Autonomous behavior in the CL agent

Clément[1] and Kaushik Subramanian[1]

[1]Cogitai

September 14, 2018

1 Motivation

As mentioned in the CL Agent design document, *"Our CL setting differs from the RL setting in that there either does not exist a single observable reward signal from environment to maximize, or it is so sparse in such a complex environment that attempting to directly optimize it from conception will be fruitless".*

In consequence, a CL agent requires behaviors which do not attempt directly at optimizing a well-defined reward function, but instead must support continuous learning in a generic way, i.e. without (or with minimal) prior knowledge on the kind of tasks the agent will have to solve. We have identified the following desiderata of what we will call from now on the **autonomous behavior**:

- It should support the learning of existing tasks (in the form of Goal Configurations provided by the user) by attracting the agent to initial states where progress is expected.
- It should support off-policy learning through a behavior policy maximizing reward on average over all existing options.
- It should support learning in the sensory cortex through an exploration policy providing a an adequate database for the learning of sensory features.
- I should be able to generate new goals when no existing one can be practiced ("no man's land" states).
- It should automatically generate a curriculum of learning experiences
- Optionally, it should show an interesting behavior for the user and help her/him discovering new tasks for the robot.

This list of desiderata emphasizes that what we call autonomous behavior is actually more a collection of behavior policies with complementary objectives, supporting continual learning in heterogeneous parts of the system. An interesting research question is however to figure out if we could learn a general policy optimizing a reward function over a large set of features, possibly including meta-features such as uncertainty, prediction error or surprise (see section 3.2).

2 Literature review

This section is work in progress, only the first sub-section describes some papers.

Indicate for each approach which of these boxes they tick:

- How to improve skill policies?
- How to grow initiation skills?

| - Adversarial generation

- How to discovery new skills?
- How to explore?

| - Intrinsic motivation
| - prediction error

2.1 Curriculum learning

(Florensa et al., 2017) proposes a method quite similar to the one we have implemented, where the agent maintains a set of "good initial states" (states resulting in a medium reward between fixed parameters R_{min} and R_{max}). It starts from initial states close to the goal, iteratively samples new ones close to those already in the set and evaluates them through practice, keeping only those with reward bounded by R_{min} and R_{max}. However, they assume that the agent can be arbitrarily reset to any state at the beginning of each episode, which is not suitable for the physical world. It should be possible to adapt their method to avoid reset by taking advantage of our initial state classifier.

Another proposition for generating a curriculum is the Goal-GAN method from (Held et al., 2017). It actually shares many similarities with the previous approach by using a generative model with the criterion of a medium reward bounded by R_{min} and R_{max}. However they generate new goals instead of new initial states, using a generative adversarial network, where the discriminator is optimized to predict if the generated goal will satisfy the reward criterion or not.

(Sukhbaatar et al., 2017) also use adversarial training to generate a curriculum, but using self-play. The authors consider two agents (or rather, one agent with two minds), called Alice and Bob, where one proposes tasks while the other attempts to achieve them. For doing so, Alice first executes a policy. Then Bob starts from where Alice stopped and attempts at coming back to the original position. The reward received by Bob is inversely proportional to the time it spent solving the task ($r_B = -\gamma t_B$). The same occurs for Alice, except that the time spent by Bob is added to her reward $r_A = \gamma max(0, t_B - t_A)$. That way, Alice is rewarded if Bob takes more time, but the negative term on her own time will encourage Alice not to take too many steps when Bob is failing. . This way, Alice is encouraged to push Bob past his comfort zone, but not give him impossible tasks, generating a curriculum of tasks which are achievable while challenging. However, their method is restricted to two classes of environment: those that are (nearly) reversible, or ones that can be reset to their initial state (at least once). The Robutler environement is nearly reversible though and their method is likely to be applicable in our case.

We see from this literature that curriculum learning can be applied to initial states (Florensa et al., 2017), goal states (Held et al., 2017) or both (Sukhbaatar et al., 2017). In the methods generating goal states, the agents use policies parameterized by those goal states.

