Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)
Sister Conferences Best Papers Track

Commonsense Reasoning to Guide Deep Learning for Scene Understanding (Extended Abstract)*

Mohan Sridharan[1] and **Tiago Mota**[2]
[1]School of Computer Science, University of Birmingham, UK
[2]Electrical and Computer Engineering, The University of Auckland, NZ
m.sridharan@bham.ac.uk, tmot987@aucklanduni.ac.nz

Abstract

Our architecture uses non-monotonic logical reasoning with incomplete commonsense domain knowledge, and incremental inductive learning, to guide the construction of deep network models from a small number of training examples. Experimental results in the context of a robot reasoning about the partial occlusion of objects and the stability of object configurations in simulated images indicate an improvement in reliability and a reduction in computational effort in comparison with an architecture based just on deep networks.

1 Motivation

Consider a robot[1] reducing clutter by clearing away toys arranged by children in different configurations. It is difficult to provide this robot many labeled examples of the objects and object configurations that it has to reason about. Also, the robot has to reason with different descriptions of incomplete domain knowledge and uncertainty. This includes qualitative descriptions of commonsense knowledge, e.g., statements such as "structures with a large object on a small object are typically unstable" that hold in all but a few exceptional circumstances. At the same time, the robot uses algorithms for sensing and actuation that model uncertainty probabilistically. Furthermore, any humans participants may not have the time and expertise to provide comprehensive feedback.

As motivating examples, we consider the visual scene understanding tasks of estimating the *partial occlusion* of objects and the *stability* of object configurations from images. Deep networks and the associated algorithms are the state of the art for these (and other such) problems in robotics and AI. These algorithms require many labeled training examples, are computationally expensive, and their operation is difficult to understand. Our architecture seeks to address these challenges by exploring the interplay between representation, reasoning, and learning. We limit perceptual input to RGB-D images of simulated scenes such as Figure 1(right), a small number of which are used as training data with occlusion labels for objects and stability labels for object structures. We assume that the robot knows the grounding (i.e., meaning in the real world) of words such as "above" and "left_of" that describe spatial relations between objects. The robot's domain knowledge also includes statements encoding defaults, constraints, and domain dynamics (more details later).

For any given image, our architecture first attempts to perform the estimation tasks by non-monotonic logical reasoning with incomplete commonsense domain knowledge and the spatial relationships extracted between objects in the image. If it is unable to do so (or provides incorrect labels on training data), it automatically identifies relevant regions of interest in the image. These regions are mapped to the desired labels by a deep network trained using similar regions extracted from the training data. The labeled examples are also used to train decision trees and incrementally learn previously unknown state constraints that are used for subsequent reasoning. Experimental results show a marked improvement in accuracy and computational efficiency in comparison with an architecture that only uses deep networks, and provides insights about the interplay between reasoning and learning; for complete details, see [Mota and Sridharan, 2019a].

2 Related Work

Scene understanding includes many estimation and prediction problems for which deep networks provide state of the art performance. For instance, a Convolutional Neural Network (CNN) has been used to predict the stability of structures [Lerer *et al.*, 2016], and movement of colliding objects [Wu *et al.*, 2015]. The training of CNNs and other deep networks requires many labeled examples and considerable computationally resources, and their operation is difficult to understand [Zhang *et al.*, 2016]. Since labeled examples are not readily available in many domains, deep networks have been trained using physics engines [Wagner *et al.*, 2018] or prior (domain) knowledge [Sünderhauf *et al.*, 2018]. The structure of deep networks has also been used to constrain learning, e.g., relational frameworks that pair objects with queries that need to be answered [Santoro *et al.*, 2017]. However, these methods do not exploit commonsense domain knowledge or the coupling between reasoning and learning.

Research in AI has developed theories and algorithms that enable agents to reason with commonsense domain knowl-

*Full paper was a Best Paper Award finalist at *Robotics: Science and Systems* conference [Mota and Sridharan, 2019a].

