DYNAMICS-AWARE UNSUPERVISED SKILL DISCOVERY

Archit Sharma, Shane Gu, Sergey Levine, Vikash Kumar, Karol Hausman
Google Brain
{architsh,shanegu,karolhausman,slevine,vikashplus}@google.com

ABSTRACT

Model-based Reinforcement Learning (MBRL) shows the potential to facilitate learning of many different tasks by taking advantage of the learned model of the world. Learning a global model that works in every part of state-space, however, can be exceedingly challenging, especially in complex, dynamical environments. To overcome this problem, we present a method that is able to discover skills together with their 'skill dynamics models' in an unsupervised fashion. By finding the skills whose dynamics are easier to learn, we are able to effectively partition the space into local models and their corresponding policies. In addition, we show how a simple MBRL algorithm can leverage the learned skills to solve a set of downstream task without any additional learning. Our results indicate that our zero-shot online planning method that uses skill dynamics can significantly outperform strong baselines that were trained specifically on downstream tasks.

1 INTRODUCTION AND RELATED WORK

Deep reinforcement learning enables autonomous learning of diverse and complex tasks with rich sensory inputs, temporally extended goals, and challenging dynamics, such as discrete game-playing domains (Mnih et al., 2013; Silver et al., 2016), and continuous control domains including locomotion (Schulman et al., 2015; Heess et al., 2017) and manipulation (Rajeswaran et al., 2017; Kalashnikov et al., 2018; Gu et al., 2017). Classically, reinforcement learning is concerned with learning one task at a time. In contrast, model-based reinforcement learning methods (Li & Todorov, 2004; Deisenroth & Rasmussen, 2011; Watter et al., 2015) can in principle acquire dynamics models that can then be utilized to perform new tasks at test time. Indeed, this capability has been demonstrated in several recent works (Levine et al., 2016; Nagabandi et al., 2018; Chua et al., 2018; Kurutach et al., 2018; Ha & Schmidhuber, 2018). But this kind of generality comes at a price: model-based RL methods must acquire an accurate *global* model of the system, which can be exceedingly challenging when the dynamics are complex and discontinuous, or when the observation space is high-dimensional. Can we retain the flexibility of model-based RL, while still making use of model-free RL to acquire proficient low-level behaviors under complex dynamics?

In this paper, we propose an unsupervised hierarchical RL framework for learning low-level skills using model-free RL with the explicit aim of making model-based control easy. Our skills directly optimize for *predictability*, providing a substantially better representation on top of which predictive models can be learned. Crucially, the skills do not require any supervision to learn, and are acquired entirely through autonomous exploration. This means that the repertoire of skills and their predictive model can be learned before the agent has been tasked with any goal or reward function. When a reward is provided at test-time, the agent can utilize its previously learned skills and model to immediately perform the task without further training.

Central to our method is the concept of skill discovery via mutual information maximization. This principle, proposed in prior work that utilized purely model-free unsupervised RL methods (Daniel et al., 2012; Florensa et al., 2017; Eysenbach et al., 2018; Gregor et al., 2016), aims to learn diverse skills via a discriminability objective: a good set of skills is one where it is easy to distinguish the skills from each other, which means they perform distinct tasks and cover the space of possible behaviors. Building on this prior work, we distinguish our skills based on their corresponding *predictive models* – that is, skills are good if they can be distinguished based on how they modify the

original uncontrolled dynamics of the system. This simultaneously encourages the skills to be both *diverse* and *predictable*. Such skills can also provide extended actions and temporal abstraction, which enable more efficient exploration for the agent to solve various tasks, as shown in other hierarchical RL approaches (Sutton et al., 1999; Bacon et al., 2017; Vezhnevets et al., 2017; Nachum et al., 2018; Hausman et al., 2018). Crucially, our method allows learning action and temporal abstraction, skill dynamics, and intrinsic motivation, all through a single optimization objective. We demonstrate the effectiveness of our approach by performing zero-shot goal navigation for ant using primitives learnt by our proposed objectives.

2 Unsupervised Learning of Dynamics-aware Skills

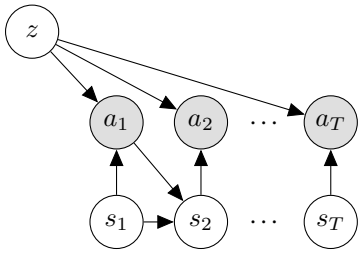

Figure 1: Graphical model for the world P in which the trajectories are generated while interacting with the environment. Shaded nodes represent the distributions we optimize.

