A. Theoretical analysis of existing methods

This section provides insight for why existing methods do not encourage the discovery state-covering skills from a theoretical lens. This is achieved by analyzing the reward function of these methods, and studying its asymptotic behavior for known and novel states. Our main result shows that the agent receives larger rewards for visiting known states than discovering new ones. The following subsections contain the derivation of this result, and Figure 6 provides a numerical example on a gridworld environment.

A.1. Reverse form of the mutual information

The objective for these methods is

$$I(S;Z) = \mathbb{E}_{s,z\sim p(s,z)}[\log p(z|s)] - \mathbb{E}_{z\sim p(z)}[\log p(z)] \tag{10}$$

$$\approx \mathbb{E}_{s,z\sim p(s,z)}[\log \rho_\pi(z|s)] - \mathbb{E}_{z\sim p(z)}[\log p(z)] \tag{11}$$

where the unknown posterior $p(z|s)$ is approximated by the distribution induced by the policy, $\rho_\pi(z|s)$. This distribution is estimated with a model $q_\theta(z|s)$ trained via maximum likelihood on (s,z)-tuples collected by deploying the policy in the environment. For this analysis, however, we will assume access to a perfect estimate of $\rho_\pi(z|s)$. When considering the discovery of N discrete skills under a uniform prior, the reward in Equation 5 becomes

$$r(s,z') = \log \rho_\pi(z'|s) - \log p(z') \tag{12}$$

$$= \log \rho_\pi(z'|s) + \log N \tag{13}$$

where $z' \sim p(z)$. We will assume that $\sum_{i=1}^{N} \rho_\pi(z_i|s) = 1$ in our analysis.

Maximum reward for known states. The reward function encourages policies to discover skills that visit disjoint regions of the state space where $\rho_\pi(z'|s) \to 1$:

$$r_{\text{max}} = \log 1 + \log N = \log N \tag{14}$$

Reward for previously unseen states. Note that $\rho_\pi(z|s)$ is not defined for unseen states, and we will assume a uniform prior over skills in this undefined scenario, $\rho_\pi(z|s) = 1/N, \forall z$:

$$r_{\text{new}} = \log \frac{1}{N} + \log N = 0 \tag{15}$$

Alternatively, one could add a *background* class to the model in order to assign null probability to unseen states (Capdevila et al., 2018). This differs from the setup in previous works, reason why it was considered in the analysis. However, note that the agent gets a larger penalization for visiting new states in this scenario:

$$r'_{\text{new}} = \lim_{\rho_\pi(z'|s)\to 0} \log \rho_\pi(z'|s) + \log N = -\infty \tag{16}$$

These observations explain why the learned skills provide a poor coverage of the state space.

A.2. Forward form of the mutual information

The objective for these methods is

$$I(S;Z) = \mathbb{E}_{s,z\sim p(s,z)}[\log p(s|z)] - \mathbb{E}_{s\sim p(s)}[\log p(s)] \tag{17}$$

$$= \mathbb{E}_{s,z\sim p(s,z)}[\log \rho_\pi(s|z)] - \mathbb{E}_{s\sim \rho_\pi(s)}[\log \rho_\pi(s)] \tag{18}$$

where the unknown distributions $p(s|z)$ and $p(s)$ are approximated using the stationary state-distribution, $p(s|z) \approx \rho_\pi(s|z)$ and $p(s) \approx \rho_\pi(s) = \mathbb{E}_z[\rho_\pi(s|z)]$. The stationary state-distribution is estimated with a model $q_\theta(s|z)$ trained via maximum likelihood on (s,z)-tuples collected by deploying the policy in the environment. For this analysis, however, we will assume access to a perfect estimate of $\rho_\pi(s|z)$. When considering the discovery of N discrete skills, the reward in Equation 8 can be expanded as follows:

$$r(s,z') = \log \rho_\pi(s|z') - \log \frac{1}{N} \sum_{\forall z_i} \rho_\pi(s|z_i) \tag{19}$$

$$= \log \frac{\rho_\pi(s|z')}{\sum_{\forall z_i} \rho_\pi(s|z_i)} + \log N \tag{20}$$

$$= \lim_{\epsilon\to 0} \log \frac{1}{1 + \sum_{\forall z_i \neq z'} \frac{\rho_\pi(s|z_i)+\epsilon}{\rho_\pi(s|z')+\epsilon}} + \log N \tag{21}$$

where $z', z_i \sim p(z)$ and we added $\epsilon \to 0$ in the last step to prevent division by 0.