[1]Terms "robot", "learner", and "agent" used interchangeably.

edge. For scene understanding, domain knowledge often includes the grounding of spatial relations (e.g., "in" and "above") and axioms governing domain dynamics. Methods have been developed to reason about and learn spatial relations between objects [Jund *et al.*, 2018] and deep networks have been used to infer these spatial relations using images and natural language expressions [Paul *et al.*, 2016]. There is also a rich history of methods for learning domain knowledge, e.g., refining first-order logic models of actions [Gil, 1994], incrementally learning domain axioms using non-monotonic logical reasoning and relational reinforcement learning [Sridharan and Meadows, 2018], and frameworks for interactive task learning [Chai *et al.*, 2018]. Our architecture draws on these insights and exploits the inter-dependencies between reasoning and learning in the context of scene understanding.

3 Proposed Architecture

Figure 1(left) shows our architecture in the context of a *Robot Assistant* (RA) domain, with a simulated robot estimating the occlusion of objects and the stability of object structures, and rearranging objects to reduce clutter. Spatial relations between objects in RGB-D images are grounded using our prior work [Mota and Sridharan, 2018]. An object is occluded if any fraction of its frontal face is hidden by another object; a structure is unstable if any object in it is unstable. Decision tree induction maps object attributes and spatial relations to the target labels; axioms representing previously unknown state constraints are constructed from these trees. Learned constraints are encoded in an Answer Set Prolog (ASP) program along with incomplete commonsense domain knowledge and the spatial relations. If ASP-based reasoning provides the desired labels, the image is not analyzed further. Otherwise, an attention mechanism identifies the image's Regions of Interest (ROIs), and a CNN is trained to map these ROIs to labels. We briefly describe these components below; for more details, see [Mota and Sridharan, 2019a].

3.1 Knowledge Representation with ASP

To represent and reason with incomplete domain knowledge, we use ASP, a declarative language that can represent recursive definitions, defaults, causal relations, and language constructs difficult to express in classical logic formalisms. ASP encodes *default negation* and *epistemic disjunction*, i.e., each literal can be true, false or unknown. It supports non-monotonic logical reasoning, i.e., adding a statement can reduce the set of inferred consequences, aiding in the recovery from errors due to reasoning with incomplete knowledge. Modern ASP solvers support efficient reasoning in large knowledge bases with incomplete knowledge, and are used by an international community [Erdem *et al.*, 2016].

A domain's description in ASP comprises a *system description* $\mathcal{D}$ and a *history* $\mathcal{H}$. $\mathcal{D}$ comprises a *sorted signature* Σ and axioms. Σ includes *basic sorts*; *statics*, i.e., domain attributes that do not change over time; *fluents*, i.e., domain attributes whose values can be changed; and actions. Domain attributes and actions are described as relations in terms of their arguments' sorts. In the RA domain, sorts include $object$, $robot$, $relation$, $surface$, and $step$ for temporal reasoning. Statics include some object attributes such as $obj\_size(object, size)$. Fluents include spatial relations between two objects, i.e., $obj\_relation(relation, object, object)$, and relations such as $in\_hand(robot, object)$, i.e., a particular object is in the robot's grasp. Actions of the RA domain include $pickup(robot, object)$ and $putdown(robot, object)$, and $holds(fluent, step)$ is a predicate implying that a particular fluent holds at a particular timestep. The RA domain's axioms model the domain's dynamics in the form of causal laws, state constraints, and executability conditions:

$$holds(in\_hand(robot, object), I+1) \leftarrow occurs(pickup(robot, object), I) \quad (1a)$$

$$holds(obj\_relation(above, A, B), I) \leftarrow holds(obj\_relation(below, B, A), I) \quad (1b)$$

$$\neg occurs(pickup(robot, object), I) \leftarrow holds(in\_hand(robot, object), I) \quad (1c)$$

The axioms also encode default statements such as "structures with larger objects on smaller objects are typically unstable". Finally $\mathcal{H}$ includes records of observations received and actions executed by the robot. To reason with domain knowledge, we construct the ASP program $\Pi(\mathcal{D}, \mathcal{H})$; planning, diagnostics, and inference are reduced to computing *answer sets* of Π, which represent beliefs of the robot associated with Π [Gelfond and Kahl, 2014]. The ASP program of the RA domain is in our repository [Mota and Sridharan, 2019b].

