UNSUPERVISED DEEP LEARNING OF STATE REPRESENTATION USING ROBOTIC PRIORS

Timothee LESORT & David FILLIAT
U2IS, ENSTA ParisTech, Inria FLOWERS, Université Paris-Saclay
828 bd des Maréchaux
91762 Palaiseau cedex, France
{timothee.lesort,david.filliat}@ensta-paristech.fr

ABSTRACT

Our understanding of the world depends highly on how we represent it. Using background knowledge about its complex underlying physical rules, our brain can produce intuitive and simplified representations which it can easily use to solve problems. The approach of this paper aims to reproduce this simplification process using a neural network to produce a simple low dimensional state representation of the world from images acquired by a robot. As proposed in (Jonschkowski & Brock, 2015), we train the neural network in an unsupervised way, using the *a priori* knowledge we have about the world as loss functions called "robotic priors" that we implemented through a siamese network. This approach has been used to learn a one dimension representation of a Baxter head position from raw images. The experiment resulted in a 97,7% correlation between the learned representation and the ground truth, and show that relevant visual features form the environment are learned.

1 INTRODUCTION

The environment we live in is ruled by complex physical laws. However humans are likely to interact with it without any detailed knowledge of these laws. The human brain constructs simple models of the world in order to come up with an easy, though approximate, understanding of it.

This paper aims to reproduce this behavior for robots. We want to build a simple representation of the world that retains enough information to make a machine able to use it to interact afterwards i.e. to perform an assigned task. Finding such a minimal representation (e.g., the position of an object extracted from an image) is the standard way to implement behaviors in robots. However, this is most of the time done in a task specific and supervised way. In this paper, we want to learn such representation without supervision, based on generic learning objectives.

This representation is learned using deep learning. The deep neural network is trained by using images of robot experiences in a given environment and has to estimates for each image a state which is the representation we want to learn. Instead of using a ground truth for supervised training, we make use of an approach that ensures consistency between the states representation. For this purpose, the states are constrained by "robotics priors" (Jonschkowski & Brock, 2015) which are an expression of the knowledge we have about physics.

The main contribution of this paper is the use of the robotics prior approach in a siamese network to train a deep convolutional neural network. The network is trained with images of the robot environment, information on the actions performed by the robot and rewards defining a task. The neural network learns a state representation usable for the robot to perform this task. The resulting neural network also displays useful feature detectors for environment analysis that could be a basis for transfer learning to similar tasks.

2 RELATED WORK

- **Robotic Priors**
  The term of prior in Bayesian statistics refers to the prior probability distribution but like in the article from (Bengio et al., 2013), (Jonschkowski & Brock, 2014) and (Jonschkowski & Brock, 2015) we use this word as a reference to an *a priori* knowledge we have and not to a probability distribution. This knowledge comes from various domains which define several kind of priors : Task-Specific, Generic, and Robotic Priors.
  (Bengio et al., 2013) state that the key to successful state representation learning is the use of "many general priors about world around us" (cited by (Jonschkowski & Brock, 2015)). As noticed by (Jonschkowski & Brock, 2015), (Bengio et al., 2013) "proposed a list of generic priors for artificial intelligence and argue that refining this list and incorporating it into a method for representation learning will bring us closer to artificial intelligence " For these authors however those generic priors are too weak to be used in the robotics fields and stronger ones have to be defined to achieve an efficient learning : the robotics priors used in this paper. Those priors, in a way similar to the approach of (Scholz et al., 2014), are physically grounded, which means that they aim at building a representation of the world consistent with physics.
- **state representation**
  The goal of state representation learning is to find a mapping from a set of observations to a set of states that makes it possible to describe an environment at a given time with enough information to fulfill a given task. In our approach we impose a dimension of the state and use the priors to guide the neural network in learning task specific states representation in this given dimension. This is an alternative approach to selecting a state representation from a set (Seijen et al., 2014) and (Konidaris & Barto, 2009) or creating a auto-encoder to compress the information into a low dimension state (Lange et al.) ,(Watter et al., 2015), (Finn et al., 2016) and (van Hoof et al., 2016).
- **Unsupervised learning**
  Using priors with neural networks is an unsupervised way for training a neural network. This approach is a different, but similar from energy based methods (Lecun et al., 2006), auto-encoder or denoising auto-encoder (Vincent et al., 2010) to train a deep neural network. The training process does not use directly energy functions but more specific functions in order to have targeted representation of task relevant parameters. Using unsupervised learning for training deep neural network may, according to (Bengio, 2009), be efficient because when the neural network does not have information about what to learn precisely and what future learning tasks are "it would appear very rational to collect and integrate as much information as possible about this world so as to learn what makes it tick". On the other hand this way of training reduces overfitting risks.
- **Model Architecture**
  The method used for this paper involves a convolutional network whose architecture is inspired by (Krizhevsky et al., 2012). It makes it possible to create easy to train deep networks. Furthermore convnets are easier to use for training state visualization than GoogleNet (Szegedy et al., 2014) or ResNet (He et al., 2015) architecture. This architecture is coupled with Siamese networks like in (Chopra et al., 2005) or (Xing et al., 2003). Siamese network are used in this paper to impose constraints on learned representation in the implementation of robotic priors.

