Empowered Skills

Alexander Gabriel, Riad Akrour, Jan Peters and Gerhard Neumann

***Abstract*— Robot Reinforcement Learning (RL) algorithms return a policy that maximizes a global cumulative reward signal but typically do not create diverse behaviors. Hence, the policy will typically only capture a single solution of a task. However, many motor tasks have a large variety of solutions and the knowledge about these solutions can have several advantages. For example, in an adversarial setting such as robot table tennis, the lack of diversity renders the behavior predictable and hence easy to counter for the opponent. In an interactive setting such as learning from human feedback, an emphasis on diversity gives the human more opportunity for guiding the robot and to avoid the latter to be stuck in local optima of the task. In order to increase diversity of the learned behaviors, we leverage prior work on intrinsic motivation and empowerment. We derive a new intrinsic motivation signal by enriching the description of a task with an outcome space, representing interesting aspects of a sensorimotor stream. For example, in table tennis, the outcome space could be given by the return position and return ball speed. The intrinsic motivation is now given by the diversity of future outcomes, a concept also known as empowerment. We derive a new policy search algorithm that maximizes a trade-off between the extrinsic reward and this intrinsic motivation criterion. Experiments on a planar reaching task and simulated robot table tennis demonstrate that our algorithm can learn a diverse set of behaviors within the area of interest of the tasks.**

I. Introduction

The application of Reinforcement Learning (RL) to robotics is widespread and has several success stories [1], [2], [3]. A RL problem is defined by sets of states and actions as well as a reward function that maps a state-action pair to a score. The goal of RL algorithms is to learn a policy which maximizes the global cumulative reward. The policy can either be deterministic or stochastic but even in the latter case, stochasticity is only introduced for the sake of exploration [4], [5] and is typically reduced over time, yielding as a result a policy exhibiting a relative small amount of diversity.

However, diversity can be an advantage in many settings. In dynamic environments, a diverse policy is often easier to adapt to a changing environment or changing task constraints [6]. Moreover, diversity can help to avoid getting stuck in local optima. For example, in the cooperative setting of [7], a robot learns grocery checkout by interacting with a human. The human can add constraints such as not moving liquids over electronics by reranking the trajectories of the robot. Unfortunately, the robot can get stuck in local optima requiring manual modification of the trajectory's waypoints [7]. Increasing the diversity in the robot's behavior can give more feedback opportunities to the human and avoids manual interventions. Another example where diverse behaviors should be preferred are competitive settings. Here, an agent with a repetitive strategy becomes predictable and thus more easily exploitable. In robot table tennis, for example, a task can be to return incoming balls to the opponent's side of the table in a way that is hard to return for the opponent. A deterministic behavior would preclude any effect of surprise and reduce the chances of defeating the opponent. Therefore, we are interested in allowing the robot to constitute a diverse library of high reward motor skills. In order to increase the diversity of the learned skills, we use intrinsic motivation [8] and empowerment [9]. We derive a new intrinsic motivation signal by enriching the description of a task with an outcome space. The outcome space represents interesting aspects of a sensorimotor stream, for example, in table tennis, the outcome space could be given by the return position of the ball or the return ball speed. Our intrinsic motivation signal is now given by the diversity or entropy of the outcomes of the policy. Maximizing the entropy of the future is a concept also known as empowerment [9], therefore, we will call our approach *Empowered Skills*. Our new algorithm maximizes a trade-off between an extrinsic reward, e.g., returning the ball on the table in table tennis, and this intrinsic motivation criterion. Experiments on a planar reaching task and a simulated robot table tennis task demonstrate that our algorithm can learn a diverse set of behaviors within the area of interest of the tasks.

A. Related Work

Intrinsic motivation can be categorized into three clusters [8]: knowledge-based models, competence-based models and morphological models.

1) Knowledge-Based Models of Intrinsic Motivation: Agents of this category typically monitor and maximize a learning progress. [10], [11] monitor the learning progress of the transition model (predicting the next state given the current state and action). Its maximization leads to the sequencing of the learning process from the simplest to the most intricate parts of the state-action space, while ignoring the parts that are already learned or impossible to learn. Such an intrinsic motivation signal is also used in [12] and is mixed with an extrinsic, goal-oriented reward. However, unlike in our algorithm, the increased exploration due to the intrinsic motivation serves only the purpose of speeding-up the learning process. Moreover, its influence decreases with time and the returned policy is similar to standard RL algorithms.