3.2 Decision Tree Induction

Previously unknown state constraints are learned using a decision tree induction algorithm that splits nodes based on the potential information gain. The spatial relations between scene objects and the attributes of objects in 50% of the labeled training samples form the nodes of the tree, and the corresponding labels form the leaf nodes. Any branch of the tree in which the leaf represents a precision higher than 95% is used to construct candidate axioms that are validated using the other 50% of the labeled examples. The effect of noise is reduced by repeating the learning and validation steps (100 times) and only retaining axioms learned more than a minimum number of times. In the RA domain, separate decision trees are learned for stability and occlusion estimation, e.g., the gray and blue branch in Figure 2 encodes: $\neg stable(A) \leftarrow obj\_relation(above, A, B), \quad obj\_surface(B, irregular)$, i.e., any object above an object with an irregular surface is unstable. Any learned axioms are merged with the existing axioms (as appropriate) and used for subsequent reasoning.

3.3 Attention Mechanism

When ASP-based reasoning cannot assign labels to objects in an image, the attention mechanism identifies and directs attention to regions of interest (ROIs) that contain information relevant to the task at hand. It first identifies each axiom in the ASP program whose head corresponds to a relation of interest. For instance, if the task is to estimate the occlusion of object structures, each axiom whose head describes whether an object is occluded or not is considered. The relations in the

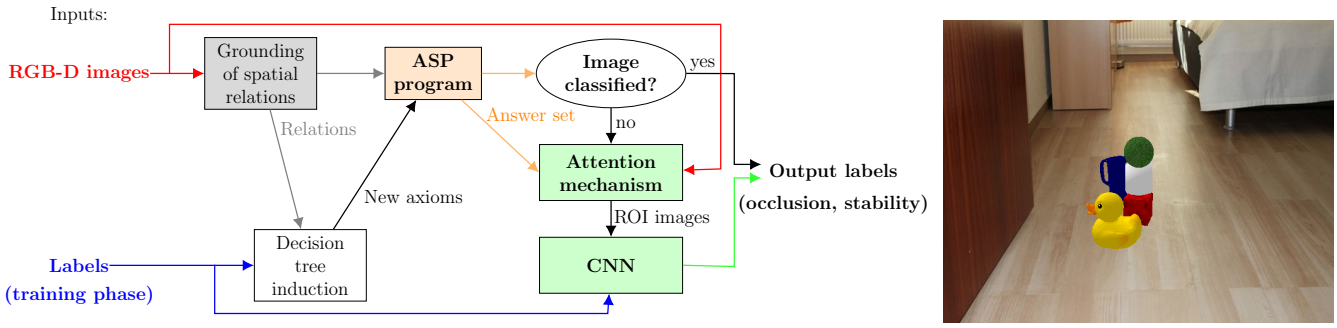

Figure 1: (Left) architecture combines the complementary strengths of non-monotonic logical reasoning, deep learning, and decision tree induction, for scene understanding; (Right) simulated scene with toys; robot has to reason about occlusion and stability.

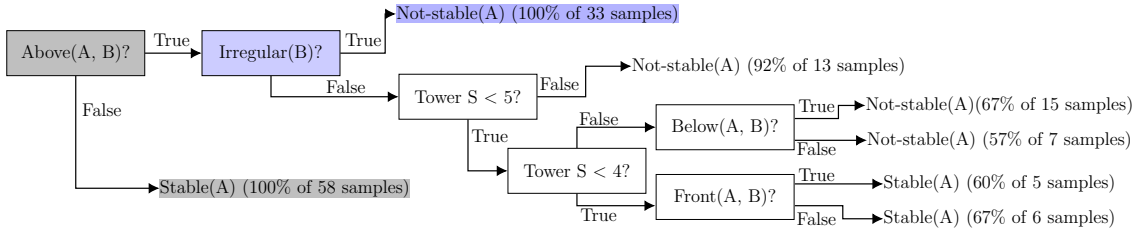

Figure 2: A decision tree constructed from some labeled examples. Highlighted branches are used to construct previously unknown axioms.

body of each such axiom are used to identify ROIs considered for further processing; the remaining image regions are not analyzed because they are unlikely to provide useful information. For instance, to estimate stability in Figure 1(right), the attention mechanism considers the stack comprising the red cube, white cylinder, and the green ball, since they satisfy the relevant relation *above*—the other two objects (yellow duck, blue pitcher) are disregarded. Any image may contain multiple ROIs, each with one or more objects.