3 STATE REPRESENTATION LEARNING

The first challenge of state representation is to define which parameters are sufficient to characterize the state of the entire environment. For example, in a visually rich environment where only one object moves through time, the environment description depends only on the object position, which happens to be the only relevant parameter. The challenge is thus to learn what are the various relevant parameters. A second challenge is to find which of those parameters are truly interesting. For this we exploit a reward function. This function will give rewards for given states of the environment according to a defined task. This function gives the learning process a way to know which parameters

are relevant to the assigned task and which are not. The neural network has finally to map the relevant parameters into a state representation of a given dimension.
To evaluate this representation there are two main possibilities:

- evaluating if the representation is compatible with a ground truth
- determining if a reinforcement learning algorithm can use this learned representation to learn to perform the assigned task (Jonschkowski & Brock, 2015)

While the second approach is more objective, the first one is simpler and we use it in this paper as a proof of concept using the correlation (Eq : 1) between learned representation and a ground truth.

$$Corr(s, \hat{s}) = \frac{\mathbf{E}[(s - \mathbf{E}[s])(\hat{s} - \mathbf{E}[\hat{s}])]}{\sigma_s \sigma_{\hat{s}}} \,, \tag{1}$$

4 METHOD

4.1 ROBOTIC PRIORS

Robotic priors are used to provide the model to train with basic knowledge about the environment physical features. They add constraints to make the learned representation altogether consistent with simple physical and task specific rules. Each prior is formalized by a cost function implemented through a siamese network. By minimizing them, the model is trained according to the prior and can learn task-specific representation. The four priors we used are the one presented in (Jonschkowski & Brock, 2015). We will use the following notations:

- $I(t)$ is the image perceived at time t
- $s(t)$ is the state at time t and $\hat{s}(t)$ is its estimation.
- ϕ is a function which to an image $I(t)$ returns a state $s(t)$. $\hat{\phi}$ is its estimation
- $r(t)$ is the reward at time t
- $a(t)$ is the action done a time t
- D is the input data (images, actions, rewards)
- $\Delta s(t) = s(t+1) - s(t)$

The definitions of loss functions associated to priors and the attached assumption are as follows:

Temporal coherence Prior: *Two states close to each other in time are also close to each other in the state representation space.*

$$L_{Temp}(D, \hat{\phi}) = \mathbf{E}[\parallel \Delta\hat{s}_t \parallel^2] \,, \tag{2}$$

Proportionality Prior: *Two identical actions should give two proportional state variations.*

$$L_{Prop}(D, \hat{\phi}) = \mathbf{E}[(\parallel \Delta\hat{s}_{t_2} \parallel - \parallel \Delta\hat{s}_{t_1} \parallel)^2 | a_{t_1} = a_{t_2}] \,, \tag{3}$$

Repeatability Prior: *Two identical actions should give similar state variations.*

$$L_{Rep}(D, \hat{\phi}) = \mathbf{E}[e^{-\|\hat{s}_{t_2} - \hat{s}_{t_1}\|^2} \parallel \Delta\hat{s}_{t_2} - \Delta\hat{s}_{t_1} \parallel^2 | \; a_{t_1} = a_{t_2}] \,, \tag{4}$$

Causality Prior: *Two states on which the same action gives two different rewards should not be close to each other in the state representation space.*

$$L_{Caus}(D, \hat{\phi}) = \mathbf{E}[e^{-\|\hat{s}_{t_2} - \hat{s}_{t_1}\|^2} \mid a_{t_1} = a_{t_2}, r_{t_1+1} \neq r_{t_2+1}] \,, \tag{5}$$

This last prior is the only one giving information about the task and helps discovering the underlying factors which give a reward.

