Relative Variational Intrinsic Control

Kate Baumli, David Warde-Farley, Steven Hansen, Volodymyr Mnih
DeepMind
{baumli, dwf, stevenhansen, vmnih} @google.com

Abstract

In the absence of external rewards, agents can still learn useful behaviors by identifying and mastering a set of diverse skills within their environment. Existing skill learning methods use mutual information objectives to incentivize each skill to be diverse and distinguishable from the rest. However, if care is not taken to constrain the ways in which the skills are diverse, trivially diverse skill sets can arise. To ensure useful skill diversity, we propose a novel skill learning objective, Relative Variational Intrinsic Control (RVIC), which incentivizes learning skills that are distinguishable in how they change the agent's relationship to its environment. The resulting set of skills tiles the space of affordances available to the agent. We qualitatively analyze skill behaviors on multiple environments and show how RVIC skills are more useful than skills discovered by existing methods when used in hierarchical reinforcement learning.

Introduction

Deep reinforcement learning (RL) methods have demonstrated the ability to successfully learn to achieve a task defined by a reward function in a variety of domains (Sutton and Barto 1998; Mnih et al. 2015; Silver et al. 2017a,b). However, the knowledge obtained by an RL agent is usually highly specific to the particular task it was trained on, and is not well suited towards transfer or generalization (Whiteson et al. 2011; Cobbe et al. 2019).

In contrast, humans can obtain and maintain repurposable knowledge about their environments and how they can behave in them, even in the absence of an explicit end goal or reward. We maintain sets of skills that can transfer from one task to the next. For example, we can learn a general skill to throw an object and we can apply slightly modified versions of that skill in different contexts to enable us to throw a paper airplane, a baseball, or a water balloon.

Mutual information based skill discovery methods such as Variational Intrinsic Control (VIC) (Gregor, Rezende, and Wierstra 2017) and Diversity Is All You Need (DIAYN) (Eysenbach et al. 2019) offer a promising direction for increasing practical applicability of deep reinforcement learning methods to real-world problems. By first learning useful skills purely from intrinsic rewards, agents can repurpose the learned skills to solve more challenging downstream tasks specified by extrinsic rewards.

These methods rely on the idea of inverse predictability for skill learning, i.e. that it should be possible to infer the skill used to generate a trajectory from the states in the trajectory. This requires each skill to be distinguishable from the others, ensuring a diverse set of skills. The conditioning of the inverse predictor varies from relying on any state along the trajectory in DIAYN to using the whole trajectory in VALOR (Achiam et al. 2018). We focus on a common intermediate case, introduced by VIC, where the inverse predictor relies on the first and last states in the trajectory.

In practice, these inverse predictability objectives are often trivial enough to achieve using only information from the end of the trajectory that they ignore given information about the beginning of the trajectory. This results in a set of skills that simply partitions the state space based on where the agent ends up at the end of each skill. We argue that this behavior is undesirable because it limits the usefulness (in terms of transferability and generalizability) of the skill set. A set of skills which each correspond to a specific target state (or a small cluster of nearby target states) is limited to only ever going between those specific regions of the state space.

In this paper, we propose a way of acquiring more composable and generalizable skills, making note that a skill's behavior should differ depending on the current state of the agent. For example, the skills an agent is afforded while sitting on an airplane with a fastened seatbelt are different than the skills afforded on a spacious football field. However, skills should generalize between different areas of the state space; picking up a dropped pen on the airplane and picking up a football are in a sense the same skill, but performing the "picking up" skill doesn't arrive at the same target state in both cases. After performing the skill, the agent's new state should be on the plane with a pen in hand and on the field with a football in hand, respectively. In other words, the final state that the skill arrives at should depend on both the skill itself and the agent's initial state.

To incentivize more meaningfully diverse skill sets, we propose a new skill learning method, Relative Variational Intrinsic Control (Relative VIC / RVIC) that utilizes two inverse predictors: one which relies on the first and last states of the trajectory and one that only relies on the last

state to predict the skill. Incentivizing the agent to maximize predictability with respect to the former and minimize predictability with respect to the latter, skills are forced to be relative to the agent's state at the beginning of the skill, guarding against state space partitioning. Instead, RVIC skills partition the space of affordances (Gibson 1977)– that is, they are diverse in the way that they change the agent's relationship to its environment.

In this paper, we introduce the Relative VIC skill learning method, qualitatively examine the skills on several domains, and show that they are more useful in a hierarchical RL set up than skills learned by existing methods.

Background

We consider a skill-conditional policy $\pi_\theta(\cdot;\Omega)$ which maps states s to a distribution over actions a. In this work, we will assume that Ω is discrete although the algorithms can also be applied to continuous skills. We will train this policy on episodes wherein we sample a skill Ω uniformly from the set of available skills and follow it for a fixed number of steps, T. We refer to the states $s_0, \ldots, s_T$ and corresponding actions $a_1, \ldots, a_T$ as a skill trajectory or a skill episode.