3.4 Convolutional Neural Networks

The pixels in the ROIs identified by the attention mechanism serve as input to a deep network, which when trained is considered to model previously unknown information relevant to the task at hand. We explored two variants of a CNN, Lenet [LeCun *et al.*, 1998] and Alexnet [Krizhevsky *et al.*, 2012], with the sigmoid activation function and the Adam optimizer in the TensorFlow implementation [Abadi *et al.*, 2016]. The training dataset comprises image ROIs in the form of suitably rescaled RGB images, and the labels to be assigned to objects and structures in the ROIs. The CNN's parameters (e.g., weights, learning rate) are tuned to learn the mapping between the pixels and labels. The number of epochs was chosen as the stopping criteria to compare networks learned with and without the attention mechanism. The learned CNNs assign labels to ROIs or the entire test image to which ASP-based reasoning is unable to assign labels. The related code is in our repository [Mota and Sridharan, 2019b].

4 Experimental Setup and Results

In this section, we summarize some experimental results; please see [Mota and Sridharan, 2019a] for more details.

To simulate experiments in a dynamic domain in which a large number of training samples are not available, we used the Bullet real-time physics engine to generate 6000 labeled images for estimating occlusion and stability of objects. Each image had ROIs with up to five objects with different colors, textures and shapes. The objects included cylinders, spheres, cubes, a duck, and five household objects from the Yale-CMU-Berkeley dataset [Calli *et al.*, 2015]. We considered different arrangements of these objects, with the vertical alignment randomized to create a stable or an unstable arrangement. Other parameters, e.g., spread between objects, lighting, orientation etc, were also randomized to create scenes with complex, partial, or no occlusion. Also, we removed some state constraints related to stability and occlusion from the ASP program. A second dataset was derived from this dataset to simulate the attention mechanism's operation, i.e., only pixels in the relevant ROIs were considered for analysis. CNNs trained using the two datasets were compared as a function of the amount of training data and the complexity of the networks. Occlusion is estimated for each object (i.e., maximum of five outputs per ROI) and stability is estimated for the object structure (i.e., one output per ROI).

The performance measures were the accuracy of the labels assigned to objects and object structures, and the precision and recall of discovering previously unknown axioms. All claims were tested for statistical significance as appropriate. Lenet and Alexnet architectures without the commonsense reasoning and attention mechanism modules, i.e., trained on the RGB-D input images, were used as the baselines.

Figure 3 indicates that using commonsense reasoning to guide deep learning improves the estimation accuracy of the deep networks. Training and using the deep networks with only relevant ROIs of images that cannot be processed by

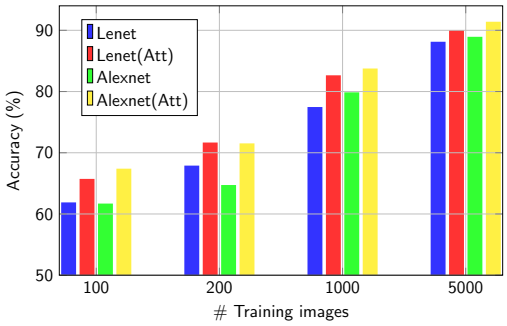

Figure 3: Accuracy of Lenet and Alexnet with and without commonsense reasoning and the attention mechanism. Our architecture improves accuracy in comparison with the baselines.

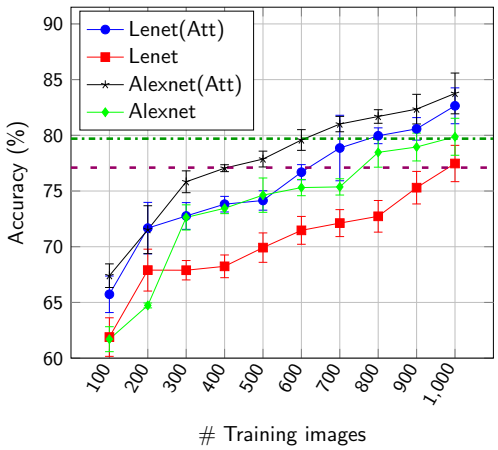

Figure 4: Accuracy of Lenet and Alexnet with and without the attention mechanism and commonsense reasoning. Any desired accuracy is achieved with a smaller training set.

commonsense reasoning (bars denoted with "Att" in legend) simplifies learning and makes it easier to learn an accurate mapping between inputs and outputs, resulting in higher accuracy than the baselines for any given number of training images. The improvement is more pronounced when the training set is smaller, but there is improvement at all training dataset sizes considered in our experiments.