Maximum reward for known states. As observed by Sharma et al. (2019), this reward function encourages skills to be predictable (i.e. $\rho_\pi(s|z') \to 1$) and diverse (i.e. $\rho_\pi(s|z_i) \to 0, \forall z_i \neq z'$):

$$r_{\text{max}} = \log 1 + \log N = \log N \tag{22}$$

Reward for previously unseen states. In novel states, $\rho_\pi(s|z_i) \to 0, \forall z_i$:

$$r_{\text{max}} = \lim_{\epsilon\to 0} \log \frac{1}{1 + \sum_{\forall z_i \neq z'} \frac{\epsilon}{\epsilon}} + \log N \tag{23}$$

$$= \log \frac{1}{1 + (N-1)} + \log N \tag{24}$$

$$= \log \frac{1}{N} + \log N \tag{25}$$

$$= 0 \tag{26}$$

This result shows that visiting known states instead of exploring unseen ones provides larger rewards to the agent, producing options that provide a poor coverage of the state space.

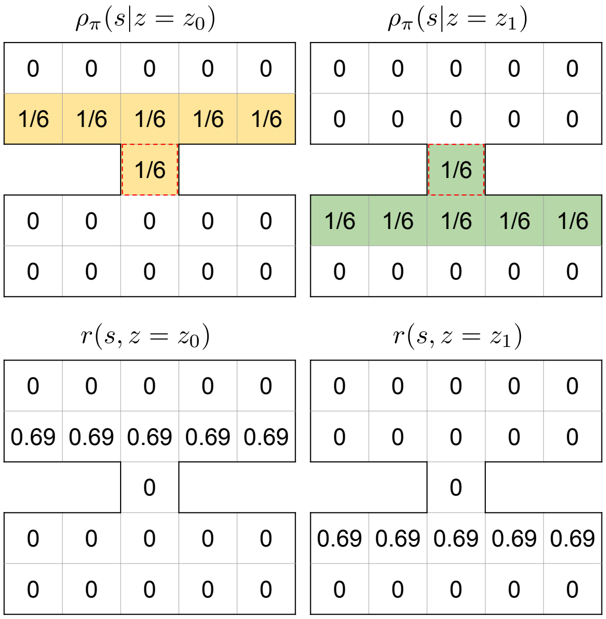

Figure 6. Analysis of the reward landscape on a toy gridworld with two skills, assuming perfect density estimation. Under this assumption, both forms of the mutual information generate the same reward landscape. Each column depicts a different skill, and all rollouts always start from the central tile which is highlighted in red. Skills are rewarded for visiting known states where they are maximally distinguishable, but receive no reward for visiting novel states.

B. Choice of mutual information's form

The main novelty of EDL is an alternative for modelling the unknown distributions, which in principle could work with either form of the mutual information. For the sake of comparison with previous works, all experiments consider discrete skills. This was achieved through a categorical posterior $p(z|s)$ that was approximated with a VQ-VAE (van den Oord et al., 2017). The encoder of the VQ-VAE takes an input x, produces output $z_e(x)$, and maps it to the closest element in the codebook, $e \in \mathbb{R}^{K\times D}$. The posterior categorical distribution $q(z|x)$ probabilities are defined as one-hot as follows:

$$q(z=k|x) = \begin{cases} 1 & \text{for } k = \text{argmin}_j ||z_e(x) - e_j||_2 \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

One could consider the reverse form of the mutual information and train the policy with a reward function as follows:

$$r(s,z) = q(z|s) \tag{28}$$

where we assumed a uniform prior over z and removed the constant $\log p(z)$ term from the reward.