Gregor, Rezende, and Wierstra (2017) introduced the idea of discovering skills by maximizing the mutual information between Ω and s_T, the final state of a skill trajectory, conditioned on the first state s_0. Their approach, known as Variational Intrinsic Control (VIC), relies on the well known lower bound of Barber and Agakov (2004) on the mutual information. When applied to the VIC objective, the lower bound takes the form

$$\begin{aligned} I(s_T, \Omega | s_0) &= H(\Omega|s_0) - H(\Omega|s_T, s_0) \\ &\geq H(\Omega|s_0) + \mathbb{E}_\Omega \mathbb{E}_{s_0, s_T \sim \pi_\Omega} \log q_\phi(\Omega|s_T, s_0) \end{aligned} \tag{1}$$

where q is a variational distribution. Following Eysenbach et al. (2019), we assume that skills Ω are sampled from a fixed distribution. Optimizing this objective involves two separate optimization steps. The first performs gradient ascent on $\log q_\phi(\Omega|s_T, s_0)$ with respect to variational parameters ϕ. This corresponds to training q to be an inverse predictor which can infer the skill Ω used to generate the skill episode from the first and final states. The second optimization step involves optimizing the parameters of the skill-conditional policy $\pi_\theta(\cdot;\Omega)$ using a reinforcement learning algorithm with a sparse reward that is proportional to $\log q_\phi(\Omega|s_T, s_0)$ at time T and zero for all other time steps[1].

Relative Variational Intrinsic Control

A drawback of the VIC approach is that, given sufficient trajectory lengths, it tends to yield skills which partition the state space with respect to their terminal states. When this occurs, the inverse predictor q learns to ignore the initial state s_0 and infers skills based only on the final state s_T while the skill-conditioned policy learns to go to a different unique state for each skill Ω.

[1]Note that we omit the constant additive term that derives from the entropy of the skill prior.

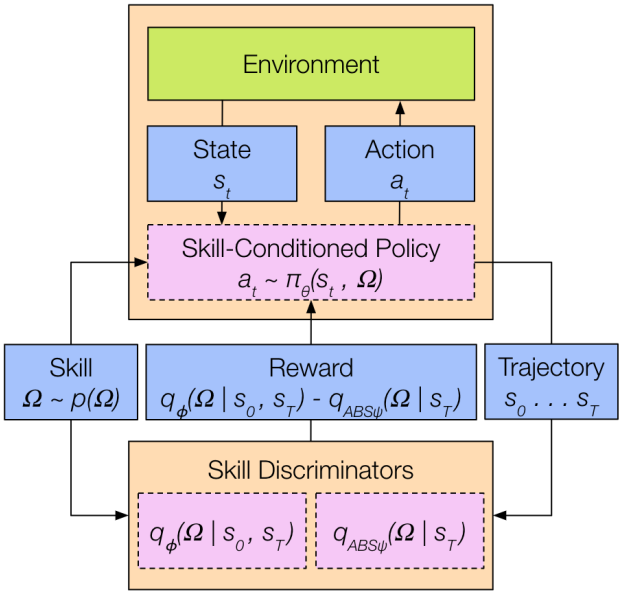

Figure 1: Relative VIC: A skill-conditioned policy interacts with the environment yielding a sampled trajectory. Two skill predictors are trained to predict the skill from the first and last states of the trajectory and only the last state in the trajectory respectively. The difference between the probability assigned by the predictors to the skill used to generate the trajectory is given to the policy as reward to incentivize learning a set of skills that is diverse in how each skill changes the agent's relationship to the environment.

This can be explained by noting that the entropy of the skill given the final state is an upper bound on the entropy of the skill given the full trajectory:

$$0 \leq H(\Omega|s_T, s_0) \leq H(\Omega|s_T), \tag{2}$$

so conditioning on the initial state will not increase the mutual information if the skills are perfectly predictable from just the final state. In large or infinite environments, there are no shortage of diverse (yet potentially meaningless) final states from which to perfectly predict skills, driving the VIC objective to collapse into state-space partitioning.

Inspired by this observation, we introduce a second inverse predictor, $q_\psi^{\text{abs}}(\Omega|s_T)$, which is trained to predict a skill's identity from the final state of the trajectory alone. We dub this predictor q^{abs} because it can only base its predictions on the "absolute" state of the environment upon skill termination, while q can make use of the agent's initial state in order to utilize information about the final state *relative* to the initial state. We then train the policy *adversarially* with respect to this secondary objective: we reward discriminability by q while simultaneously punishing discriminability by q^{abs}.

Minimizing the difference of these predictors also has its own information theoretic interpretation. If both predictors perfectly estimate their respective conditional distributions, then the optimization process amounts to maximizing the mutual information between the skills and the initial state,

given the final state:

$$
\begin{aligned}
I(s_0, \Omega | s_T) &= H(\Omega | s_T) - H(\Omega | s_T, s_0) \\
&= \mathbb{E}_\Omega \mathbb{E}_{\pi_\Omega} \log p(\Omega | s_T, s_0) - \log p(\Omega | s_T) \\
&\approx \mathbb{E}_\Omega \mathbb{E}_{\pi_\Omega} \log q_\phi(\Omega | s_T, s_0) - \log q_\psi^{\text{abs}}(\Omega | s_T).
\end{aligned} \tag{3}
$$

Since the skills are sampled independently from the initial state, maximizing this mutual information implies that each skill must communicate information about the initial state through its policy. This reinforces our intuition that our adversarial predictors should yield *relative* skills.

Note that we are not maximizing a lower bound on this mutual information, as minimizing the discriminability of q^{abs} *upper* bounds $H(\Omega | s_T)$. Previous work has shown this to be unproblematic (Sharma et al. 2020), likely due to these conditional distributions being relatively easy to accurately approximate with modern over-parametrized models.