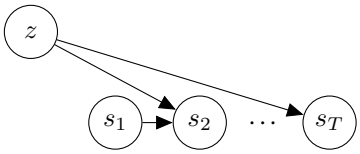

Figure 2: Graphical model for the world Q which is the desired representation of the world.

We assume the conventional Markov Decision Process (MDP) setting for RL. While we focus on the continuous control domain, this discussion does not preclude discrete control domains. At each time step t, the environment $\mathcal{E}$ emits a state $s_t \in \mathbb{R}^{d_s}$. The agent takes an action $a_t \in \mathbb{R}^{d_a}$ that is sampled from the control policy parameterized as $\pi(a_t|s_t, z)$. Here, z represents the latent space for skills or primitives, which can be discrete or continuous. We assume a continuous $z \in \mathbb{R}^{d_z}$ for the rest of this section. The underlying dynamics of the environment are represented by $p(s_{t+1}|s_t, a_t)$, starting with an initial state distribution of $p(s_1)$.

Our method consists of two phases – *Unsupervised Skill Discovery*, and *Planning*. During *Unsupervised Skill Discovery* we optimize the control policy to discover diverse behaviors in the environment without any extrinsic rewards. More importantly, the objective is constructed to simultaneously learn a transition function $q_\phi(s' \mid s, z)$, coined as skill dynamics. For downstream *Planning* on a given task, we leverage the skill-dynamics to compose the learnt primitives. As long as the underlying dynamics of the environment are the same, the downstream task can have any reward function.

2.1 Unsupervised Skill Discovery

We take the probabilistic route to obtain our unsupervised RL objective. We setup a graphical model P as shown in Figure 1, which represents the distribution of trajectories generated by a given policy π. The joint distribution is given by:

$$p(s_1, a_1 \dots a_{T-1}, s_T, z) = p(z)p(s_1) \prod_{t=1}^{T-1} \pi(a_t|s_t, z)p(s_{t+1}|s_t, a_t).$$

We setup another graphical model Q, which represents the desired model of the world. In particular, we are interested in approximating $p(s'|s, z)$, which represents the transition function for a particular primitive. The joint distribution for Q shown in Figure 2 is given by:

$$q(s_1, \dots s_T, z) = p(z)p(s_1) \prod_{t=1}^{T-1} q(s_{t+1}|s_t, z).$$

The goal of our approach is to optimize the distribution $\pi(a|s, z)$ in the graphical model P to minimize the information lost when transforming to the representation of the graphical model Q. In

particular, we are interested in minimizing the KL divergence between p and q - $\mathcal{D}_{KL}(p||q)$. However, since q is not known apriori, we setup the objective as $\min_{q \in Q} \mathcal{D}_{KL}(p||q)$, which is the reverse information projection (Csiszár & Matus (2003)). Interestingly, it was shown in Friedman et al. (2001) that:

$$\min_q \mathcal{D}_{KL}(p||q) = \mathcal{I}_P - \mathcal{I}_Q,$$

where $\mathcal{I}_P$ and $\mathcal{I}_Q$ represents the multi-information for distribution P on the respective graphical models. The multi-information (Slonim et al. (2005)) for a graphical model G with nodes g_i is defined as:

$$\mathcal{I}_G = \sum_i I(g_i; Pa(g_i)),$$

where $Pa(g_i)$ denotes the nodes upon which g_i has conditional dependence in G. Using this definition, we can compute the multi-information terms:

$$\mathcal{I}_P = \sum_{t=1}^{T} I(a_t; \{s_t, z\}) + \sum_{t=2}^{T} I(s_t; \{s_{t-1}, a_{t-1}\}) \quad \text{and} \quad \mathcal{I}_Q = \sum_{t=2}^{T} I(s_t; \{s_{t-1}, z\}).$$

Here, $I(s_t; \{s_{t-1}, a_{t-1}\})$ is constant as we assume the underlying dynamics to be fixed (and unknown), and we can safely ignore this term. The final objective to be maximized is given by:

$$\begin{aligned} R(\pi) &= \sum_{t=1}^{T-1} I(s_{t+1}; \{s_t, z\}) - I(a_t; \{s_t, z\}) \\ &= \sum_{t=1}^{T-1} \mathbb{E}_\pi \Big[\log \frac{p(s_{t+1}|s_t, z)}{p(s_{t+1})} - \log \frac{\pi(a_t|s_t, z)}{\pi(a_t)} \Big]. \end{aligned}$$