2) Competence-Based Models of Intrinsic Motivation: Agents of this category set challenges or goals for themselves which consist of a specific sensorimotor configuration with

associated difficulty level. They then plan their actions to achieve this goal and get a reward based on the difficulty of the challenge and their actual performance. [8] suggests three possible performance measures based on incompetence, competence and competence progress. Since then, [13] has examined the use of a competence-based model to support an agent's autonomous decision on what skills to learn. They measure the improvement rate of competence on the basis of the Temporal-Difference learning signal. [14] enhanced the framework introduced in [10] with a competence-based intrinsic motivation system. Here, learning is directed by a measure of interestingness which characterizes goals in task space. [15] applies this framework in the context of tool use discovery. They split a high-dimensional structured sensorimotor space into subspaces with an associated measure of interestingness. A multi armed bandit algorithm then selects in which subspace to perform goal babbling [16] based on the subspaces' interestingness.

3) Morphological Models of Intrinsic Motivation: This category comprises algorithms that maximize mathematical properties of the sensorimotor space of the robot and as such is closely related to the diversity in behavioral space we are striving for. In Evolutionary Robotics, [17] designed intrinsic online fitness functions that encourage a population of robots to seek out experiences that neither the current nor previous generations of robots had before. Thus, the algorithm grows a population of curious explorers that are motivated not by reward but by diverse actions and experiences alone. [17] and [18] evolved neural network controllers where the typical goal-oriented fitness function is replaced by an intrinsically motivated term. Specifically, [18] introduced a distance metric in behavioral space and aimed to find behaviors that are maximally different to all the previously experienced behaviors. However, enumerating all such behaviors might not be sustainable in high dimensional spaces. The authors in [17] propose to search for a behavior exhibiting maximal entropy in its sensorimotor stream. For some settings, such behaviors can result in interesting solutions to a task such as navigating through a maze. Yet, high entropy behaviors do not necessarily coincide with high reward behaviors and as such a trade-off between the intrinsic and extrinsic reward is necessary.

Empowerment: The notion of empowerment introduced in [19] and extended to the continuous case in [9] is defined on the state space. The empowerment of a state is proportional to the number of distinct states that could be reached from it. It was demonstrated in [9] that for a task such as pole balancing, the optimal swing up policy coincides with the search of the most empowered state as the upright position has the highest diversity of future states. However, similar to the discussion of [17], such behaviors do not always emerge for other tasks. As such, in our algorithm, the intrinsic motivation is not used on its own but is combined with the extrinsic reward of the task such that we can restrict the diverse solution space to a subset of useful solutions.

Similar to [17], we measure diversity in our algorithm with an entropy function. However, as the sensorimotor stream can be excessively large, we restrict the domain of the entropy to the outcome space (Sec. II-A) that only captures task-relevant aspects of the stream. We call our algorithm *Empowered Skills* and distinguish it from the notion of empowerment of states [19], [9] as we search for (motor) *skills* that result in diverse *outcomes*.

II. Empowered Skills

The Empowered Skills algorithm is couched in the Direct Policy Search (DPS) framework [4] and therefore reuses most of its terminology (Sec. II-A). In the following sections, we will describe our algorithm first for discrete and then for continuous outcome spaces.

A. Notation and Problem Statement

Given a reward function $\mathcal{R} : \mathrm{T} \mapsto \mathbb{R}$ mapping a robot trajectory $\tau \in \mathrm{T}$ to a real value $\mathcal{R}(\tau)$, the goal of DPS algorithms is to find a set of parameters $\theta \in \Theta$ maximizing $\mathbb{E}_{p_\theta(\tau)}[\mathcal{R}(\tau)]$ that we denote with a slight abuse of notation $\mathcal{R}(\boldsymbol{\theta})$. Specifically, DPS is a class of iterative algorithms maintaining search distributions over Θ that we refer to as policies[1]. The main objective of a DPS algorithm can be formulated as maximizing the policy return $J(\pi) = \mathbb{E}_{\boldsymbol{\theta}\sim\pi}[\mathcal{R}(\boldsymbol{\theta})]$.