In the experiments that follow, we instantiate q and q^{abs} as neural networks which share parameters, specifically a convolutional sub-network which processes pixel observations. The observations s_0 and s_T are each processed independently with this sub-network. The resulting representation of s_T is then processed by a multi-layer perceptron whose parameters are specific to q^{abs}, while the concatenated representations of s_0 and s_T are processed by another multi-layer perceptron representing q.

For the policy, we train R2D2 (Kapturowski et al. 2019) on fixed length "skill episodes" constructed on top of environment-specified episodes such that the final observation of one skill episode becomes the initial observation of the next, with each actor periodically resetting the base environment, following Warde-Farley et al. (2019). While Gregor, Rezende, and Wierstra (2017) employed rewards in the log domain, we find that a difference of probabilities $q_\phi(\Omega | s_T, s_0) - q_\psi^{\text{abs}}(\Omega | s_T)$ works well in practice.

See Figure 1 and Algorithm 1 for further summary of the Relative Variational Intrinsic Control method.

Algorithm 1: RELATIVE VIC

Input : Environment dynamics p_E, behavior policy π_θ, policy parameters θ, relative predictor parameters ϕ, absolute predictor parameters ψ, skill episode length T, discount γ, final step discount γ_T, skill episode count M.

repeat
- $s_0 \sim p(s_0)$ /* Reset the environment */
- **for** $m \leftarrow 1 \dots M$ **do**
  - $\Omega \sim p(\Omega)$
  - **for** $t \leftarrow 1 \dots T$ **do**
    - Observe state s_{t-1}
    - $a_t \sim \pi_\theta(a | s_{t-1}, \Omega)$
    - $s_t \sim p_E(s_t | s_{t-1}, a_t)$
  - **end**
  - /* Give same reward, post-hoc, for all steps */
  - $r_1^T \leftarrow q_\phi(\Omega | s_T, s_0) - q_\psi^{\text{abs}}(\Omega | s_T)$
  - $\gamma_1^{T-1} \leftarrow \gamma$ /* γ_T given as separate input */
  - Update θ with an off-policy reinforcement learning algorithm on $(a_1^T, s_0^T, r_1^T, \gamma_1^T)$,
  - Update ϕ by ascending $\nabla_\phi \log q_\phi(\Omega | s_T, s_0)$
  - Update ψ by ascending $\nabla_\psi \log q_\psi^{\text{abs}}(\Omega | s_T)$
  - **if** $m < M$ **then** $s_0 \leftarrow s_T$
- **end**

until *termination*

Experiments

In this section, we evaluate the skills learned by RVIC both qualitatively and quantitatively (via hierarchical reinforcement learning) on the DeepMind Control Suite (Tassa et al. 2018) and Atari 2600 games from The Arcade Learning Environment (ALE) (Bellemare et al. 2013). All experiments on both Atari and DeepMind Control Suite domains are done from pixels. On experiments on the DeepMind Control Suite, we first discretize the continuous action space to enable value learning with R2D2. Using a fixed discount at every time step and allowing bootstrapping between skill episodes worked best for experiments on the DeepMind Control Suite, while experiments on Atari performed better when given a zero discount at the end of each skill episode, in addition to a zero discount upon loss of life. All final values used for hyperparameters can be found in Table 1 in the Appendix.

As a baseline for all experiments, we compare against VIC (Gregor, Rezende, and Wierstra 2017) with a fixed skill prior (as demonstrated to work better in DIAYN (Eysenbach et al. 2019)). DIAYN is explicitly shown in the analysis of Eysenbach et al. (2019) to learn skills that partition the state-space, though we do not compare against DIAYN directly, as it does not attempt to condition the inverse predictor on the initial state, instead conditioning it on any independently drawn sample state from the trajectory. Therefore, VIC, which tries (but often fails) to condition the inverse predictor on the initial state is a more relevant baseline to compare against. We do not compare against Sharma et al. (2020) as the method is constrained to working with crafted features rather than pixels, and it is non-obvious how to adapt DADS to work for experiments from pixels. Our choice of baseline is therefore the closest ablation to Relative VIC, as the only major difference between the two methods is the two-predictor reward objective used by RVIC.

For both methods, we experimented with giving skill rewards in dense way, where the reward calculated for the entire skill episode $q_\phi(\Omega | s_0, s_T) - q_\psi^{\text{abs}}(\Omega | s_T)$ is given at every timestep t in the skill trajectory, which is easily done in an off-policy learning set up. We found this dense reward to work better empirically for both methods than the sparse reward used in VIC. Skills from both methods were trained for 175 million learner steps before being analyzed qualitatively and used in HRL experiments.