We can leverage the fact that $\int \pi(a) \log \frac{\pi(a)}{p(a)} da \geq 0$ for some uniform prior $p(a)$ over the bounded action space. Similarly, $\int p(s'|s, z) \log \frac{p(s'|s,z)}{q_\phi(s'|s,z)} ds' \geq 0$ for any variational approximation $q_\phi(s'|s, z)$. Ignoring the constant $p(a)$, the objective can now be written as

$$R(\pi) \geq \sum_{t=1}^{T-1} \mathbb{E}_\pi [\log q_\phi(s_{t+1}|s_t, z) - \log p(s_{t+1}) - \log \pi(a_t|s_t, z)] = R(\pi, q_\phi).$$

This results in an unsupervised skill learning objective that explicitly fits a model for transition behaviors, while providing a grounded connection with probabilistic graphical models. Note, unlike the setup of "control as inference" (Levine (2018), Ziebart et al. (2008)) which casts policy learning as variational inference, the policy here is assumed to be part of the generative model itself (and thus the resulting difference in the direction of $\mathcal{D}_{KL}$). The procedure carried out above has been explicated in Alemi & Fischer (2018) in the context of supervised learning, having its roots in the work on information bottleneck (Tishby & Zaslavsky (2015), Alemi et al. (2016)). The information bottleneck interpretation can be also applied here - our algorithm maximizes the information between next state and the augmented current state (i.e. (s_t, z)), while constraining the information between actions and the augmented state.

Maximizing $R(\pi, q_\phi)$ automatically suggests an alternating optimization scheme, as shown in Algorithm 1. Notice, $\nabla_\phi R(\pi, q_\phi) = \sum_{t=1}^{T-1} \mathbb{E}_\pi [\nabla_\phi \log q(s_{t+1}|s_t, z))]$, which is simply maximum likelihood over the transitions generated by the current policy. The optimization of the policy π can be interpreted as entropy-regularized RL with a reward function $\log q_\phi(s_{t+1}|s_t, z) - \log p(s_{t+1})$. Unfortunately, $\log p(s_{t+1})$ is intractable to compute so we have to resort to approximations. We choose to re-use the skill-dynamics model to approximate $p(s_{t+1}) = \int p(s_{t+1}|s, z) p(s, z) dz ds \approx \frac{1}{L} \sum_{i=1}^{L} p(s_{t+1}|s_t, z_i)$, where z_i are sampled from the prior $p(z)$ and s is fixed to s_t. The final reward function can be written as:

$$r_z(s, a, s') = \log \frac{q_\phi(s_{t+1}|s_t, z)}{\sum_{i=1}^{L} q_\phi(s_{t+1}|s_t, z_i)} + \log L, z_i \sim p(z).$$

While there exist other potential sampling schemes to approximate $\log p(s_{t+1})$, we found that this formulation encourages the diversity of primitives better than others. A more clear motivation for this approximation can be understood from the perspective of empowerment (Mohamed & Rezende (2015)), which is discussed in Appendix C.

2.2 PLANNING USING SKILL DYNAMICS

Optimizing the reward function given above yields a latent-conditioned policy $\pi(a|s, z)$, which generates diverse behaviors as discernible under the skill-dynamics model $q_\phi(s'|s, z)$. The reward function incentivizes the agent to produce consistent behavior for a given z, while producing diverse state transitions for different z. The biggest benefit of this setup is that we can use planning algorithms for downstream tasks, which can be extremely sample-efficient when compared to model-free RL.

We provide a simple greedy algorithm (Algorithm 2) which exploits the skill dynamics to temporally compose primitives for a given downstream task. For a diverse set of discrete primitives, the latent space planning can be fairly straightforward and can even be done online without any learning.

3 EXPERIMENTS

We discuss experimental observations in this section. We begin with a qualitative observations about the unsupervised skill learning, and then we show quantitative results on downstream planning tasks. For all our experiments with the Ant environment (Todorov et al., 2012), we assume the observation space for the skill-dynamics to be x-y plane, also utilized in (Eysenbach et al., 2018), something which helps guide the diversity of unsupervised skills in the right subspace (Appendix D). This assumption helps us focus on the more important contribution of planning using skill dynamics.