In addition to the maximization of the reward, we introduce an intrinsic motivation term to the objective function of our algorithm with the goal to enforce diverse solutions. Similar to other intrinsically motivated algorithms [12], [17], [9], [20], diversity is measured by an entropy term. However, instead of computing the entropy over the whole sensorimotor stream $\mathcal{X}$ (like in [17]) which is typically very high dimensional, we provide additional guidance to the algorithm by singling out relevant parts of the sensorimotor stream which we call *outcomes*.

Formally, the outcome space $\mathcal{O}$ is defined as the image of a non-injective mapping $f : \mathcal{X} \mapsto \mathcal{O}$ such that typically $\mathrm{card}(\mathcal{O}) \ll \mathrm{card}(\mathcal{X})$. For instance, a trajectory τ, that is part of $\mathcal{X}$ and comprised of all joint positions of the robot as well as the positions of the ball during the execution of a robot table tennis strike could retain as outcome $\boldsymbol{o} = f(\tau)$ only the 2D ball position when it hits the table after returning the ball.

Upon the definition of the outcome space, the intrinsic term is simply given[2] by the entropy $\mathcal{H}_\mathcal{O}(\pi) = -\sum_{\boldsymbol{o}} p_\pi(\boldsymbol{o}) \ln p_\pi(\boldsymbol{o})$ of the outcome probabilities $p_\pi(\boldsymbol{o}) = \int p(\boldsymbol{o}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}$. Finally, the policy π^* returned by our algorithm is optimal w.r.t. a trade-off between the extrinsic reward and intrinsic motivation, i.e.,

$$\pi^* = \arg\max_\pi J(\pi) + \beta\mathcal{H}_\mathcal{O}(\pi),$$

[1]For simplicity of notation, only policies of the form $\pi(\boldsymbol{\theta})$ are considered throughout the paper. An extension to the contextual case $\pi(\boldsymbol{\theta}|\boldsymbol{c})$ for some initial i.i.d. task-dependent contexts $\boldsymbol{c} \in C$ is straightforward and similar to that of previous DPS algorithms (see chapter 2.4.3.2 in [4])

[2]$\mathcal{H}_\mathcal{O}(\pi)$ is given for a discrete outcome space; extension to continuous outcomes follows in Section II-C

with trade-off parameter β. Higher values for β will put more emphasis on the entropy and result in policies exhibiting higher diversity.

B. Finite Outcome Space

Inspired by information-theoretic policy search algorithms [21], [22], we solve this optimization problem in an iterative scheme where at each iteration the new policy is updated by solving the following constrained optimization problem

$$\begin{aligned}
\arg\max_{\pi,\hat{p}_\pi} \quad & J(\pi) + \beta\mathcal{H}_{\mathcal{O}}(\pi), \\
\text{s.t.} \quad & \epsilon > \mathrm{KL}(\pi||q), && (1)\\
& 1 = \int \pi(\boldsymbol{\theta})\, d\boldsymbol{\theta}, && (2)\\
\forall \boldsymbol{o}: \quad & \hat{p}_\pi(\boldsymbol{o}) = \int p(\boldsymbol{o}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\, d\boldsymbol{\theta}, && (3)\\
& 1 = \sum_{\boldsymbol{o}} \hat{p}_\pi(\boldsymbol{o}). && (4)
\end{aligned}$$

The Kullback-Leibler Divergence $\mathrm{KL}(\pi||q)$ is used to specify the step-size [21], [22] of the policy update and is given by

$$\mathrm{KL}(\pi||q) = \int \pi(\boldsymbol{\theta}) \log\left(\frac{\pi(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}\right) d\boldsymbol{\theta}.$$

Without the outcome entropy term $\mathcal{H}_{\mathcal{O}}(\pi)$ and conditions 3 and 4, this optimization problem would be a standard Policy Search problem where the information loss, given by the Kullback-Leibler Divergence between the last and current policy, is bounded by ϵ [21], [22]. The use of KL-constraints is widespread in the robotic RL community whether it is in the context of Policy Search [21], Policy Gradient Methods [23] or Optimal Control [20].