Qualitative Results

For the qualitative experiments, both methods learn a set of 16 skills with a skill episode length of 90 for experiments on the DeepMind Control Suite and 25 for experi-

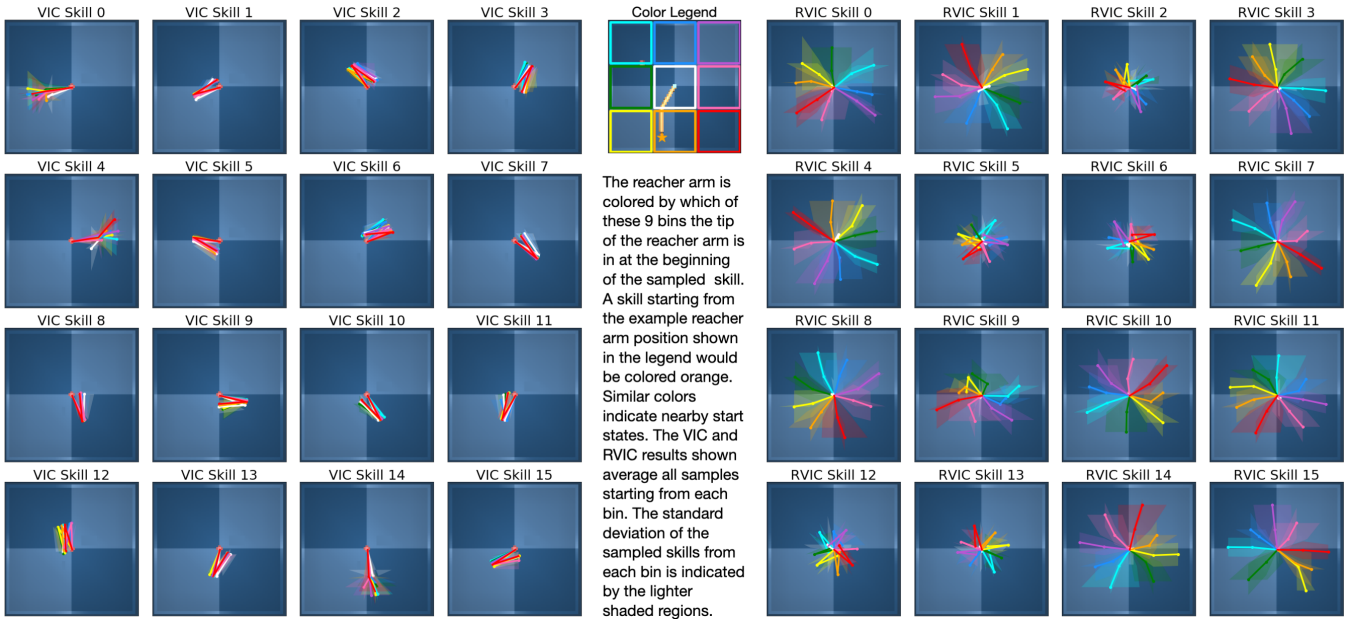

Figure 2: VIC (left) and RVIC (right) skills on Reacher in the DeepMind Control Suite. Each facet shows the end state of the reacher arm for a skill overlaid across multiple runs with different start states (indicated by color - see legend for details).

ments on Atari. We want to observe how each skill behaves from various start states. We visualize how skills learned by the two skill learning methods affect the controllable environment state. In the Reacher domain on the DeepMind Control Suite, the controllable part of the environment is the position of the two-link reacher arm. Visualizing how skills from each method change the position of the reacher arm from various starting positions yields insights into the difference between baseline VIC and RVIC.

While the baseline VIC skills on the left in Figure 2 reliably reach diverse end states, the skills disregard their starting positions (indicated by color) in achieving that diversity. Regardless of where the reacher arm starts, each skill only goes to a single end state. Additionally, note how the baseline VIC skill diversity only spans a portion of the state space and the reacher arm is rarely even opened as it is easy enough to achieve the VIC objective without doing so. In contrast, the set of RVIC skills shown partitions the rotation space of both joint angles, therefore covering more of the relevant state space. In other words, an RVIC skill will rotate the joint angles a certain amount from wherever the reacher arm starts. This can be seen in the right side of Figure 2 as the ordering of colors (where nearby colors indicate nearby start states) is preserved in each skill, but rotated to a different degree for each skill.

Similarly, on Atari games Seaquest and Montezuma's Revenge, we visualize how different skills move the avatar through the X-Y plane, since the agent has direct control over only the (X, Y) coordinates of the avatar. We extract this information about the avatar coordinates from RAM only for the visualizations, as all learning is done directly from pixels. To clearly show how each skill learning method utilizes (or disregards) information about the start state, we roll out sample skills from every possible starting (X, Y) coordinate in the frame (excluding positions in Montezuma's Revenge where the avatar is free-falling through the air and has no control over its behavior). The samples are then divided into 16 bins (uniform 4x4 grid for Seaquest; meaningfully different state areas such as platforms or ladders in Montezuma's Revenge). Sample trajectories from each bin are then averaged, colored such that similar colors indicate nearby bins, and plotted along with their standard deviation in Figure 3. The final state of each averaged trajectory is denoted with a star.

The baseline VIC method on Seaquest obtains skill diversity via state-space partitioning, converging to a single final state no matter where the trajectory starts out. Partial state-space partitioning behavior of VIC can also be seen to a lesser degree on Montezuma's Revenge. Due to the difficulty in Montezuma's Revenge of exploring to the bottom parts of the frame during unsupervised training, the VIC skills seem to only pay attention to partitioning the top part of the frame with some skills clearly choosing a target state, for example, the top of the center ladder in skill 13. When the avatar coordinates are set to the bottom of the frame at analysis time, the agent has probably never seen those states before and likely does not know how to reach the skill's target states. When performing these VIC skills from the unfamiliar positions at the bottom of the frame, behaviors are inconsistent with the behaviors learned for the top of the frame, often doing nothing at all. This illustrates the dangers of skills that partition the state space as behavior does not generalize well to the unseen states at the bottom of the frame and skills do not perform consistent behavior everywhere. The skills learned by RVIC on both of these games are different depending on where the agent starts out and can loosely be

