[7] Varun Jampani, Raghudeep Gadde, and Peter V Gehler. Video propagation networks.

[8] Yunzhu Li, Jiajun Wu, Jun-Yan Zhu, Joshua B Tenenbaum, Antonio Torralba, and Russ Tedrake. Propagation networks for model-based control under partial observation. *arXiv preprint arXiv:1809.11169*, 2018.

[9] Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013.

[10] Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search.

[11] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning.

[12] Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. In *ICML*, 2016.

[13] Roozbeh Mottaghi, Hessam Bagherinezhad, Mohammad Rastegari, and Ali Farhadi. Newtonian scene understanding: Unfolding the dynamics of objects in static images. In *CVPR*.

[14] Roozbeh Mottaghi, Mohammad Rastegari, Abhinav Gupta, and Ali Farhadi. ”what happens if...” learning to predict the effect of forces in images. In *ECCV*.

[15] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. In *ICLR*, 2016.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[17] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting.

[18] Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li F Fei-Fei, Josh Tenenbaum, and Daniel L Yamins. Flexible neural representation for physics prediction. In *NIPS*, pages 8813–8824, 2018.

[19] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *ICLR*, 2019.

[20] Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. Visual interaction networks: Learning a physics simulator from video.

[21] Sjoerd van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *ICLR*, 2018.