The addition of the outcome entropy term $\mathcal{H}_{\mathcal{O}}(\pi)$ introduces an interdependency between $p_\pi(o)$ and π. This interdependency prohibits the derivation of a closed form policy update directly from the constrained optimization problem[3]. A set of auxiliary variables $\hat{p}_\pi(\boldsymbol{o})$ that we optimize for is therefore introduced to break this dependency yielding $\mathcal{H}_{\mathcal{O}}(\pi) = -\sum_o \hat{p}_\pi(\boldsymbol{o}) \ln \hat{p}_\pi(\boldsymbol{o})$. For a finite (and relatively small) outcome space, constraints 3 and 4 then allow us to enforce for all $\boldsymbol{o} \in \mathcal{O}$ the equality[4] $\hat{p}_\pi(\boldsymbol{o}) = \int p(\boldsymbol{o}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\, d\boldsymbol{\theta}$, resulting in the closed form policy update

$$\pi(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta}) \exp\left(\frac{1}{\eta}\left(\mathcal{R}(\boldsymbol{\theta}) + \sum_{\boldsymbol{o}} \mu_o\, p(\boldsymbol{o}|\boldsymbol{\theta})\right)\right),$$

where μ_o are the Lagrangian multipliers for the constraints given in Equation 3. In comparison to the standard solution for DPS, see the episodic REPS algorithm given in [22], we can identify the term $\sum_{\boldsymbol{o}} \mu_o\, p(\boldsymbol{o}|\boldsymbol{\theta})$, which is added to the reward function $\mathcal{R}(\boldsymbol{\theta})$ as intrinsic motivation. This term depends on the Lagrangian multipliers μ_o that are obtained by optimizing the dual function of the optimization problem.

[3]The interdependency results in a log-sum expression in the Lagrangian that cannot be solved in closed form.

[4]Within the internal optimization routine of a policy update, a sample estimates of the r.h.s. $\int p(\boldsymbol{o}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\, d\boldsymbol{\theta}$ of each equality constraint needs to be computed for different π. This can be done solely from data generated in previous iterations using importance sampling. However, we do not elaborate further as these constraints will not appear in the continuous outcome case.

C. Continuous Outcome Space

In the continuous case, adding a constraint for every possible outcome is impossible. As a consequence, we need to resort to matching feature expectations instead of single probability values. Hence, we replace the constraints 3 and 4 with

$$\int \boldsymbol{\phi}(\boldsymbol{o})\, \hat{p}_\pi(\boldsymbol{o})\, d\boldsymbol{o} = \int \boldsymbol{\phi}(\boldsymbol{o}) \int p(\boldsymbol{o}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\, d\boldsymbol{\theta} d\boldsymbol{o}, \quad (5)$$

$$1 = \int \hat{p}_\pi(\boldsymbol{o})\, d\boldsymbol{o}. \quad (6)$$

The expression of $\mathcal{H}_{\mathcal{O}}(\pi)$ in the continuous case is simply obtained by replacing the sum over the domain $\mathcal{O}$ by an integral.

D. Solving the Optimization Problem

The policy update can now again be obtained by Lagrangian optimization and is given by

$$\pi(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta}) \exp\left(\frac{1}{\eta}\delta_{\boldsymbol{\mu}}(\boldsymbol{\theta})\right),$$

where $\delta_{\boldsymbol{\mu}}(\boldsymbol{\theta}) = \mathcal{R}(\boldsymbol{\theta}) + \boldsymbol{\mu} \int p(\boldsymbol{o}|\boldsymbol{\theta})\boldsymbol{\phi}(\boldsymbol{o})\, d\boldsymbol{o}$ and $\boldsymbol{\mu}$ is a vector of Lagrangian multipliers for the constraint given in Equation 5. The policy depends on the Lagrangian multipliers η and $\boldsymbol{\mu}$ which can be determined by minimizing the dual function

$$\begin{aligned}
g(\eta, \boldsymbol{\mu}) = \eta\epsilon + \eta \log\left(\int q(\boldsymbol{\theta}) \exp\left(\frac{1}{\eta}\delta_{\boldsymbol{\mu}}(\boldsymbol{\theta})\right) d\boldsymbol{\theta}\right) \\
+ \beta \log\left(\int \exp\left(\frac{-\boldsymbol{\mu}\boldsymbol{\phi}(\boldsymbol{o})}{\beta}\right) d\boldsymbol{o}\right). \quad (7)
\end{aligned}$$