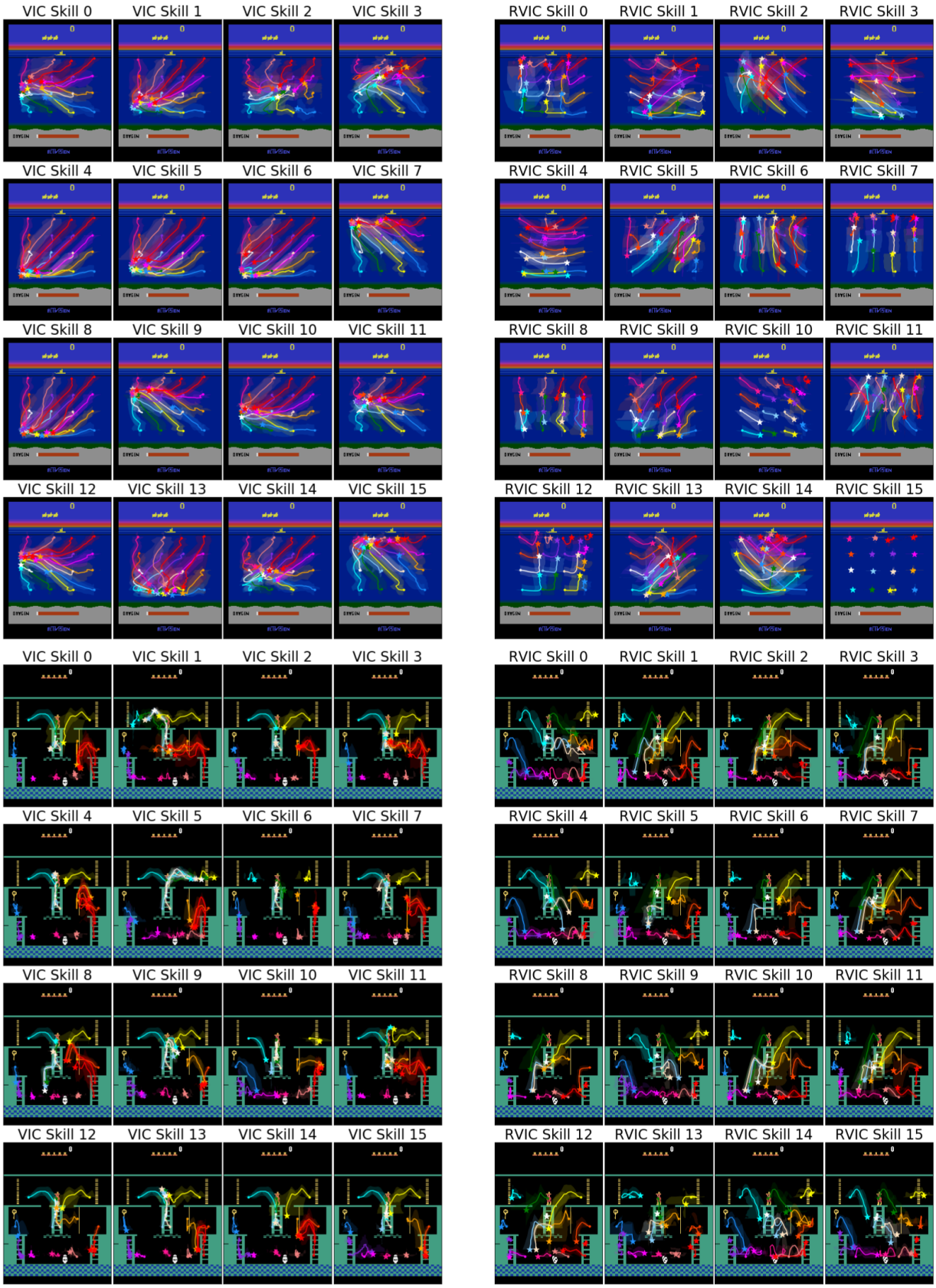

Figure 3: Learned VIC (left) and RVIC (right) skills on Seaquest (top) and Montezuma's Revenge (bottom).