3.1 DISCRETE V/S CONTINUOUS LATENT SPACE

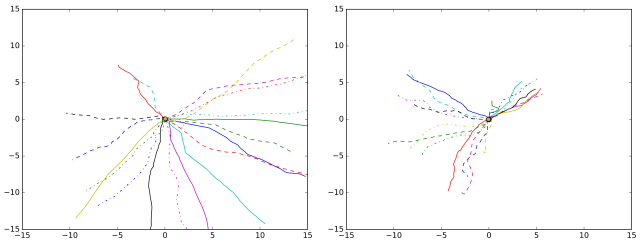

Figure 3: Ant trajectory samples from two-dimensional latent space (*left*) and discrete latent space (*right*)

Unlike previous works, we found our setup to be capable of embedding skills in a continuous latent space $p(z)$ which we assume to be a continuous uniform distribution. It is clear from Figure 3 that more diversity can be embedded into a smaller continuous space. We also found the continuous latent space to be semantically meaningful, as is depicted by the interpolation between the samples from the primitive space shown in Appendix E. The diversity in continuous space is controlled by the number of samples from the prior $p(z)$ when approximating $p(s_{t+1})$. However, a discrete primitive space has benefits on the downstream task as planning in the discrete space can be simpler.

3.2 MODEL-BASED REINFORCEMENT LEARNING

The primary motivation to learn the parametric model $q_\phi(s'|s, z)$ was to be able to use planning algorithms for downstream tasks, which can be extremely sample-efficient. While we lose some diversity of primitives by constraining ourselves ourselves to a small set of discrete primitives, we can now demonstrate something significantly stronger in regards to sample-efficiency: Zero-shot online planning. For the downstream task, we choose the task of ant-navigation. We use the Ant model from the OpenAI gym (Brockman et al., 2016), where we provide a dense reward for navigating to a specific goal.

We choose to compare with the more conventional model-based RL, which learns a global model $p(s'|s, a)$. Model-based methods are known to be sample-efficient as compared to model-free RL, and exhibit some generalization. We compare with a few variants of the model-based approach:

- Random-MBRL: We train the model $p(s'|s,a)$ on randomly collected trajectories, and we test the zero-shot generalization of the model on the test distribution of goals. This variant of model-based RL matches the assumptions made by our approach.
- Weak-oracle MBRL: We train the model $p(s'|s,a)$ on trajectories generated for navigating to randomly sampled goals, the same distribution of goals from which the test set is sampled.
- Strong-oracle MBRL: We train the model $p(s'|s,a)$ for every goal in the test set.

Note that the Weak-oracle MBRL and Strong-oracle MBRL, benefit from goal directed exploration provided by the oracle, a significant advantage over our model-based method which only benefits from curiosity based exploration (in the right sub-space of x-y plane). We report the performance of the converged models on the test set of goals. We report the d as the metric, which represents the distance to the goal g averaged over the episode (with a fixed horizon for all models), normalized by the initial distance to the goal g. Therefore $0 \leq d \leq 1$ (assuming the agent goes closer to the goal), and lower the d, the better the performance. For every g, we average d over 10 trials.

	(10, 10)	(10, -10)	(-10, -10)	(-10, 10)	(0, -10)	(5, 10)
Random-MBRL	0.78	0.92	0.83	0.88	0.74	0.72
Weak-oracle MBRL	0.77	0.65	0.66	0.72	0.54	0.64
Strong-oracle MBRL	0.65	0.62	0.62	0.64	0.5	0.53
Skill-Dynamics (Discrete Primitives)	0.56	0.82	0.42	0.54	**0.38**	0.56
Skill-Dynamics (Continuous Primitives)	**0.35**	**0.35**	**0.38**	**0.44**	0.40	**0.33**

From the table, it is clear that the zero-shot planning outperforms/is comparable to the model-based RL baselines, despite the significant advantage to two of the baselines. There is no primitive that goes into the bottom right corner, as is clear from the Figure 3 (right), which is why it struggles to reach the goal (10, -10). A comparison with Random-MBRL shows the significance of mutual information based exploration, especially with the right parameterization.

3.3 RL Controller v/s Planning

There are several approaches to learning a diverse set of low-level primitives (Eysenbach et al. (2018); Gregor et al. (2016); Achiam et al. (2018)). However, none of the approaches can leverage a planning based algorithms to solve the downstream tasks, and are constrained to use a task-specific meta-controller which is again trained by RL. To demonstrate the sample-efficiency of planning, we run a controlled experiment where we provide the same frozen lower-level primitives to both (as the performance is heavily contingent on the quality of lower-level primitives). The skill-dynamics based planning approach outperforms the model-free RL consistently. However, we note that the nature of this problem is very well suited for the greedy planner, and in general it might be hard to perform superior to model-free RL. But, our main emphasis is on sample efficiency of planning based approaches while achieving good performance on a task, rather than striving for optimality.