We have already implemented a version of curriculum learning during previous sprints (described and evaluated in this Authorea document), dealing with the selection of initial states. The agent selects states which are at a reasonable distance to the goal (in terms of the norm classifier output) given its previous performances in achieving that goal. The mentioned document shows that this can speed up learning in certain conditions, e.g. filtering out initial states which are too far to the goal and therefore not suitable for an efficient learning. However it has been observed that this can sometimes impair generalization by focusing the agent on a too small number of initial states.

In section 3.1, we propose a method for generating a curriculum over existing tasks, together with preliminary results in an idealized case.

2.1.1 Other references on curriculum learning

- In supervised learning

- Curriculum learning (Bengio et al., 2009),
- Automated Curriculum Learning for Neural Networks (Graves et al., 2017). There are more like this, especially with LSTM
- Active Learning of Inverse Models with Intrinsically Motivated Goal Exploration in Robots (Baranes and Oudeyer, 2013)

- In RL

- From the Goal-GAN paper: (Kumar et al., 2010; Jiang et al., 2015), (Karpathy & Van De Panne, 2012), (Sharma & Ravindran, 2017), (Sukhbaatar et al., 2017)
- From "Alice & Bob" paper: Andrychowicz et al. (2017) form an implicit curriculum by using internal states as a target. Florensa et al. (2017) automatically generate a series of increasingly distant start states from a goal. Pinto et al. (2017) use an adversarial framework to perturb the 4 Under review as a conference paper at ICLR 2018 environment, inducing improved robustness of the agent. Held et al. (2017) propose a scheme related to our "random Alice" strategy2 .

2.2 Exploration

- Assuming a single stationary MDP: Rmax, E3, Thomson sampling (limited interest for us)

Optimistic exploration

Randomized exploration

Bayesian Exploration Bonus

2.2.1 In single-task (deep) RL:

- #Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning (Tang et al., 2017a)

- Incentivizing exploration in reinforcement learning with deep predictive models (Stadie et al., 2015b)

- Unifying count-based exploration and intrinsic motivation (Bellemare et al., 2016b)

- Surprise-based intrinsic motivation for deep reinforcement learning (Achiam and Sastry, 2017b)

- Deep Exploration via Bootstrapped DQN (Osband et al., 2016)

- Bootstrapped DQN modifies DQN to approximate a distribution over Q-values via the bootstrap. At the start of each episode, bootstrapped DQN samples a single Q-value function from its approximate posterior. The agent then follows the policy which is optimal for that sample for the duration of the episode. This is a natural adaptation of the Thompson sampling heuristic to RL that allows for temporally extended (or deep) exploration [21, 13].

- Curiosity-driven Exploration by Self-supervised Prediction (Pathak et al., 2017b)

2.2.2 In multi-task RL

- The Intentional Unintentional Agent: Learning to Solve Many Continuous Control Tasks Simultaneously https://arxiv.org/pdf/1707.03300.pdf

- Hybrid Reward Architecture forReinforcement Learning https://arxiv.org/pdf/1706.04208.pdf

Mult-tasking: Learning to Multi-Task by Active Sampling (Sharma et al., b)

- Similar setup as in section 3.1. They propose different measures.

- Exploration for Multi-task Reinforcement Learning with Learning with Deep Generative Models (Bangaru et al., 2016b)

- Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation (Kulkarni et al., 2016) – more a mechanism to generate goals with image masks

2.3 Measures (uncertainty, channel capacity)

- VIME: Variational Information Maximizing Exploration (Houthooft et al., 2016b)

- Value distribution (Bellemare et al., 2017)

- Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning (Gal and Ghahramani, 2016b)

- Empowerment (Salge et al., 2014)

2.4 Attention

- Deep Object-Centric Representations for Generalizable Robot Learning (Devin et al., 2017b) – Goker RG paper

- Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning (Zhu et al., 2017)

- Show, attend and tell: Neural image caption generation with visual attention (Xu et al., 2015)

2.5 Drive reduction

(Sanchez-Fibla et al., 2010a)

(Vouloutsi et al., 2013)

(Moulin-Frier et al., 2017)

2.6 Optimal reward

(Singh et al., 2010b)

2.7 From CL Meeting

From Goker:

RL2: Fast Reinforcement Learning via Slow Reinforcement Learning

https://arxiv.org/abs/1611.02779

From James:

Showing versus Doing: Teaching by Demonstration

https://papers.nips.cc/paper/6413-showing-versus-doing-teaching-by-demonstration.pdf