We can foresee two issues with this reward function. It is sparse, i.e. many states provide no reward at all, which might hinder training unless proper exploration strategies are used (Ecoffet et al., 2019; Trott et al., 2019). A similar behavior was observed in existing methods using the reverse form of the mutual information (c.f. Figure 9). Moreover, the fact that many states produce a maximum reward of 1 might lead to unpredictable skills when paired with an entropy bonus. Such unpredictability might not be desirable when training a metra-controller to solve a downstream task by combining the learned skills (Sharma et al., 2019).

C. Implementation Details

Environments. The maze environments are adapted from the open-source implementation[3] by Trott et al. (2019). The agent does not observe the walls, whose location needs to be inferred from experience and makes exploration difficult. The initial state for each episode is sampled from a 1×1 tile. See Table 2 for details about the environments and the topology of each maze.

Parameter	Value
State space	$\mathcal{S} \in \mathbb{R}^2$
Action space	$\mathcal{A} \in [-0.95, 0.95]^2$
Episode length	50
Size: Bottleneck maze (Figure 1)	10×10
Size: Square maze (Figure 2)	5×5
Size: Corridor maze (Figure 3)	1×12
Size: Tree maze (Figure 4)	7×7

Table 2. Environment details.

RL Agents. Policy networks emit the parameters of a Beta distribution (Chou et al., 2017), which are then shifted and scaled to match the task action range. Entropy regularization is employed to prevent convergence to deterministic behaviors early in training. We use a categorical distribution with uniform probabilities for the skill prior $p(z)$. Agents are trained with PPO (Schulman et al., 2017) and the Adam optimizer (Kingma & Ba, 2014). Hyperparameters are tuned for each method independently using a grid search. See Table 3 for details.

Exploration. When relying on State Marginal Matching (SMM) (Lee et al., 2019) for exploration, we implement the version that considers a mixture of policies with a uniform target distribution $p^*(s)$. The density model $q(s)$ is approximated with a VAE. We use states in the replay buffer

[3] https://github.com/salesforce/sibling-rivalry

Hyperparameter	Value
Discount factor	0.99
λ_{GAE}	0.98
λ_{entropy}	$\{0.001, 0.005, 0.01, 0.025\}$
$\epsilon_{\text{SiblingRivalry}}$	$\{2.5, 5.0, 7.5\}$
Optimizer	Adam
Learning rate	$\{0.0003, 0.001\}$
Learning rate schedule	Constant
Advantage normalization	Yes
Input normalization	{Yes, No}
Hidden layers	2
Units per layer	128
Non-linearity	ReLU
Horizon	2500
Batch size	250
Number of epochs	4

Table 3. Hyperparameters used in the experiments. Values between brackets were used in the grid search, and tuned independently for each method.

as a non-parametric approach to sampling from the desired $p(s)$ (Warde-Farley et al., 2019). Sampling states from the replay buffer is similar to a uniform Historical Averaging strategy. This worked well in our experiments, but exponential sampling strategies might be needed in other environments to avoid oversampling states collected by the initially random policies (Hazan et al., 2019). Our implementation follows the open-source code released by the authors[4], which relies on SAC for policy optimization. Hyperparameters are tuned for each environment independently using a grid search. See Table 4 for details.

Skill discovery. The skill discovery stage in the proposed method is done with a VQ-VAE (van den Oord et al., 2017), which allows learning discrete latents. We implement the version that relies on a commitment loss to learn the dictionary. The size of the codebook is set to the number of desired skills. Hyperparameters are tuned for each environment and exploration method independently using a grid search. See Table 5 for details.

D. Figure details

All experiments in the paper consider agents that learn 10 skills. This value was selected to provide a good balance between learning a variety of behaviors and ease of visualization. Given the stochastic nature of the learned policies, we report 20 rollouts per skill. When visualizing states visited by a random policy, we collect 100 rollouts with each (untrained) skill. Trajectories from these skills highly overlap with each other, so we use a single color for all of them to reduce clutter.