4.2 SIAMESE NETWORKS

The training using the priors needs the simultaneous estimation of several states to perform the optimization process. At this end our approach uses siamese networks. They are neural networks which share all their parameters. With this method the cost functions can be applied on several states computed at the same time. Those cost functions requires to choose the right images as input for siamese networks. For example for applying temporal prior, two input images are chosen following each other in time. Two siamese networks are used to compute the state estimation for each image and applying the temporal cost function. The backward propagation can then be done by computing gradient based on the loss function. Another example would be to use proportionality prior in which two state variations are estimated with the state of two images. Therefore four images and four siamese networks are needed to compute the proportionality cost function. The figure 1 shows the global network architecture. An image is input on each network and the application of the cost function is used on all the outputs. The layers are shared among all the siamese networks.

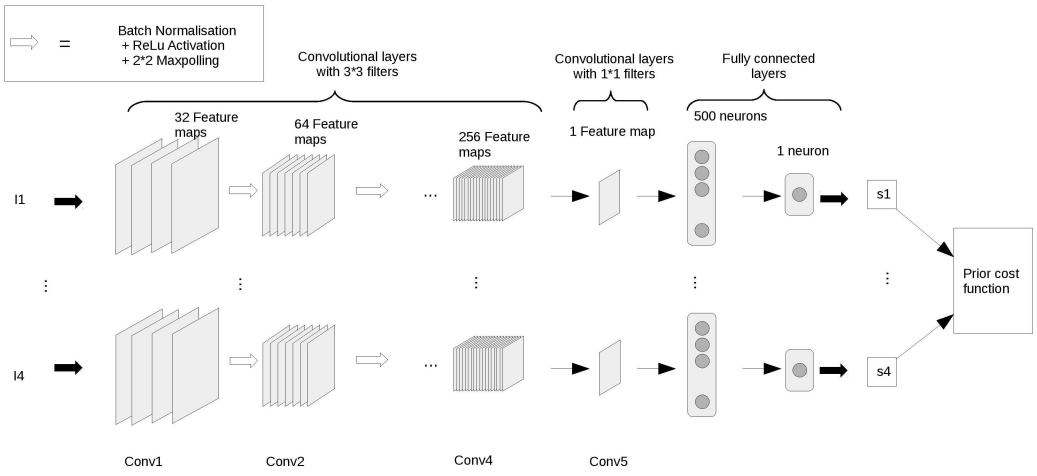


Figure 1: Illustration of the architecture with four Siamese networks

4.3 MODEL ARCHITECTURE

The architecture of the network uses four stacked convolution layers with 3*3 filters (inspired by (Simonyan & Zisserman, 2014)) with for each convolution layer, batch normalization, ReLu activation and 2*2 maxpooling (in that order). The convolution layers have 32-64-128-256 filters/Layer respectively. On the top of the network, there are two stacked fully connected layer (500 neurons and then one neurons for one dimension output). Between the last convolution layer with 3*3 filters and the fully connected layers, We insert a convolution layer with one 1*1 filters. This architecture helps the network to choose which feature map it wants to use to build its representation. This leads to a reduction of parameters in the fully connected layer by a factor of 256.

We use batch Normalization (Ioffe & Szegedy, 2015) for training, which helps to keep reasonable internal values and to make the training possible. In the experiment we make without batch normalization, the network is unable to learn the representation. Relu is used for fast learning and increasing the sparsity of neuron activation.

4.4 DATA

The data we used for training is a set of RGB images 200 pixels * 200 pixels which come from simulation. Those images are taken by the head camera of a Baxter robot. Images thus represent the front view of the robot and what it is able to see with its camera. The inputs images are normalized with 0 mean and 1 standard deviation before use, but for training we also add data augmentation to the images in order to improve robustness to noise and luminosity variations. To make the training

resistant to those perturbation, we add a random color filter to training sample like in (Krizhevsky et al., 2012) weighted with a Gaussian with random parameters. This transformation aims to make the representation invariant to the luminosity variation. We also add noise with the same mean and standard deviation than images. Those two data augmentation are added online during the training process. This method force the neural network to learn a lot of feature detector to make its representation robust.