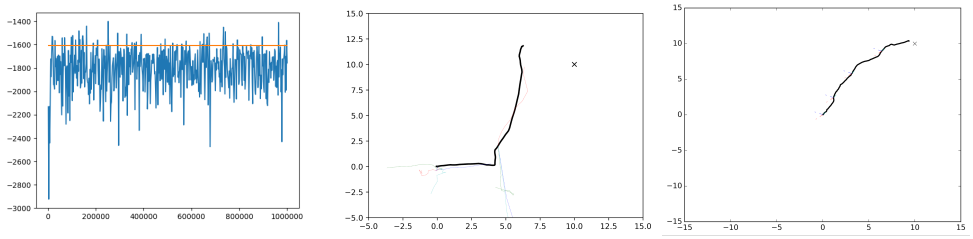

Figure 4: (*left*) Comparison of performance of Planning (*Orange*) and RL trained meta-controller (*Blue*). The RL controller takes about $150,000$ samples to get an average performance close to the zero-shot planning performance on ant-navigation to (10, 10). Planning using (*middle*) skills learnt with explicit rewards and (*right*) skills learnt with predictable dynamics.

3.4 Fit Skill Dynamics to Explicitly learnt policies, and then plan

The previous set of experiments showcase the utility of $q(s'|s, z)$ as a model for planning. But, the proposed unsupervised objective is not the only way to learn $q(s'|s, z)$. In principle, one could learn the skills z using explicit reward functions for every skills or other unsupervised objectives and then fit $q(s'|s, z)$. Such an algorithm would first have a computational overhead. The skills learnt through the proposed objective are more predictable, and and more easily plannable. To demonstrate this, we train the ant skills for locomotion using explicit rewards and then fit the skill-dynamics model to the demonstrations generated by the expert skills. From Figure 4, it is clear that skill-dynamics can still be used for planning when fit to skills learnt via explicit rewards, but more effective when trained to be predictable.

4 Conclusion and Future Work

Some obvious follow ups: (1) Extend planning algorithms to continuous primitives (which resembles the more conventional model-based continuous control) and (2) Extend to hierarchical control with different observation spaces (example: Ant in a Maze with locomotion primitives). A more challenging and possibly fruitful direction of research would be to understand the role of state-space representation in unsupervised skill-learning, and possibly develop an algorithm to learn these representations. Overall, we present a novel unsupervised learning objective which can effectively plan on downstream tasks, combining the benefits of model-free and model-based RL.

References

Joshua Achiam, Harrison Edwards, Dario Amodei, and Pieter Abbeel. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

David Barber Felix Agakov. The im algorithm: a variational approach to information maximization. *Advances in Neural Information Processing Systems*, 16:201, 2004.

Alexander A Alemi and Ian Fischer. Therml: Thermodynamics of machine learning. *arXiv preprint arXiv:1807.04162*, 2018.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *CoRR*, abs/1606.01540, 2016. URL `http://arxiv.org/abs/1606.01540`.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pp. 4759–4770, 2018.

Imre Csiszár and Frantisek Matus. Information projections revisited. *IEEE Transactions on Information Theory*, 49(6):1474–1490, 2003.

Christian Daniel, Gerhard Neumann, and Jan Peters. Hierarchical relative entropy policy search. In *Artificial Intelligence and Statistics*, pp. 273–281, 2012.

Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465–472, 2011.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Carlos Florensa, Yan Duan, and Pieter Abbeel. Stochastic neural networks for hierarchical reinforcement learning. *arXiv preprint arXiv:1704.03012*, 2017.

Nir Friedman, Ori Mosenzon, Noam Slonim, and Naftali Tishby. Multivariate information bottleneck. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pp. 152–161. Morgan Kaufmann Publishers Inc., 2001.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3389–3396. IEEE, 2017.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, pp. 2455–2467, 2018.

Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rk07ZXZRb`.

Nicolas Heess, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, SM Eslami, Martin Riedmiller, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. *arXiv preprint arXiv:1802.10592*, 2018.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Weiwei Li and Emanuel Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *ICINCO (1)*, pp. 222–229, 2004.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pp. 2125–2133, 2015.

Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 3307–3317, 2018.

Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7559–7566. IEEE, 2018.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I Jordan, and Philipp Moritz. Trust region policy optimization. In *Icml*, volume 37, pp. 1889–1897, 2015.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Noam Slonim, Gurinder S Atwal, Gasper Tkacik, and William Bialek. Estimating mutual information and multi–information in large networks. *arXiv preprint cs/0502017*, 2005.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5. IEEE, 2015.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3540–3549. JMLR. org, 2017.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, pp. 2746–2754, 2015.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

A ALGORITHMS

Initialize π, q_ϕ;
while *not converged* **do**
  Collect new M on-policy samples;
  Update q_ϕ using K_1 steps of mini batch gradient descent on M transitions;
  Compute $r_z(s, a, s')$ for M transitions;
  Update π using K_2 steps of SAC on M transitions;
end

Algorithm 1: Unsupervised Skill Discovery

$s \leftarrow s_0$;
for $t \leftarrow 1$ **to** H_E **do**
  $r_{max} \leftarrow -\infty, z_{max} \leftarrow 1$;
  for $z \leftarrow 1$ **to** $|Z|$ **do**
    $r_{cur} \leftarrow 0, s_{run} \leftarrow s_{cur}$;
    for $k \leftarrow 1$ **to** H_P **do**
      Sample a from $\pi(a|s_{run}, z)$;
      Sample s' from $q_\phi(\cdot|s_{run}, z)$;
      $r_{cur} += r_{env}(s', a, s_{run})$;
      $s_{run} \leftarrow s'$;
    end
    if $r_{cur} > r_{max}$ **then**
      $r_{max} \leftarrow r_{cur}, z_{max} \leftarrow z$;
    end
  end
  Sample a from $\pi(\cdot|s_{cur}, z_{max})$;
  Sample s_{cur} from environment;
end

Algorithm 2: Greedy Planning for Discrete Primitives

B DIVERSITY IS ALL YOU NEED

We can carry out the exercise for the reward function in Diversity is All You Need (DIAYN) (Eysenbach et al. (2018)) to provide a graphical model interpretation of the objective used in the paper. To conform with objective in the paper, we assume to be sampling to be state-action pairs from skill-conditioned stationary distributions in the world P, rather than trajectories. Again, the objective

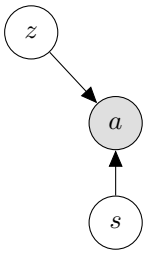

Figure 5: Graphical model for the world P representing the stationary state, action distribution. Shaded nodes represent the distributions we optimize.

Figure 6: Graphical model for the world Q using which we is the representation we are interested in. Shaded nodes represent the distributions we optimize.

to be maximized is given by

$$\begin{aligned} R(\pi) &= -\mathcal{I}_P + \mathcal{I}_Q \\ &= -I(a; \{s, z\}) + I(z; s) \\ &= \mathbb{E}_\pi[\log \frac{p(z|s)}{p(z)} - \log \frac{\pi(a|s,z)}{\pi(a)}] \\ &\geq \mathbb{E}_\pi[\log q_\phi(z|s) - \log p(z) - \log \pi(a|s,z)] = R(\pi, q_\phi) \end{aligned}$$

where we have used the variational inequalities to replace $p(z|s)$ with $q_\phi(z|s)$ and $\pi(a)$ with a uniform prior over bounded actions $p(a)$ (which is ignored as a constant).

C Interpretation as Empowerment in the Latent Space

Recall, the empowerment objective (Mohamed & Rezende (2015)) can be stated as

$$I(s'; a|s) = \mathcal{H}(a|s) - \mathbb{E}_{s'}[\mathcal{H}(a|s', s)] \geq \mathcal{H}(a|s) + \mathbb{E}_{p(s'|s,a)\pi(a|s)}[\log q_\phi(a|s', s))]$$

where the we are are learning a flat policy $\pi(a|s)$, and using the variational approximation $q(a|s', s)$ for the true action-posterior $p(a|s', s)$. We can connect our objective with empowerment if we assume a latent-conditioned policy $\pi(a|s, z)$ and optimize $I(s'; z|s)$, which can be interpreted as empowerment in the latent space z. There are two ways to decompose this objective:

$$\begin{aligned} I(s'; z|s) &= \mathcal{H}(z|s) - \mathbb{E}_{s'}[\mathcal{H}(z|s, s')] \\ &= \mathcal{H}(s'|s) - \mathbb{E}_{p(z|s)}[\mathcal{H}(s'|s, z)] \end{aligned}$$