Hyperparameter	Value
Discount factor	0.99
Target smoothing coefficient	0.005
Target update interval	1
α_{entropy}	$\{0.1, 1, 10\}$
β_{VAE}	$\{0.01, 0.1, 1\}$
Optimizer	Adam
Policy: Learning rate	0.001
SMM discriminator: Learning rate	0.001
VAE: Learning rate	0.01
Learning rate schedule	Constant
Policies in the mixture	4
Input normalization	No
Policy: Hidden layers	2
SMM discriminator: Hidden layers	2
VAE encoder: Hidden layers	2
VAE decoder: Hidden layers	2
Units per layer	128
Non-linearity	ReLU
Gradient steps	1
Batch size	128
Replay buffer size	50k

Table 4. Hyperparameters used for exploration using SMM. Values between brackets were used in the grid search, and tuned independently for each environment. Training ends once the buffer is full.

E. Additional visualizations

We include visualizations that provide further insight about the results presented in the paper, and that could not be included there due to space constraints. These include the goal states discovered by methods using the forward for of the mutual information (Figure 7), visualization of the reward landscape of each method (Figures 8, 9 and 10), and additional skill interpolations (Figure 11).

[4] https://github.com/RLAgent/state-marginal-matching

Hyperparameter	**Value**
Code size	16
$\beta_{\text{commitment}}$	$\{0.25, 0.5, 0.75, 1.0, 1.25\}$
Optimizer	Adam
Learning rate	0.0002
Learning rate schedule	Constant
Batch size	256
Number of samples	4096
Input normalization	Yes
Encoder: Hidden layers	2
Decoder: Hidden layers	2
Units per layer	128
Non-linearity	ReLU

Table 5. Hyperparameters used for training the VQ-VAE in the skill discovery stage. Values between brackets were used in the grid search, and tuned independently for each environment and exploration method.

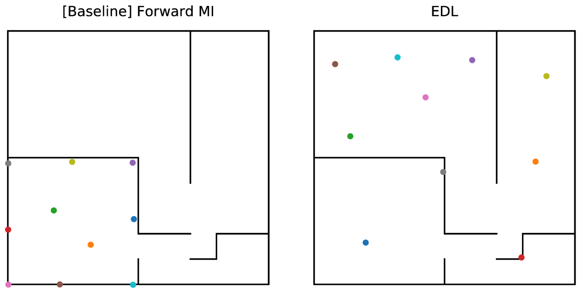

Figure 7. Goal states discovered by methods using the forward form of the mutual information in Figure 1. We define a goal state as the most likely state under $q_\phi(s|z)$ for each skill, i.e. $g_i = \text{argmax}_s q_\phi(s|z_i)$. The baseline method relies on the stationary state-distribution induced by the policy to discover goals. This policy seldom leaves the initial room, limiting the goals that can be discovered. In contrast, the uniform distribution over states in EDL enables the discovery of goals across the whole maze.