4.5 TRAINING

The training has been done with Adagrad Duchi et al. (2011) with the following hyper-parameters: - Batch Size : 12 - Learning rate : 0.001 - weight Decay : 0 - Epoch :200 - iterations/Epoch : 10
The training is very fast in comparison with imageNet (Deng et al., 2009) classification training for example. Our understanding of this behavior is that the network does not need as many high level feature detectors as in classification because it is specialized in only one environment. A training without data augmentation is done for bootstrapping the neural network before the training with data augmentation.

5 EXPERIMENT

5.1 TASK AND ENVIRONMENT DESCRIPTION

This environment is produced by a simulation of a Baxter robot developed with the gazebo software. The robot is in front of a table (figure 2). The objective is to produce a representation that will make it possible to control the head joint position. The images are taken when the robot moves its head from right to left or left to right, a unit reward is obtained when the head is at maximum left or maximum right. In this context actions are defined by movement of the head between t and $t + 1$. The representation constructed by the neural network is in one dimension and should be consistent with the actual head position which is used as ground truth. The dataset based on the image generated is a set of 27 chronological series of images. Each series is composed of approximately 75 images. For each series the robot arms has a different position but only the state of the head joint change trough time. We know for each image at what time it has been taken and if the actual state gives a reward. We also know which action has been made between image at time t and image at time t+1. The actions are “move left” and “move right” with a certain angle. With these actions we can find pairs of images with same angle variation therefore compatible with proportionality and repeatability priors. Furthermore, with reward information we can find which images generate a reward with which action. Those images are then gathered to constitute an image set compatible with the causality cost function.

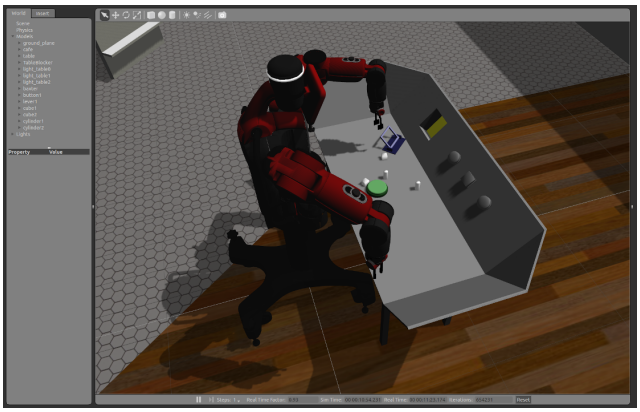


Figure 2: Illustration of the Baxter simulation used to generate data

During the training process, the sets of images compatible with a certain prior are randomly sampled for training inside 26 of the 27 series. The last series is used for validation.

5.2 LEARNING PROCESS

The training process is done by minimizing each of the cost functions but those functions are in conflict to impose their constraint. The result of the training is an equilibrium between cost functions minimization. The result of the sum of the priors costs is presented on 3 which shows the decreasing of the global cost. The value of each cost functions separately is on figure 4. It is not surprising that all the cost functions are not minimized the same way. For example, the temporal function aims at minimizing the distance between representations when causality aims at maximizing it.

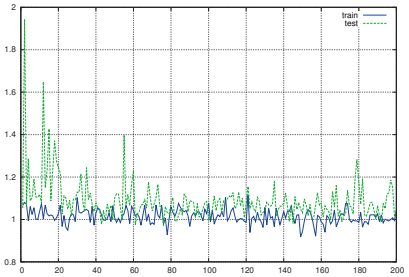


Figure 3: Sum of the cost functions at each epoch

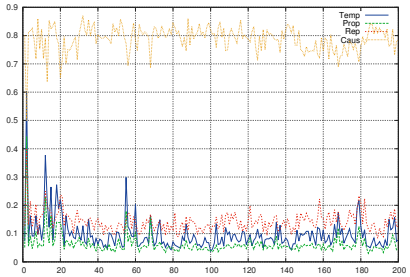
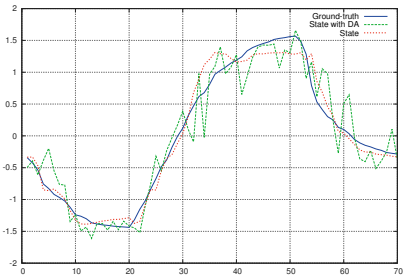


Figure 4: cost functions at each epoch

5.3 RESULT

The resulting state representation of the head position for the validation data, after training with the deep model presented above and the results after training with 1 fully connected layer model similar to the one used in (Jonschkowski & Brock, 2015) are on figure 5. The correlation computed between state representation learned for both models and ground truth are in table 1. Those results show that both models are able to learn a good state representation of the Baxter head position in the case where no noise is present. However, table 1 shows that the deep neural network is much more robust to noise and luminosity perturbation than the one-layer-network. The evolution at each epoch of the correlation during bootstrapping and training are in figure 6.