Figure 4 shows that using the attention mechanism and reasoning with commonsense knowledge helps achieve any desired level of accuracy with much fewer training examples. The purple dashed (horizontal) line in Figure 4 indicates that the baseline Lenet needs $\approx$ 1000 images to reach an accuracy of 77%, whereas Lenet(Att) only needs $\approx$ 600. A similar difference is observed between Alexnet and Alexnet(Att) for $\approx$ 80% accuracy—the dark green dash-dotted (horizontal) line in Figure 4. In other words, the use of commonsense knowledge helps train deep networks with fewer examples, reducing both the computational and storage requirements.

Table 1 indicates the ability to learn previously unknown axioms. Errors are mostly variants of the target axioms that are not in the most generic form, i.e., they have irrelevant literals but are not wrong. The lower precision and recall with defaults is because it is challenging to distinguish between defaults and their exceptions. We do not describe it here,

Axiom type	Precision	Recall
Unknown (normal)	98%	100%
Unknown (default)	78%	62%

Table 1: Precision and recall for previously unknown axioms (normal, default) using decision tree induction.

but work in our group indicates that reasoning with commonsense knowledge and decision trees provides (at least partial) explanations for the decisions made by the architecture [Riley and Sridharan, 2019; Sridharan and Meadows, 2019].

Finally, we evaluated the robot's ability to compute minimal plans to pickup and clear particular objects. The number of plans computed when the learned axioms were included in the ASP program was much smaller than when the axioms were not included; the learned axioms helped eliminate certain paths in the transition diagram. In one scene, with all the axioms the robot computed three plans; all were minimal and correct. With some axioms missing, the robot found as many as 64 plans, many of which were incorrect. A plan was considered to be correct if executing it (in simulation) resulted in the corresponding goal being achieved.

5 Discussion and Conclusions

Deep networks are the state of the art for many tasks in robotics and AI, but they require large training datasets and considerable computational resources, and make it difficult to understand their operation. Our architecture seeks to address these limitations by integrating the principles of nonmonotonic logical reasoning with commonsense knowledge, decision tree induction, and deep learning. Commonsense knowledge is available in almost every domain—in fact, such knowledge is often used to determine the structure and parameters of the deep networks. Our architecture exploits this knowledge to simplify learning, focusing on aspects of the domain not encoded by the existing knowledge. A more accurate mapping is thus learned between inputs and outputs using a smaller set of labeled examples. Experimental results indicate that our architecture improves accuracy, especially when large labeled training datasets are not readily available, and reduces storage and computation requirements. In the future, we will explore the interplay between reasoning and learning to better understand the operation of deep network models, building on work that uses relational logical structures to explain decisions, beliefs, and experiences [Sridharan and Meadows, 2019]. Furthermore, we will enable the learning of different types of domain knowledge [Sridharan and Meadows, 2018], and reason at different resolutions [Sridharan *et al.*, 2019] to test this architecture on physical robots.

Acknowledgements

This work was supported in part by the Asian Office of Aerospace Research and Development award FA2386-16-1-4071 the U.S. Office of Naval Research Science of Autonomy Award N00014-17-1-2434. Opinions and conclusions described in this paper are those of the authors.

References

[Abadi *et al.*, 2016] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. TensorFlow: Large-scale Machine Learning on Heterogeneous Distributed Systems. Technical report, https://arxiv.org/abs/1603.04467, 2016.

[Calli *et al.*, 2015] Berk Calli, Aaron Wallsman, Arjun Singfh, and Siddhartha S. Srinivasa. Benchmarking in Manipulation Research. *IEEE Robotics and Automation Magazine*, (September):36–52, 2015.

[Chai *et al.*, 2018] Joyce Y. Chai, Qiaozi Gao, Lanbo She, Shaohua Yang, Sari Saba-Sadiya, and Guangyue Xu. Language to Action: Towards Interactive Task Learning with Physical Agents. In *International Joint Conference on Artificial Intelligence (IJCAI)*, Stockholm, Sweden, July 13-19, 2018.