Using the first decomposition, we can construct a an objective using a variational lower bound which learns the network $q_\phi(z|s', s)$. This is a DIAYN like inference network, which learns to discriminate skills based on the transitions they generate in the environment and not the state-distribution induced by each skill. However, we are interested in learning the network $q_\phi(s'|s, z)$, which is why we work with the second decomposition. But, again we are stuck with marginal transition entropy, which is intractable to compute. We can handle it in a couple of ways:

$$\begin{aligned} I(s'; z|s) &\geq \mathbb{E}_z[\mathbb{E}_{p(s'|s,z)}[\log \frac{q_\phi(s'|s, z)}{p(s'|s)}]] \\ &\approx \mathbb{E}_z[\mathbb{E}_{p(s'|s,z)}[\log \frac{q_\phi(s'|s, z)}{\sum_{i=1}^{L} q_\phi(s'|s, z_i)} + \log L]] \end{aligned}$$

where $p(s'|s)$ represents the distribution of transitions from the state s. Note, we are using the approximation $p(s'|s) = \int p(s'|s, z)p(z)dz \approx \frac{1}{L}\sum_{i=1}^{L} q_\phi(s'|s, z_i)$. Our use of $q(s'|s, z)$ encodes the intuition that the q should represent the distribution of transitions from s under different primitives, and thus the marginal of q should approximately represent $p(s'|s)$. We re-exploit this approximation in our objective as well, noting that this strongly encourages the skills to produce diverse transitions. However, this procedure does not yield entropy-regularized RL by itself, but arguments similar to

those provided for Information Maximization algorithm by Mohamed & Rezende (2015) can be made here to justify it in this empowerment perspective.

Note, this procedure makes an assumption $p(z|s) = p(z)$ when approximating $p(s'|s)$. While every skill is expected to induce a different state-distribution in principle, this is not a bad assumption to make as we often times expect skills to be almost state-independent (consider locomotion primitives, which can essentially be activated from the state-distribution of any other locomotion primitive). The impact of this assumption can be further attenuated if skills are randomly re-sampled from the prior $p(z)$ within an episode of interaction with the environment. Irrespective, we can avoid making this assumption if we use the variational lower bounds from Agakov (2004), which is the second way to learn for $I(s'; z|s)$. We use the following inequality from the appendix of Hausman et al. (2018):

$$\mathcal{H}(x) \geq \int p(x, z) \log \frac{q(z|x)}{p(x, z)} dx dz$$

where q is a variational approximation to the posterior $p(z|x)$.

$$\begin{aligned} I(s'; z|s) &= -\mathbb{E}_{p(z|s)}[\mathcal{H}(s'|s, z)] + \mathcal{H}(s'|s) \\ &\geq \mathbb{E}_{p(s',z|s)}[\log q_\phi(s'|s, z)] + \mathbb{E}_{p(s',z|s)}[\log q_\alpha(z|s', s)] + \mathcal{H}(s', z|s) \\ &= \mathbb{E}_{p(s',z|s)}[\log q_\phi(s'|s, z) + \log q_\alpha(z|s', s)] + \mathcal{H}(s', z|s) \end{aligned}$$

where we have used the inequality for $\mathcal{H}(s'|s)$. Further decomposing the leftover entropy:

$$\mathcal{H}(s', z|s) = \mathcal{H}(z|s) + \mathbb{E}_{p(z|s)}[\mathcal{H}(s'|s, z)]$$

Reusing the variational lower bound for marginal entropy from Agakov (2004), we get:

$$\begin{aligned} \mathcal{H}(s'|s, z) &\geq \int p(s', a|s, z) \log \frac{q(a|s', s, z)}{p(s', a|s, z)} ds' da \\ &= -\log c + \mathcal{H}(s', a|s, z) \\ &= -\log c + \mathbb{E}_{\pi(a|s,z)}\big[\mathcal{H}(s'|s, a)\big] + \mathcal{H}(a|s, z) \end{aligned}$$

Since, the choice of posterior is upon us, we can choose $q(a|s', s, z) = 1/c$ to induce a uniform distribution for the bounded action space. Notice now, this corresponds to entropy-regularized RL when the dynamics of the system are deterministic. Even for stochastic dynamics, the analogy might be a good approximation (assuming the dynamics are not very entropic or noisy). The final objective (making the low-entropy dynamics assumption) can be written as:

$$I(s'; z|s) \geq \mathbb{E}_{p(s',z|s)}[\log q_\phi(s'|s, z) + \log q_\alpha(z|s', s) - \log p(z|s)] + \mathbb{E}_{p(z|s)}[\mathcal{H}(a|s, z)]$$

While we do not extensively experiment with this objective, we found the optimization for this objective to be unstable, possibly because of the interplay of three networks.