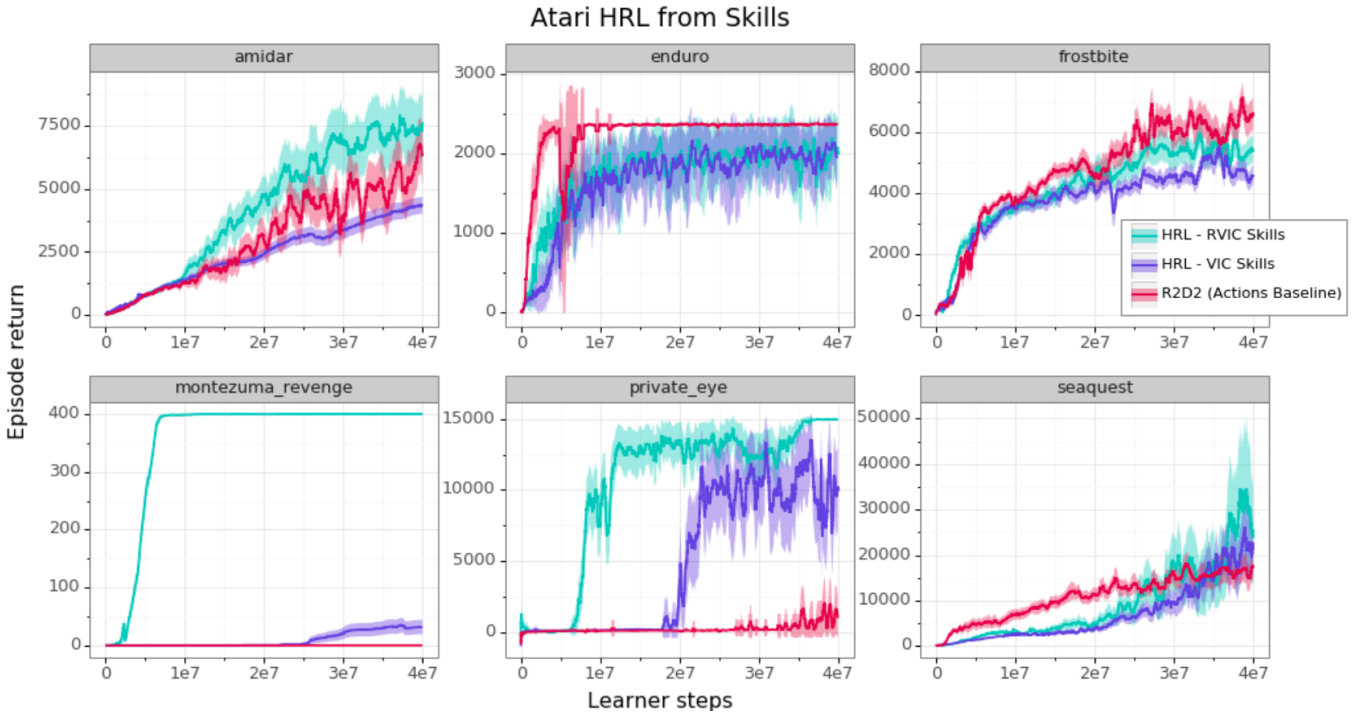

Figure 4: HRL Results. The option to execute a pre-trained skill for a fixed duration is added to the set of possible actions.

seen as directional skills that are consistent, generalizable, and composable.

Hierarchical RL on Skills

Hierarchical Reinforcement Learning (HRL) decomposes the RL problem into temporally extended sub-problems solved with multiple hierarchical levels of planning or control (Sutton, Precup, and Singh 1999; Precup 2001). To test the usefulness of the learned skills, we perform HRL experiments on six Atari games, using the pre-trained skill policies as the low-level controller. After training skills for both the baseline and RVIC, we introduce a second phase of training where a meta-controller agent can use the pre-trained, frozen skill policies to maximize the extrinsic rewards from the environment. In this second phase, a meta-controller learns a policy that at each timestep can either act with a temporally-extended skill or a primitive action for a single timestep. From the perspective of the meta-controller, a skill is just another type of action available, though what actually happens after acting with a skill is temporally extended behavior. Rewards collected during the skill episode are appropriately summed and discounted over the skill duration and returned to the meta-controller policy.

Since at the beginning of training the meta-controller may be incentivized to only select primitive actions to obtain more fine-grained control, we also introduce a fixed meta-action cost (similar to the "deliberation cost" in Harb et al. (2018)) that is deducted from the reward given to the agent every time the meta-controller selects an action, incentivizing it to use the temporally extended skills where possible. Like the skill-policies, the meta-controller policy is also trained using R2D2, though prioritized experience replay is disabled as priorities are inconsistent to calculate between single-step primitive actions and temporally-extended behaviors. Additionally, we use shorter unroll length (40) and burn-in length (10) than plain R2D2 to account for the use of temporally-extended behaviors.

For both skill HRL variants, we choose the best meta-action cost from (0.0, 0.1, 0.15) and the best skill episode length from (10, 15, 25) for each game. The best values for each Atari level are recorded in Table 2 in the Appendix. In all levels, the number of skills is fixed to 16. Results are averaged over three random seeds.

In Figure 4, we compare the speed of learning over the first 40 million learner steps of an R2D2 agent that acts with either Relative VIC skills and primitive actions, baseline VIC skills and primitive actions, or just primitive actions. The results are mixed across levels although the Relative VIC skills show a learning advantage over the baseline VIC skills and action-baseline R2D2 on most of the games.

To explain the advantage of RVIC skills, we note that in an environment that is sufficiently large or hard to explore during unsupervised training (for example, Montezuma's Revenge which has many rooms that are hard to find by accident), a skill set that partitions the state space will likely restrict skills to partition an incomplete part of the state space, (say, only the first room in Montezuma's Revenge). Using such a set would never allow a meta-controller to explore outside of the already seen subset of the state-space. RVIC can be seen as removing potential overfitting to the distribution of final states seen during the unsupervised pre-training of skills.