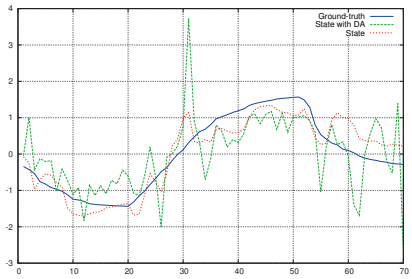


Figure 5: Comparison of the ground truth of the head position for each image of the validation set (Ground truth), the estimation of the state base on the original images (State) and the state based on the images with noise and random luminosity perturbations (State with DA). The left hand figure shows the result after training a deep neural network the right hand figure show the result after training a one layer fully connected neural network

	one Layer Network	Deep Network
without Data Augmentation	97.0 %	97.7 %
with Data Augmentation	61.7 %	96.4 %

Table 1: Influence of the neural network deepness on Correlation between learned representation and ground-truth on the validation set.

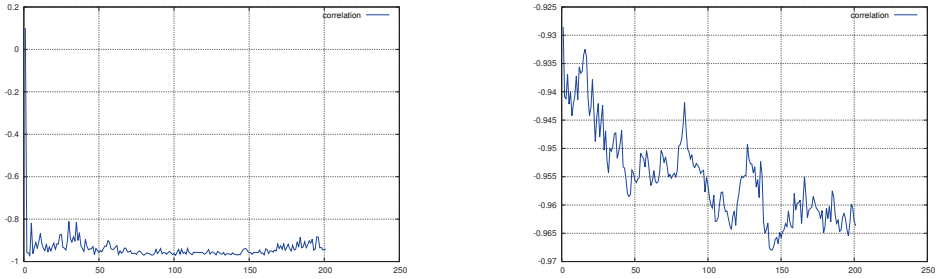


Figure 6: Those figures show the evolution of the correlation between ground-truth and learned state representation on the validation set. The left hand figure shows the correlation at each Epoch of the bootstrapping training. The right hand figure shows the correlation at each Epoch of the training with data augmentation. The final results in table 1 and 2 are the absolute values of the correlation

Beside the performance gain, our deep model makes it possible to learn relevant visual features that could be interesting for other tasks in a transfer learning scenario. For example, the left image of figure 7 shows that a button of the environment has been learned to be a good feature for the current task, but could obviously be used in other scenarios.

Regarding these features, using data augmentation makes it possible to train the neural network to use a larger part of the image. To illustrate this assumption we train the network with and without data augmentation to compare the activation on the last convolution layer. Those activation are shown on figure 7. We can see that the training without data augmentation makes the neural network use only the position of the blue button of the image when with data augmentation the neural network use much more pieces of information such as the table top border. Furthermore, Table 2 shows that training with data augmentation slightly increases the performance of the deep neural network.

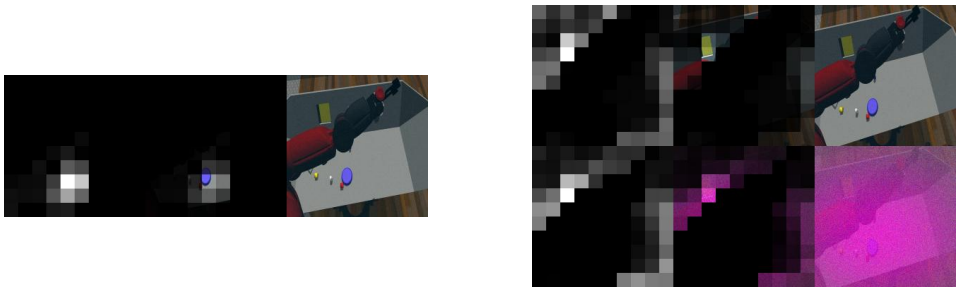


Figure 7: The figures show a representation of the activation produced by the neural network on its last convolution layer (10 pixels*10pixels). For each figure, on the left is presented a feature map for a given image , on the right the original image (200 pixels*200pixels) and in the middle the superposition of both images to show which part of the image produces activation. The left hand side image shows activation produce by the image after training without data-augmentation. The right hand side image furthermore compares activation between cases with and without data augmentation after training with data-augmentation.

train \ test	with Data Augmentation	without Data Augmentation
with Data Augmentation	96.4 %	97.7 %
without Data Augmentation	94.0 %	97.2 %

Table 2: Correlation results between learned representation and ground-truth on the validation set after training deep neural network with and without data augmentation.