D THE CURSE OF DIMENSIONALITY

Unsupervised skill learning presents the possibility of learning novel behaviors without any extrinsic task reward. This is a real boon in small state spaces like that of a point-mass environment, as demonstrated by the skills learnt in Figure 7. However, the behaviors (if understood as as sequence of states) grows exponentially in the dimension of the state. We consider the problem of learning skills in the Ant environment (Todorov et al. (2012)), which has a 29 dimensional continuous state-space. Since, we are interested in locomotion perspective of the skills, we project the skills onto the x-y plane. Trajectories generated by a fixed skill can be highly entropic in the x-y space when skills are learnt on the full observation space, which makes it tough to compose them temporally on downstream tasks. On the other hand, if we restrict the observation space to the x-y space, we can get highly structured skills with consistent locomotion behavior. The restriction of the observation space can be seen as an inductive bias to guide the diversity of skills.

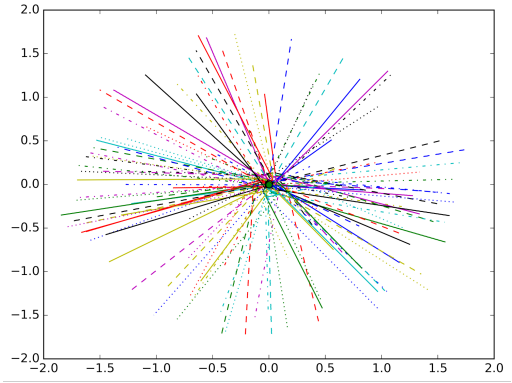

Figure 7: 128 unsupervised skills partition the 2D space of point-mass

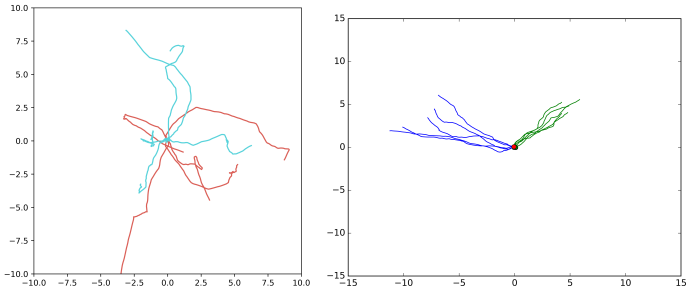

Figure 8: Trajectories generated by a fixed skill learnt by DIAYN on the full observation space (*left*); Trajectories generated by a fixed skill learnt via Skill-Dynamics on the x-y observation space (*right*)

E Interpolation in Continuous Latent Space

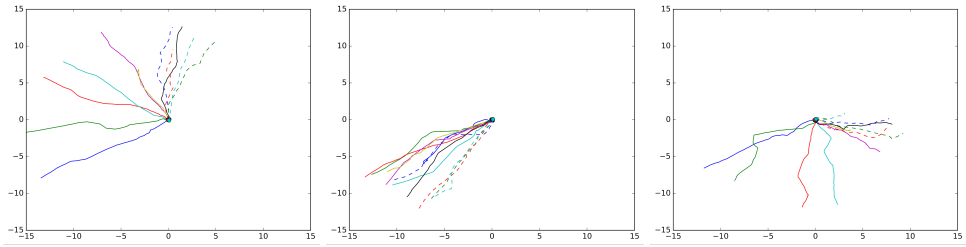

Figure 9: Interpolation in the continuous primitive space learnt using skill-dynamics on the Ant environment corresponds to interpolation in the trajectory space. (*Left*) Interpolation from $z = [1.0, 1.0]$ (solid blue) to $z = [-1.0, 1.0]$ (dotted cyan); (*Middle*) Interpolation from $z = [1.0, 1.0]$ (solid blue) to $z = [-1.0, -1.0]$ (dotted cyan); Interpolation from $z = [1.0, 1.0]$ (solid blue) to $z = [1.0, -1.0]$ (dotted cyan).