Related Work

The idea of maximizing the mutual information between an agent's behavior and the outcome of that behavior can be traced back to Klyubin, Polani, and Nehaniv (2005), where the term 'empowerment' was coined to describe this objective. The empowerment objective was initially limited to small domains due to the costliness of mutual information estimation. Mohamed and Rezende (2015) introduced the idea of maximizing a variational lower bound to scale the empowerment objective to larger domains. Specifically, the Barber-Agakov bound was used (Barber and Agakov 2003), which decomposes the mutual information into a difference of entropies. The negative condition entropy can be lower-bounded by learning a 'reverse predictor'. This can then be combined with knowledge of the true marginal entropy term (e.g. the policy) to provide a lower bound on the mutual information. Gregor, Rezende, and Wierstra (2017) combined variational empowerment with a latent variable for capturing closed-loop temporally extended behavior (i.e. an 'option', or, as we refer to in this paper, a 'skill'). This approach, Variational Intrinsic Control (VIC), parametrized the skill distribution, policy, and skill predictor with neural networks. A skill was sampled and used to condition the policy for some duration. Subsequently, the reverse predictor would predict the sampled skill from the initial and final state. The entropy of the skill distribution and the skill prediction were optimized directly, and reward functions were derived for appropriate credit assignment to the skill distribution and policy.

'Diversity is All You Need' (DIAYN) simplified the VIC algorithm by fixing the option distribution to be the marginal maximum entropy distribution, which most subsequent methods have done as well, including all of those presented here (Eysenbach et al. 2019). While this work also added an action entropy term to the objective, we follow Hansen et al. (2020) in disregarding it, since it is generally less beneficial in discrete-action environments like the Atari suite. Indeed, DIAYN differs significantly from this work in that it only considered environments with explicit state-representations. This both simplifies the perception aspect of the policy learning problem as well as provides a strong inductive bias to the reverse predictor by way of forcing predictability to only arise from these dimensions. Hansen et al. (2020) does show strong results on pixel-based environments, but only when using continuous skill distributions that would significantly increase the burden (by exploding the effective action space) when used for down-stream hierarchical reinforcement learning.

'Discriminative Embedding Reward Networks' (DISCERN) introduced the idea of chaining together sampled skills rather than resetting the environment state between each one (Warde-Farley et al. 2019). This greatly increases the entropy of the initial state distribution, which increases the difficulty of the learning problem in exchange for decoupling skill duration from the final state distribution. For example, if skills reset the environment on termination, then the skill duration would have to be hundreds of steps long to get anywhere in most Atari games. Our method differs from DISCERN in that we are explicitly interested in relative skills as opposed to the achievement of absolute goals represented by desired observations.

Achiam et al. (2018) propose a trajectory-conditional reverse predictor motivated by the idea of learning diverse 'behaviors' rather than diverse goals. Like DIAYN, this method has not been shown to be effective on pixel-based environments and relies on a low-frequency heuristic that would likely be inappropriate for learning skills in Atari games.

Finally, Sharma et al. (2020) maximize a partial lower bound in the same sense that RVIC does. Namely, a difference of entropies decomposition is used even though the marginal entropy is not known a priori and must also be approximated. However, this method is also only shown to work from explicit state-representations and it is non-obvious how to modify it to work from pixels. The empirical stability of both methods suggest that a 'proper' lower bound on a mutual information is not necessary for empowerment based approaches to succeed.

The concept of affordances was first introduced by Gibson (1977) within psychology to refer to action possibilities or opportunities the environment affords an animal at any given time. An affordance emerges from the relationship between an agent and its environment. Gibson suggested that humans are able to easily perceive affordances. Further, Gibson argues that humans actively alter the environment to change what the environment affords us. Within the context of reinforcement learning, Cruz et al. (2014) demonstrate that giving agents affordances as prior knowledge can greatly speed up convergence, and Khetarpal et al. (2020) demonstrate the ability to use affordances to aid planning in RL with partial-models and enable better generalization.

Our work differs from these approaches in its ability to learn affordance-like skills directly from interaction, without any prior knowledge as to what kinds of behavior the environment affords. Indeed, RVIC can be seen as a possible mechanism by which knowledge of affordances can arise. Integrating this mechanism with the rest of the literature on affordances is a promising avenue for future work.

Discussion

Relative VIC learns meaningfully diverse skills that partition the space of affordances by incentivizing the skills to be distinguishable given their first and last states but not distinguishable given only the last state. We analyzed the difference between Relative VIC learned skills and VIC learned skills on several domains and demonstrate the ability to learn affordance-like skills from pixels in both the DeepMind Control Suite and Atari domains. We demonstrate the usefulness of Relative VIC skills in the Hierarchical RL framework on Atari and their ability to generalize across various parts of thei state space.

Some limitations of the method include the use of a fixed discrete uniform skill prior which implies that all skills should exist at every state, even if some options don't make sense at every state. Additionally, the fixed skill duration may prove to be too rigid for some environments to use effectively in HRL. Other potential future direction of this work include learning the meta-controller policy and skill policies simultaneously.

Acknowledgements

We thank Simon Osindero and Doina Precup for reviewing the paper. We also are grateful to Ellen Clancy, Rachel Foley, Catalin Ionescu, Malcolm Reynolds, Stephen Spencer, Melissa Tan, and many others at DeepMind for their continual feedback and support.

Appendix

Number of actors	256
Batch size	64
Optimizer	Adam (2015)
Dense skill reward	True
Skill episode count	10
q_ϕ learning rate	10^{-4}
q_ψ^{abs} learning rate	10^{-4}
π_θ learning rate	10^{-4}
π_θ target update period	10000
q_ϕ, q_ψ^{abs} target update period	10
Actor update period	100
0 γ at skill end (Control Suite)	False
0 γ at skill end (Atari)	True
Skill length (Control Suite)	90
Skill length (Atari)	See Table 2
Number of skills	16
q_ϕ, q_ψ^{abs} shared torso	DQN Conv Torso
q_ϕ, q_ψ^{abs} head hidden size	512

Table 1: A table of hyperparameters for skill learning experiments. Network architecture and hyperparameters for HRL experiments are identical to those in R2D2. Network architecture for the Skill-conditioned Q-network is identical to the recurrent, duelling Q-network used in R2D2, with the addition of a two layer (sizes 256, 512) MLP skill torso whose output is concatenated with the output of the convolutional observation torso before being input to the recurrent core.