From Kaushik:

Exploration from Demonstration for Interactive Reinforcement Learning

https://www.cc.gatech.edu/~isbell/papers/efd-aamas-2016.pdf

From Peter:

Learning Exploration Strategies in Model-Based Reinforcement Learning

http://www.cs.utexas.edu/~pstone/Papers/bib2html/b2hd-AAMAS13-hester.html

3 Proposed approaches

This section proposes approaches to address the desiderata expressed in section 1. The approach in section 3.1 is being implemented in the current sprint and therefore provides more detail. Section 3.2 instead attempts at linking part of the literature above to the problem of designing a generic exploration policy, possibly optimizing intrinsic reward over a large set of (meta-)features.

Benchmarks are proposed for both.

3.1 Active task selection maximizing learning progress

This is being implemented in the current Sprint 31. The aim is to implement a procedure for prioritizing the practice of existing tasks, according to an empirical measure of learning progress for each of them. This will be used when the robot is in a state which is initial for multiple tasks, in order to decide which one to focus on. Such a situation occurs e.g. when the agent has several objects in its field of view, with different goal configurations existing for each of them. In that case, some of the tasks can be unnecessary to practice, either because the agent already performs well on them, or because they are too complicated to learn (e.g. because they require learning other skills first, or because the user has provided inaccurate GC examples).

The agent has therefore interest in practicing tasks where it expects maximal learning progress. To do so, we propose to keep track of the history of episodes for each existing task (see Fig. 1 below). Knowing the evolution of the final cumulative reward collected in each episode of a given task, one can empirically measure learning progress by estimating the derivative wrt time (e.g. through a linear regression). Note that we don't use the history of rewards within a single episode here, but instead the history of cumulative rewards across episodes. A probabilistic selection of the next task to be practiced can then be done according to their respective learning progress measures (e.g. with a softmax distribution). Fig. 2 and 3 show it on a simple example, as a proof of concept.

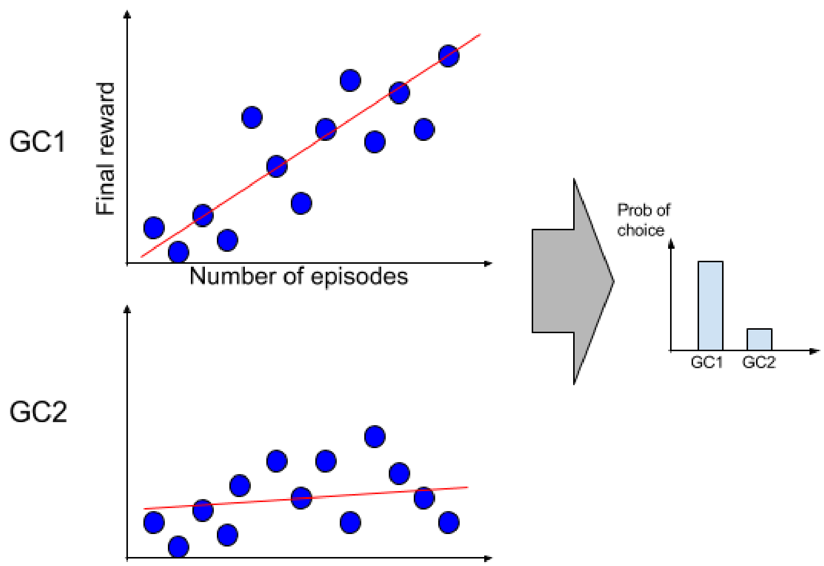

Figure 1: Illustrative example of empirically-measured learning progress: For each task (GC1 and GC2), we keep track of the cumulative reward (y-axis) returned at the end of each episode (x-axis) and fit a linear regression on the obtained data points providing an empirical measure of learning progress. The progress measures are then converted to probabilities of selecting each tasks, using e.g. a softmax.

3.1.1 Benchmark

The agent can practice four tasks, provided by the users through goal configurations and with overlapping initial states. *Task1* is already learned and always results in a high reward. *Task2* has to be learned but is relatively easy (e.g. orient to an object). *Task3* is more difficult, e.g. approach an object. *Task4* is very difficult, or even impossible (e.g. move on top of the table).The selection of tasks based on learning progress will first favor the selection of *Task2*, then of *Task3*, while avoiding the practice of *Task1* and *Task4* (unless the two others are perfectly learned). The evaluation will show that the agent will perform better on average on the four task compared to an agent which randomly select tasks.