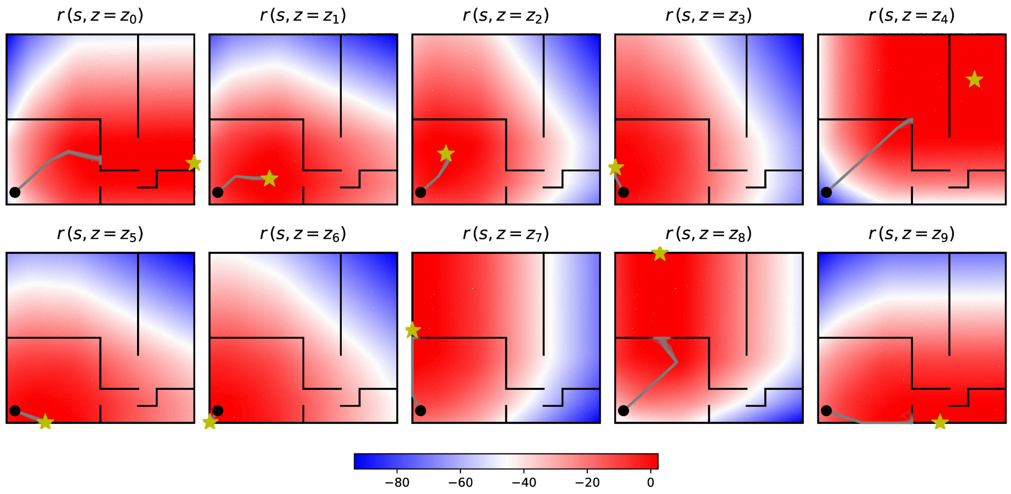

Figure 8. Reward landscape per skill at convergence for the agent in Figure 1 (left). Trajectories from each skill starting from the black dot are plotted in gray. The yellow star indicates the point of maximum reward for each skill. For some skills, this point belongs to an unexplored region of the state space, contrary to the intuition in Section A. Note that this is due to the Gaussian assumption over $p(s|z)$ in the density model.

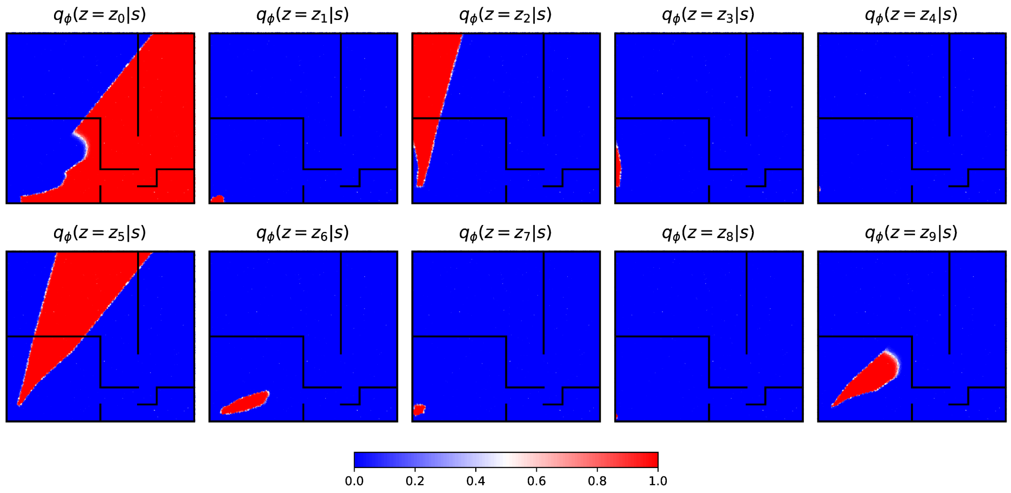

Figure 9. Approximate posterior $q_\phi(z|s)$ at convergence for the agent in Figure 1 (middle). Recall that the reward function for this agent is $r(s, z) = \log q_\phi(z|s) - \log p(z)$, and $\log p(z)$ is constant in our experiments due to the choice of prior over latent variables. The state space is partitioned in disjoint regions, so that skills only need to enter their corresponding region in order to maximize reward. Note how $q_\phi(z|s)$ extrapolates this partition to states that have never been visited by the policy. When combined with an entropy bonus, this reward landscape results in skills that produce highly entropic trajectories within each region.