	Baseline VIC		**Relative VIC**	
	Meta action cost	Skill episode length	Meta action cost	Skill episode length
Amidar	0.15	25	0.15	10
Enduro	0.00	10	0.00	10
Frostbite	0.15	10	0.00	10
M. Revenge	0.00	25	0.10	10
Private Eye	0.00	10	0.15	25
Seaquest	0.15	10	0.00	10

Table 2: A table of the best hyperparameters for each level of Atari in the HRL experiments. The hyperparameter combination with the best HRL performance when averaged over 3 random seeds was chosen. Results displayed in Figure 4.

References

Achiam, J.; Edwards, H.; Amodei, D.; and Abbeel, P. 2018. Variational Option Discovery Algorithms. *CoRR* abs/1807.10299.

Barber, D.; and Agakov, F. 2003. The IM algorithm: a variational approach to Information Maximization. In *NIPS*, 201–208.

Barber, D.; and Agakov, F. V. 2004. Information Maximization in Noisy Channels : A Variational Approach. In Thrun, S.; Saul, L. K.; and Schölkopf, B., eds., *NIPS*, 201–208. MIT Press.

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research* 47: 253–279.

Cobbe, K.; Klimov, O.; Hesse, C.; Kim, T.; and Schulman, J. 2019. Quantifying Generalization in Reinforcement Learning. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, 1282–1289. PMLR.

Cruz, F.; Magg, S.; Weber, C.; and Wermter, S. 2014. Improving reinforcement learning with interactive feedback and affordances. In *ICDL-EPIROB*, 165–170. IEEE.

Eysenbach, B.; Gupta, A.; Ibarz, J.; and Levine, S. 2019. Diversity is All You Need: Learning Skills without a Reward Function. In *ICLR (Poster)*.

Gibson, J. J. 1977. The theory of affordances .

Gregor, K.; Rezende, D. J.; and Wierstra, D. 2017. Variational Intrinsic Control. In *ICLR (Workshop)*.

Hansen, S.; Dabney, W.; Barreto, A.; Warde-Farley, D.; de Wiele, T. V.; and Mnih, V. 2020. Fast Task Inference with Variational Intrinsic Successor Features. In *ICLR*.

Harb, J.; Bacon, P.; Klissarov, M.; and Precup, D. 2018. When Waiting Is Not an Option: Learning Options With a Deliberation Cost. In *AAAI*, 3165–3172. AAAI Press.

Kapturowski, S.; Ostrovski, G.; Quan, J.; Munos, R.; and Dabney, W. 2019. Recurrent Experience Replay in Distributed Reinforcement Learning. In *ICLR (Poster)*.

Khetarpal, K.; Ahmed, Z.; Comanici, G.; Abel, D.; and Precup, D. 2020. What can I do here? A Theory of Affordances in Reinforcement Learning. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, 5243–5253. PMLR.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

Klyubin, A. S.; Polani, D.; and Nehaniv, C. L. 2005. Empowerment: a universal agent-centric measure of control. In *Congress on Evolutionary Computation*, 128–135. IEEE.

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. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540): 529–533.

Mohamed, S.; and Rezende, D. J. 2015. Variational information maximisation for intrinsically motivated reinforcement learning. In *NIPS*, 2125–2133.

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

Sharma, A.; Gu, S.; Levine, S.; Kumar, V.; and Hausman, K. 2020. Dynamics-Aware Unsupervised Discovery of Skills. In *ICLR*.

Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; Lillicrap, T. P.; Simonyan, K.; and Hassabis, D. 2017a. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *CoRR* abs/1712.01815.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; Chen, Y.; Lillicrap, T. P.; Hui, F.; Sifre, L.; van den Driessche, G.; Graepel, T.; and Hassabis, D. 2017b. Mastering the game of Go without human knowledge. *Nature* 550(7676): 354–359.

Sutton, R. S.; and Barto, A. G. 1998. *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st edition. ISBN 0262193981.

Sutton, R. S.; Precup, D.; and Singh, S. P. 1999. Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial Intelligence* 112(1-2): 181–211.

Tassa, Y.; Doron, Y.; Muldal, A.; Erez, T.; Li, Y.; de Las Casas, D.; Budden, D.; Abdolmaleki, A.; Merel, J.; Lefrancq, A.; Lillicrap, T. P.; and Riedmiller, M. A. 2018. DeepMind Control Suite. *CoRR* abs/1801.00690.

Warde-Farley, D.; de Wiele, T. V.; Kulkarni, T. D.; Ionescu, C.; Hansen, S.; and Mnih, V. 2019. Unsupervised Control Through Non-Parametric Discriminative Rewards. In *ICLR (Poster)*.

Whiteson, S.; Tanner, B.; Taylor, M. E.; and Stone, P. 2011. Protecting against evaluation overfitting in empirical reinforcement learning. In *ADPRL*, 120–127. IEEE.