Another interesting benchmark relates to sequential goal configurations. In that case, the user will train two goal configurations (e.g. *G1: "orient to blue ball"; "G2: approach blue ball"*) as well as a sequencial

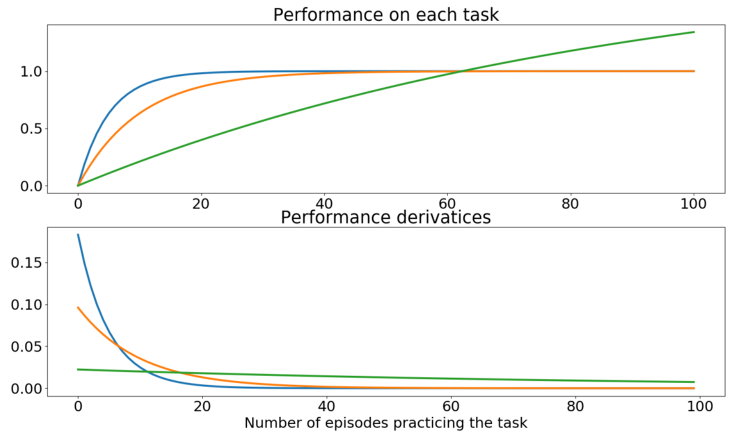

Figure 2: Idealized example. The agent can perform three different tasks, each displaying a predefined learning curve (top plot, where the x-axis is the number of episodes practicing one particular task). The blue curves corresponds to an easy task, the orange one to a more difficult one and the green one to an even more difficult one. The bottom plot shows the derivative of each learning curve, i.e. learning progress.

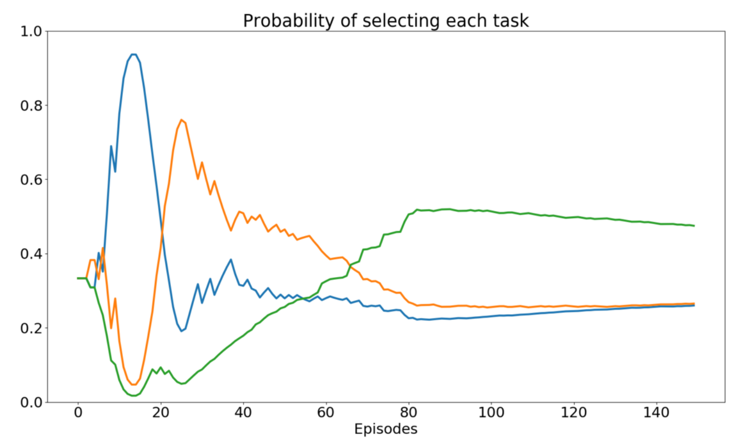

Figure 3: Probability of selecting each task when maximizing learning progress. At each time step, the agent selects a task and practices it, receiving the reward defined by the learning curves of Fig. 2. The agent keeps track of the returned reward for each task (as illustrated in Fig. 1). The resulting empirically-measured learning progress (through linear regression as in Fig. 1) is converted to a probability distribution over tasks using a softmax. The agent automatically generates a curriculum from easy (blue) to more difficult (orange, then green) tasks. This speeds up learning because the agent spents less time exploring tasks which are already learned (or which can't be learned yet, a case not shown here)

configuration *G3: "achieve G1, then G2"*. The prediction is that the task selection system described above will automatically generate the following curriculum: *G1 -> G2 -> G3*. The reason is that G1 is easier to achieve than G2 and that, in order to progress in G3, the agent must first master both G1 and G2 (or maybe at least G1). We will compare the time required to learned G3 using a random selection of tasks versus an active one as proposed above.

3.1.2 Perspectives

As noted in section 2.1, curriculum learning can be applied to initial states, to goal states, or both. The method we have just described generates a curriculum over a discrete set of existing tasks, while those previously mentioned (section 2.1) instead generate new initial or goal states (Florensa et al., 2017; Held et al., 2017; Sukhbaatar et al., 2017). We also have an existing system generating a curriculum over the initial states (described in this document). An interesting question is whether we can integrate these different aspects, generating a curriculum both on existing initial states and tasks, as well as generating new ones. Implementing generative models of initial and goal states complementing the existing norm classifiers is a possible direction.