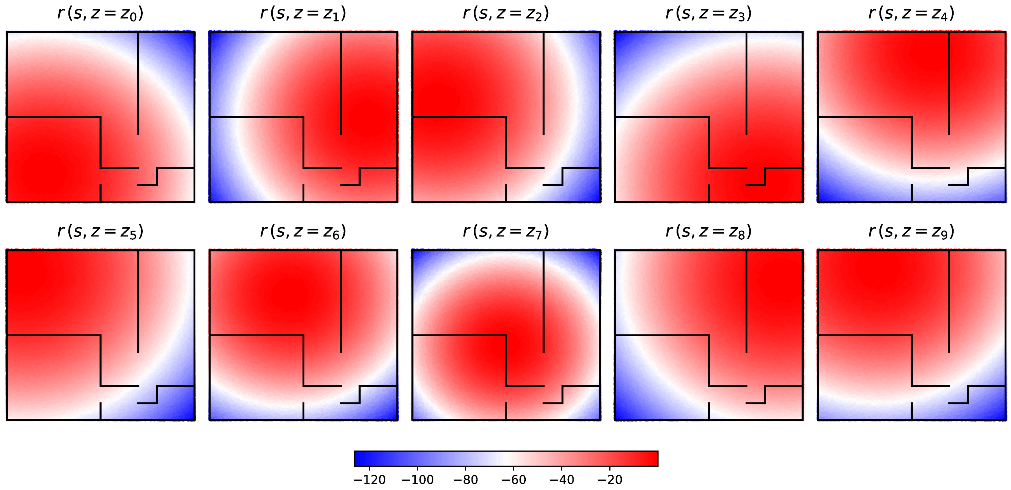

Figure 10. Reward landscape per skill at convergence for the agent in Figure 1 (right). The reward functions follow a bell shape centered at each of the centroids in Figure 7 (right). These are dense signals that ease optimization, but training is prone to falling in local optima due to their deceptive nature.