[Erdem *et al.*, 2016] Esra Erdem, Michael Gelfond, and Nicola Leone. Applications of Answer Set Programming. *AI Magazine*, 37(3):53–68, 2016.

[Gelfond and Kahl, 2014] Michael Gelfond and Yulia Kahl. *Knowledge Representation, Reasoning and the Design of Intelligent Agents*. Cambridge University Press, 2014.

[Gil, 1994] Yolanda Gil. Learning by Experimentation: Incremental Refinement of Incomplete Planning Domains. In *International Conference on Machine Learning*, pages 87–95, New Brunswick, USA, July 10-13, 1994.

[Jund *et al.*, 2018] Philipp Jund, Andreas Eitel, Nichola Abdo, and Wolfram Burgard. Optimization Beyond the Convolution: Generalizing Spatial Relations with End-to-End Metric Learning. In *International Conference on Robotics and Automation*, 2018.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Lerer *et al.*, 2016] Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. Technical report, https://arxiv.org/abs/1603.01312, 2016.

[Mota and Sridharan, 2018] Tiago Mota and Mohan Sridharan. Incrementally Grounding Expressions for Spatial Relations between Objects. In *International Joint Conference on Artificial Intelligence*, pages 1928–1934, 2018.

[Mota and Sridharan, 2019a] Tiago Mota and Mohan Sridharan. Commonsense Reasoning and Knowledge Acquisition to Guide Deep Learning on Robots. In *Robotics Science and Systems*, Freiburg, Germany, June 22-26, 2019.

[Mota and Sridharan, 2019b] Tiago Mota and Mohan Sridharan. Software related to the paper, 2019. https://github.com/tmot987/Scenes-Understanding.

[Paul *et al.*, 2016] Rohan Paul, Jacob Arkin, Nicholas Roy, and Thomas Howard. Efficient Grounding of Abstract Spatial Concepts for Natural Language Interaction with Robot Manipulators. In *Robotics: Science and Systems*, Ann Arbor, USA, June 18-22, 2016.

[Riley and Sridharan, 2019] Heather Riley and Mohan Sridharan. Integrating Non-monotonic Logical Reasoning and Inductive Learning with Deep Learning for Explainable Visual Question Answering. In *Frontiers in Robotics and AI, special issue on Combining Symbolic Reasoning and Data-Driven Learning for Decision-Making*, 6:125, 2019.

[Santoro *et al.*, 2017] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Tim Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017.

[Sridharan and Meadows, 2018] Mohan Sridharan and Ben Meadows. Knowledge Representation and Interactive Learning of Domain Knowledge for Human-Robot Collaboration. *Advances in Cognitive Systems*, 7:77–96, December 2018.

[Sridharan and Meadows, 2019] Mohan Sridharan and Benjamin Meadows. Towards a Theory of Explanations for Human-Robot Collaboration. *Kunstliche Intelligenz*, 33(4):331–342, December 2019.

[Sridharan *et al.*, 2019] Mohan Sridharan, Michael Gelfond, Shiqi Zhang, and Jeremy Wyatt. REBA: A Refinement-Based Architecture for Knowledge Representation and Reasoning in Robotics. *Journal of Artificial Intelligence Research*, 65:87–180, May 2019.

[Sünderhauf *et al.*, 2018] Niko Sünderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, et al. The limits and potentials of deep learning for robotics. *The International Journal of Robotics Research*, 37(4-5):405–420, 2018.

[Wagner *et al.*, 2018] Misha Wagner, Hector Basevi, Rakshith Shetty, Wenbin Li, Mateusz Malinowski, Mario Fritz, and Ales Leonardis. Answering Visual *What-If* Questions: From Actions to Predicted Scene Descriptions. In *Visual Learning and Embodied Agents in Simulation Environments Workshop at ECCV*, Munich, Germany, September 9, 2018. https://arxiv.org/abs/1809.03707.

[Wu *et al.*, 2015] 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. In *Advances in neural information processing systems*, pages 127–135, 2015.

[Zhang *et al.*, 2016] Renqiao Zhang, Jiajun Wu, Chengkai Zhang, William T Freeman, and Joshua B Tenenbaum. A Comparative Evaluation of Approximate Probabilistic Simulation and Deep Neural Networks as Accounts of Human Physical Scene Understanding. Technical report, https://arxiv.org/abs/1605.01138, 2016.