3.2 Generic exploration policy

There is an extensive literature on exploration strategies improving learning speed on single-task RL (e.g. Rmax, E3, Thomson sampling...). However most of them are considering state spaces of reasonable dimensionality and are not applicable in our setup. Over the last few years, methods have been proposed to tackle the exploration problem in larger spaces in the context of value approximation based on deep neural nets. The main motivation for this is to allow learning in sparse-reward environments, where many action steps might be required to observe a single reward (e.g. Montezuma Revenge in Atari games). This constraint of sparse reward can appear as less relevant in our setup, where a norm classifier continuously provides a reward as a function of the distance to the goal. However, we still have to solve the problem of sparse reward when the robot is in a state where no initiation classifier fires ("no man's land" states). There are several ways of solving this problem.

The first solution is a navigation policy. This is the current solution for bringing the agent to initial states, through a prewired default behavior where the robot navigates from room to room. There are obviously ways to improve this default behavior, e.g. through built-in attention mechanisms attracting the agent towards salient visual features. Such salient features are likely to relate to existing goal configs or to provide useful data for training the sensory cortex. Existing methods modeling visual attention models using deep neural nets (Devin et al., 2017a) or target-driven visual navigation (Zhu et al., 2017) might be relevant here.

The second solution is a learned policy maximizing an intrinsic reward over a large set of features, including "meta-features" such as prediction error (Pathak et al., 2017a), surprise (Achiam and Sastry, 2017a), model-based exploration bonuses (Stadie et al., 2015a; Bangaru et al., 2016a), or count-based exploration in large state spaces (Bellemare et al., 2016a; Tang et al., 2017b; Yin and Pan, 2017; Fu et al., 2017). These methods have been mostly applied in the context of single-task RL, only a few being considering muti-task RL (Bangaru et al., 2016a; Sharma et al., a). A challenge is to figure out how to adapt single-task methods to our multi-task setup, e.g. for reaching initial states which can belong to any type of tasks (of course these methods could also be applied to explore while learning a single task, but this not (or less) the scope of this document). Recent methods measuring model uncertainty in (deep) RL are also relevant here (Gal and Ghahramani, 2016a; Bellemare et al., 2017), as are other information-theoretic measures (Houthooft et al., 2016a; Salge et al., 2014). Another interesting question relates to the possibility of combining multiple intrinsic rewards (as those mentioned above) in a way which would maximize extrinsic reward on average over the set of existing tasks, in particular in the context of off-policy learning. This links to the concept of optimal reward (Singh et al., 2010a), i.e. a reward function that maximizes the expected fitness over the distribution of environments ("environments" being "tasks" in our case, although optimizing over environments is also interesting and could leverage the Continua platform).

Methods generating a curriculum though initial or goal state generation presented in section 2.1 are also of interest and they seem relatively easy to implement (in particular the one using self play (Sukhbaatar et al., 2017)).

3.2.1 Benchmark

We already have as a baseline the current default behavior navigating from room to room. We will evaluate the new policy, being based either on a navigation policy or on intrinsic rewards (I thing the second option has more interest for us and that we should focus on it) on its performance to learn a set of user-defined tasks. We will train a set of goal configurations and evaluate the learning performance on all the provided tasks. We will compare the performance of the new exploration policy to the existing default behavior. In an on-policy mode, this performance will mostly depend on the ability of the exploration policy to reach relevant initial states. In an off-policy mode, it will also depend on the ability of the exploration policy to generate transitions which are informative for optimizing several policies in parallel.