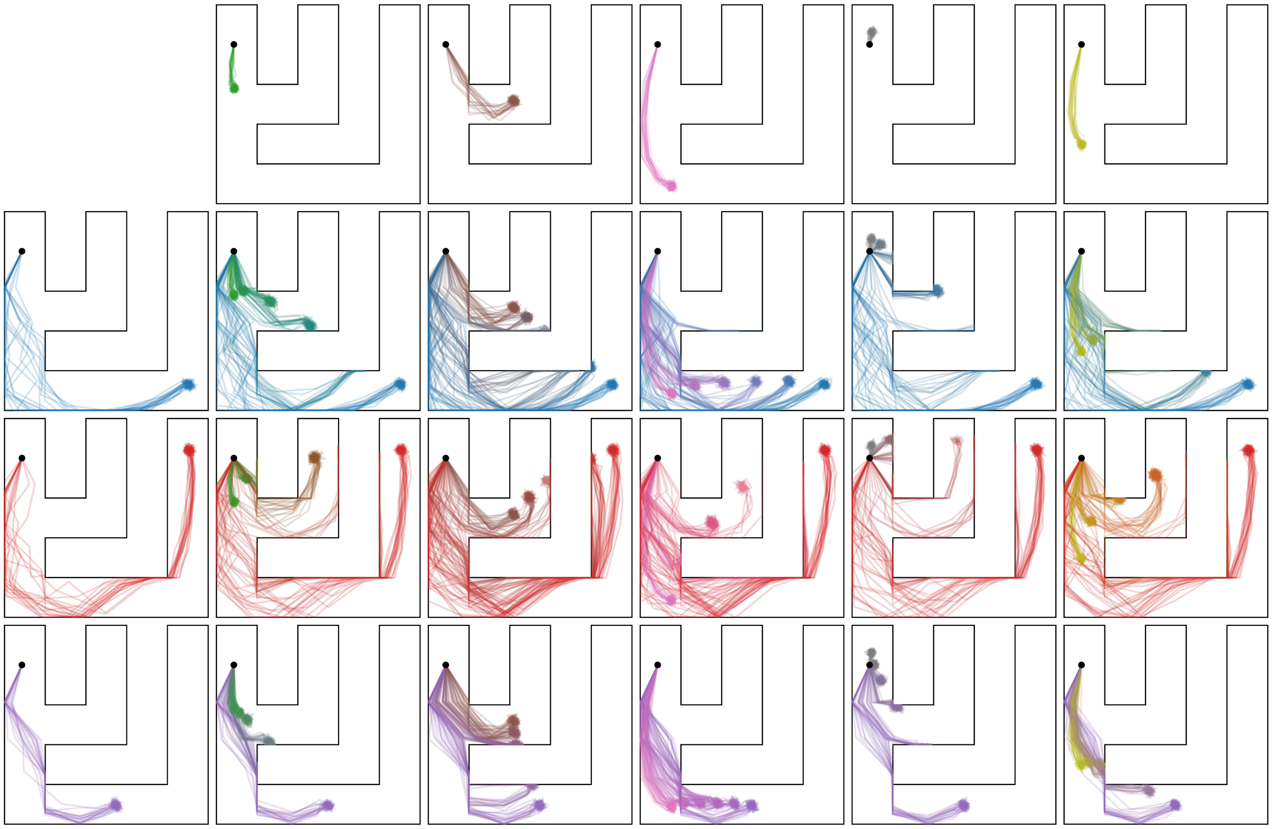

Figure 11. Interpolating skills learned by EDL. Interpolation is performed at the latent variable level by blending the z vector of two skills. The first row and column show the original skills being interpolated, which were selected randomly from the set of learned options. When plotting interpolated skills, we blend the colors used for the original skills.

References

Achiam, J., Edwards, H., Amodei, D., and Abbeel, P. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. Hindsight experience replay. In *NIPS*, 2017.

Bacon, P.-L., Harb, J., and Precup, D. The option-critic architecture. In *AAAI*, 2017.

Barber, D. and Agakov, F. V. The IM algorithm: a variational approach to information maximization. In *NIPS*, 2003.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H. P., and Silver, D. Successor features for transfer in reinforcement learning. In *NIPS*, 2017.

Barreto, A., Borsa, D., Quan, J., Schaul, T., Silver, D., Hessel, M., Mankowitz, D., Zidek, A., and Munos, R. Transfer in deep reinforcement learning using successor features and generalised policy improvement. In *ICML*, 2018.

Beattie, C., Leibo, J. Z., Teplyashin, D., Ward, T., Wainwright, M., Küttler, H., Lefrancq, A., Green, S., Valdés, V., Sadik, A., et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *NIPS*, 2016.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 2013.

Borsa, D., Barreto, A., Quan, J., Mankowitz, D., Munos, R., van Hasselt, H., Silver, D., and Schaul, T. Universal successor features approximators. In *ICLR*, 2019.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Capdevila, J., Cerquides, J., and Torres, J. Mining urban events from the tweet stream through a probabilistic mixture model. *Data mining and knowledge discovery*, 2018.

Chou, P.-W., Maturana, D., and Scherer, S. Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution. In *ICML*, 2017.

Conti, E., Madhavan, V., Such, F. P., Lehman, J., Stanley, K. O., and Clune, J. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *arXiv preprint arXiv:1712.06560*, 2017.

Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. Robots that can adapt like animals. *Nature*, 2015.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., and Clune, J. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *ICML*, 2018.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. In *ICLR*, 2019.

Florensa, C., Duan, Y., and Abbeel, P. Stochastic neural networks for hierarchical reinforcement learning. In *ICLR*, 2017.

Florensa, C., Degrave, J., Heess, N., Springenberg, J. T., and Riedmiller, M. Self-supervised learning of image embedding for continuous control. *arXiv preprint arXiv:1901.00943*, 2019.

Fox, R., Krishnan, S., Stoica, I., and Goldberg, K. Multi-level discovery of deep options. *arXiv preprint arXiv:1703.08294*, 2017.

Frans, K., Ho, J., Chen, X., Abbeel, P., and Schulman, J. Meta learning shared hierarchies. In *ICLR*, 2018.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Gregor, K., Rezende, D. J., and Wierstra, D. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

Guss, W. H., Codel, C., Hofmann, K., Houghton, B., Kuno, N., Milani, S., Mohanty, S., Liebana, D. P., Salakhutdinov, R., Topin, N., et al. The minerl competition on sample efficient reinforcement learning using human priors. *arXiv preprint arXiv:1904.10079*, 2019.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *ICML*, 2017.

Hansen, S., Dabney, W., Barreto, A., Van de Wiele, T., Warde-Farley, D., and Mnih, V. Fast task inference with variational intrinsic successor features. In *ICLR*, 2020.