6 DISCUSSION

This approach makes it possible to train a deep neural network to learn specialized feature detectors used to build state representation in an unsupervised way. Those trained feature detectors could be used or transferred for learning another state representation in this environment.

In the reported preliminary experiments, the simulation environment is not as rich as the real world, therefore the variability of input image is low. However, this approach should work with real images in order to make the neural network learn more specialized feature detectors. It will be tested in further experiments.

A limitation of our approach is the assessment of the the training quality. In the case presented in this paper, correlation between state representation and ground truth is a possible measurement of the training quality. On the other hand, had we tried to learn a representation in higher dimension, the correlation could have been irrelevant. Furthermore if the process is applied to a situation where ground truth is unavailable, the correlation cannot be measured. A possible method would be to use a reinforcement learning algorithm to measure if the learned representation is suitable to the task like in (Jonschkowski & Brock, 2015).

7 CONCLUSION

This approach provides us with evidences that a deep network trained by the method of robotics priors can learn state representations. This technique makes it possible to learn a one dimension representation and furthermore to train a network to be robust to both noise and luminosity perturbations. The next step to be done will be to learn more complex representations like objects positions in three dimensions, and to use those representations within a reinforcement learning process to check if a robot can use the learned representations to perform various tasks. The use of real image for training is also one of our goals.

ACKNOWLEDGMENTS

The authors would like to thanks Clement Masson for fruitful discussion and his help in generating the data used in this paper. This work is supported by the DREAM project[1] through the European Union Horizon 2020 research and innovation program under grant agreement No 640891.

REFERENCES

Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1), 2009. ISSN 1935-8237. doi: 10.1561/2200000006.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. ISSN 0162-8828. doi: doi.ieeecomputersociety.org/10.1109/TPAMI.2013.50.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July 2011. ISSN 1532-4435.

C. Finn, Xin Yu Tan, Yan Duan, T. Darrell, S. Levine, and P. Abbeel. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 512–519, May 2016. doi: 10.1109/ICRA.2016.7487173.

[1] http://www.robotsthatdream.eu

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

Rico Jonschkowski and Oliver Brock. State representation learning in robotics: Using prior knowledge about physical interaction. In *Proceedings of Robotics: Science and Systems*, July 2014.

Rico Jonschkowski and Oliver Brock. Learning state representations with robotic priors. *Autonomous Robots*, 39(3):407–428, 2015. ISSN 0929-5593.

George Konidaris and Andrew Barto. Efficient skill learning using abstraction selection. In *In Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. 2012.

Sascha Lange, Martin Riedmiller, and Arne Voigtlander. Autonomous reinforcement learning on raw visual input data in a real world application. doi: 10.1109/IJCNN.2012.6252823.

Yann Lecun, Sumit Chopra, Raia Hadsell, Fu Jie Huang, G. Bakir, T. Hofman, B. Schlkopf, A. Smola, and B. Taskar (eds. A tutorial on energy-based learning. In *Predicting Structured Data*. MIT Press, 2006.

Jonathan Scholz, Martin Levihn, Charles Lee Isbell, and David Wingate. A physics-based model prior for object-oriented mdps. In *ICML*, 2014.

Harm Seijen, Shimon Whiteson, and Leon Kester. Efficient abstraction selection in reinforcement learning. *Comput. Intell.*, 30(4):657–699, November 2014. ISSN 0824-7935.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

Herke van Hoof, Nutan Chen, Maximilian Karl, Patrick van der Smagt, and Jan Peters. Stable reinforcement learning with autoencoders for tactile and visual data. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2016, Daejeon, South Korea, October 9-14, 2016*, pp. 3928–3934, 2016. doi: 10.1109/IROS.2016.7759578.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, December 2010. ISSN 1532-4435.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 2746–2754. Curran Associates, Inc., 2015.

Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side-information. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 15*, pp. 505–512. MIT Press, 2003.