3.3 Self-regulation / drive reduction

This is less important than the two previous approaches, but let's still mention it. In the biological world, achieving new goals is not the primary source of motivation for an organism (Hull, 1943; Maslow, 1943; Sterling, 2012). Instead, the first motivation is to self-regulate internal needs through drive reduction mechanisms (e.g. reducing hunger through foraging). This mechanisms are in part innate and bootstrap learning by generating actions, perceptions and rewards. In our robotics setup, this needs are less obvious, although a few will still be present in physical robots: e.g. maintaining battery level, or avoiding collisions and overheating. In the context of the autonomous behavior we are considering in this document, implementing self-regulation mechanisms could at least bootstrap behavior and learning out of the box, typically when no prior knowledge exists and no task has been provided by the user yet. Example of such primary drives could be obstacle avoidance, reducing energy consumption while moving, or finding the charging station. A few models of drive reduction for robotics, dealing with possibly conflicting drives, have been proposed (e.g. (Sanchez-Fibla et al., 2010b; Vouloutsi et al., 2013; Moulin-Frier et al., 2017)).

4 Workplan

Below is a proposition of sprint goals, starting from the current sprint. They refer to the sections above providing details or ideas on how to achieve them, as well as benchmarks. A more detailed diagram is available here: here: `https://docs.google.com/drawings/d/1he9U66Ok_5R462eHLy3WUwd9RsUuGsB34KWFL94huZU/edit`

- **Sprint 31** (current sprint): implement an active selection of tasks based on an empirical measure of learning progress (section 3.1) and complete this document
- **Sprint 32**: Evaluate the active task selection of Sprint 31 on Robutler (section 3.1.1) and write a proposal on the generic exploration policy (section 3.2)
- **Sprint 33**: Implement the generic exploration policy based on the proposal of Sprint 32
- **Sprint 34**: Evaluate the generic exploration policy (section 3.2.1) and plan the next steps

Other possibilities:

- Implement and evaluate a navigation policy (section 3.2, first bullet point).

- Implement and evaluate a drive reduction system (section 3.3)

References

Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017a.

Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017b.

Sai Praveen Bangaru, JS Suhas, and Balaraman Ravindran. Exploration for Multi-task Reinforcement Learning with Deep Generative Models. *arXiv preprint arXiv:1611.09894*, 2016a.

Sai Praveen Bangaru, JS Suhas, and Balaraman Ravindran. Exploration for Multi-task Reinforcement Learning with Deep Generative Models. *arXiv preprint arXiv:1611.09894*, 2016b.

Adrien Baranes and Pierre-Yves Oudeyer. Active Learning of Inverse Models with Intrinsically Motivated Goal Exploration in Robots. *Robotics and Autonomous Systems*, 61(1):49–73, 2013. doi: 10.1016/j.robot. 2012.05.008. URL http://arxiv.org/abs/1301.4862v1;http://arxiv.org/pdf/1301.4862v1.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016a.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016b.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A Distributional Perspective on Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning*, jul 2017. URL http://arxiv.org/abs/1707.06887.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

Coline Devin, Pieter Abbeel, Trevor Darrell, and Sergey Levine. Deep Object-Centric Representations for Generalizable Robot Learning. *arXiv preprint arXiv:1708.04225*, 2017a.

Coline Devin, Pieter Abbeel, Trevor Darrell, and Sergey Levine. Deep Object-Centric Representations for Generalizable Robot Learning. *arXiv preprint arXiv:1708.04225*, 2017b.

Carlos Florensa, David Held, Markus Wulfmeier, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. *arXiv preprint arXiv:1707.05300*, 2017.

Justin Fu, John Co-Reyes, and Sergey Levine. Ex2: Exploration with exemplar models for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2574–2584, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Maria Florina Balcan and Kilian Q Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 2016a. PMLR. URL http://proceedings.mlr.press/v48/gal16.html.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Maria Florina Balcan and Kilian Q Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 2016b. PMLR. URL http://proceedings.mlr.press/v48/gal16.html.

Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. *arXiv preprint arXiv:1704.03003*, 2017.

David Held, Xinyang Geng, Carlos Florensa, and Pieter Abbeel. Automatic Goal Generation for Reinforcement Learning Agents. may 2017. URL http://arxiv.org/abs/1705.06366.

Rein Houthooft, Xi Chen, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. VIME: Variational Information Maximizing Exploration. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1109–1117. Curran Associates, Inc., 2016a. URL http://papers.nips.cc/paper/6591-vime-variational-information-maximizing-exploration.pdf.

Rein Houthooft, Xi Chen, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. VIME: Variational Information Maximizing Exploration. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1109–1117. Curran Associates, Inc., 2016b. URL http://papers.nips.cc/paper/6591-vime-variational-information-maximizing-exploration.pdf.