Hazan, E., Kakade, S. M., Singh, K., and Van Soest, A. Provably efficient maximum entropy exploration. In *ICML*, 2019.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Hénaff, O. J., Razavi, A., Doersch, C., Eslami, S., and Oord, A. v. d. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Dulac-Arnold, G., et al. Deep q-learning from demonstrations. *arXiv preprint arXiv:1704.03732*, 2017.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *NIPS*, 2016.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. Vime: Variational information maximizing exploration. In *NIPS*, 2016.

Jabri, A., Hsu, K., Gupta, A., Eysenbach, B., Levine, S., and Finn, C. Unsupervised curricula for visual meta-reinforcement learning. In *NeurIPS*, 2019.

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks. In *ICLR*, 2017.

Jaderberg, M., Czarnecki, W. M., Dunning, I., Marris, L., Lever, G., Castaneda, A. G., Beattie, C., Rabinowitz, N. C., Morcos, A. S., Ruderman, A., et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 2019.

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., et al. In-datacenter performance analysis of a tensor processing unit. In *ISCA*, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *ICLR*, 2014.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 2015.

Lee, L., Eysenbach, B., Parisotto, E., Xing, E., Levine, S., and Salakhutdinov, R. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

Lehman, J. and Stanley, K. O. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 2011a.

Lehman, J. and Stanley, K. O. Novelty search and the problem with objectives. In *Genetic programming theory and practice IX*. Springer, 2011b.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. In *ICLR*, 2016.

Liu, H., Trott, A., Socher, R., and Xiong, C. Competitive experience replay. In *ICLR*, 2019.

Lynch, C., Khansari, M., Xiao, T., Kumar, V., Tompson, J., Levine, S., and Sermanet, P. Learning latent plans from play. *arXiv preprint arXiv:1903.01973*, 2019.

Meyerson, E., Lehman, J., and Miikkulainen, R. Learning behavior characterizations for novelty search. In *GECCO*, 2016.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

Mohamed, S. and Rezende, D. J. Variational information maximisation for intrinsically motivated reinforcement learning. In *NIPS*, 2015.

Mouret, J.-B. and Clune, J. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

Nachum, O., Gu, S. S., Lee, H., and Levine, S. Data-efficient hierarchical reinforcement learning. In *NeurIPS*, 2018.

NVIDIA. Nvidia tesla v100 gpu architecture. 2017.

Oh, J., Guo, Y., Singh, S., and Lee, H. Self-imitation learning. In *ICML*, 2018.

Osband, I., Van Roy, B., and Wen, Z. Generalization and exploration via randomized value functions. In *ICML*, 2016.

Parr, R. and Russell, S. J. Reinforcement learning with hierarchies of machines. In *NIPS*, 1998.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.

Pong, V. H., Dalal, M., Lin, S., Nair, A., Bahl, S., and Levine, S. Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*, 2019.

Precup, D. Temporal abstraction in reinforcement learning. 2001.

Pugh, J. K., Soros, L. B., and Stanley, K. O. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 2016.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

Salge, C., Glackin, C., and Polani, D. Empowerment – an introduction. In *Guided Self-Organization: Inception*. Springer, 2014.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. In *ICML*, 2015.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *ICML*, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *Nature*, 2017.

Sukhbaatar, S., Lin, Z., Kostrikov, I., Synnaeve, G., Szlam, A., and Fergus, R. Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv preprint arXiv:1703.05407*, 2017.

Sutton, R. S., Precup, D., and Singh, S. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 1999.

Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, O. X., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. # exploration: A study of count-based exploration for deep reinforcement learning. In *NIPS*, 2017.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *IROS*, 2012.

Trott, A., Zheng, S., Xiong, C., and Socher, R. Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards. In *NeurIPS*, 2019.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. Neural discrete representation learning. In *NeurIPS*, 2017.

Večerík, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., and Riedmiller, M. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 2019.

Warde-Farley, D., Van de Wiele, T., Kulkarni, T., Ionescu, C., Hansen, S., and Mnih, V. Unsupervised control through non-parametric discriminative rewards. In *ICLR*, 2019.