Clark Hull. Principles of behavior. 1943.

Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems*, pages 3675–3683, 2016.

Abraham H Maslow. A theory of human motivation. *Psychological review*, 50(4):370, 1943.

Clément Moulin-Frier, T Fischer, M Petit, G Pointeau, J.-Y. Puigbo, U Pattacini, S C Low, D Camilleri, P Nguyen, M Hoffmann, H J Chang, M Zambelli, A.-L. Mealier, A Damianou, G Metta, T Prescott, Y Demiris, P.-F. Dominey, and P Verschure. DAC-h3: A Proactive Robot Cognitive Architecture to Acquire and Express Knowledge About the World and the Self. *IEEE Transactions on Cognitive and Developmental Systems*, 2017. URL https://arxiv.org/abs/1706.03661.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. In *Advances in neural information processing systems*, pages 4026–4034, 2016.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, volume 2017, 2017a.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, volume 2017, 2017b.

Christoph Salge, Cornelius Glackin, and Daniel Polani. Empowerment–An Introduction. pages 67–114. Springer, Berlin, Heidelberg, 2014. doi: 10.1007/978-3-642-53734-9_4. URL http://link.springer.com/10.1007/978-3-642-53734-9{_}4.

Marti Sanchez-Fibla, Ulysses Bernardet, Erez Wasserman, Tatiana Pelc, Matti Mintz, Jadin C Jackson, Carien Lansink, Cyriel Pennartz, and Paul F M J Verschure. Allostatic control for robot behavior regulation: a comparative rodent-robot study. *Advances in Complex Systems*, 13(3):377–403, 2010a.

Marti Sanchez-Fibla, Ulysses Bernardet, Erez Wasserman, Tatiana Pelc, Matti Mintz, Jadin C Jackson, Carien Lansink, Cyriel Pennartz, and Paul F M J Verschure. Allostatic control for robot behavior regulation: a comparative rodent-robot study. *Advances in Complex Systems*, 13(3):377–403, 2010b.

Sahil Sharma, Ashutosh Kumar Jha, Parikshit S Hegde, and Balaraman Ravindran. Learning to Multi-Task by Active Sampling. a.

Sahil Sharma, Ashutosh Kumar Jha, Parikshit S Hegde, and Balaraman Ravindran. Learning to Multi-Task by Active Sampling. b.

Satinder Singh, Richard L Lewis, Andrew G Barto, and Jonathan Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82, 2010a.

Satinder Singh, Richard L Lewis, Andrew G Barto, and Jonathan Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82, 2010b.

Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015a.

Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015b.

Peter Sterling. Allostasis: A model of predictive regulation. *Physiology and Behavior*, 106(1):5–15, 2012. ISSN 00319384. doi: 10.1016/j.physbeh.2011.06.004.

Sainbayar Sukhbaatar, Ilya Kostrikov, Arthur Szlam, and Rob Fergus. Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv preprint arXiv:1703.05407*, 2017.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. #Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2750–2759. Curran Associates, Inc., 2017a. URL http://papers.nips.cc/paper/6868-exploration-a-study-of-count-based-exploration-for-deep-reinforcement-learning.pdf.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. #Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2750–2759. Curran Associates, Inc., 2017b. URL http://papers.nips.cc/paper/6868-exploration-a-study-of-count-based-exploration-for-deep-reinforcement-learning.pdf.

V Vouloutsi, S Lallée, and P Verschure. Modulating behaviors using allostatic control. In Nathan F Lepora, Anna Mura, Holger G Krapp, Paul F M J Verschure, and Tony J Prescott, editors, *Biomimetic and Biohybrid Systems*, volume 287–298 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-39801-8. doi: 10.1007/978-3-642-39802-5. URL http://link.springer.com/10.1007/978-3-642-39802-5.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

Haiyan Yin and Sinno Jialin Pan. Hashing Over Predicted Future Frames for Informed Exploration of Deep Reinforcement Learning. *arXiv preprint arXiv:1707.00524*, 2017.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3357–3364. IEEE, may 2017. ISBN 978-1-5090-4633-1. doi: 10.1109/ICRA.2017.7989381. URL http://ieeexplore.ieee.org/document/7989381/.