Similar to other DPS algorithms such as episodic REPS [22], we can only solve this optimization problem for a finite set of samples. The result of this optimization is then given by a weighting $w_i = \exp\left(\frac{1}{\eta}(r_i + \boldsymbol{\mu}\boldsymbol{\phi}(\boldsymbol{o}_i))\right)$ for each sample. These weights are subsequently used to estimate a new parametric policy using a Weighted Maximum Likelihood (WML) estimate [22].

E. Estimating the New Policy

Because of their simplicity, we use multi-variate Gaussian distributions for our policies which is also a common assumption in DPS. The mean and covariance matrix of the new policy π are given by

$$\begin{aligned}
\boldsymbol{\mu}_\pi &= \frac{\sum w_i \boldsymbol{\theta}_i}{\sum w_i}, \\
\boldsymbol{\Sigma}_\pi &= \frac{\sum w_i (\boldsymbol{\theta}_i - \boldsymbol{\mu}_\pi)(\boldsymbol{\theta}_i - \boldsymbol{\mu}_\pi)^T}{Z}, \\
Z &= \frac{(\sum w_i)^2 - \sum w_i^2}{\sum w_i}
\end{aligned}$$

which is the WML estimator for samples θ_i drawn from the previous policy q.

III. Experiments

The objectives of our experiments are twofold. Firstly we investigate the ability of our algorithm to learn policies showing high diversity in the outcome space. Secondly we investigate how we can shape the learned policies by choosing different characteristics as outcomes.

We assessed the abilities and performance of our algorithm in two scenarios: a 2D reaching task involving a 5 DoF robot arm and the table tennis task featuring a 9 DoF robot arm as shown in Figure 4. In all experiments the policy learned by the algorithm is represented by a multivariate Gaussian distribution over parameters of the Dynamic Movement Primitives (DMP) that control the motion of the respective robot arm. As the outcome space is continuous in all of the experiments we use Radial Basis Functions (RBF) to approximate the distribution of outcomes $p_\pi(\boldsymbol{o})$. As basis we use 20 randomly chosen outcomes. For each basis, we weight the distance to each outcome by the squared median of distances to this basis. We chose the median to have an outlier resistant scaling factor for every basis.

A. Reaching Task

In the reaching task the robot's objective is to move the end effector from the top to a target position on the right. It performs this task in 100 time steps. The parameter space is 30 dimensional and consists of the weights of the DMP (5 basis per joint and the target location in joint space). The reward $\mathcal{R}$ depends on the action cost $\boldsymbol{u}$ and the distance d between end effector and target in the last time step i.e.

$$\mathcal{R}(d, \boldsymbol{u}) = -10^5 d^2 - 0.7 \boldsymbol{u}^T \boldsymbol{u}.$$

In this experiment scenario we define the positions of the middle joint at the end of the trajectory as the outcome. This results in a two-dimensional, continuous outcome space.

To learn the policy, the algorithms runs 75 iterations of policy updates. In every iteration they each draw 50 sample trajectories from their current policy.

Comparison to State-of-the-Art RL Algorithms

The first set of experiments pits our algorithm against state-of-the-art RL algorithms.

Figure 1 shows how outcome entropy, policy entropy and reward evolve over the 75 iterations of the algorithms. First we note that for all the runs, the reward stabilizes from iteration 30 onwards while the policy entropy continues to shrink. The outcome entropy converges more slowly than the reward. This is even more apparent for our algorithm, where the outcome entropy declines more slowly than the policy entropy. MORE [24] introduces an entropy constraint on the search distribution regulating the reduction of the parameter entropy at each iteration. The purpose of this constraint is to have a better control over the exploration trading-off slower initial progress as apparent in Figure 1 for better reward at convergence as can be seen in Table I. The effect of the entropy constraint of MORE results in a slower and steadier decrease of the policy entropy. However, the higher entropy in parameter space does not translate into entropy in outcome space where our algorithm clearly outperforms the others.

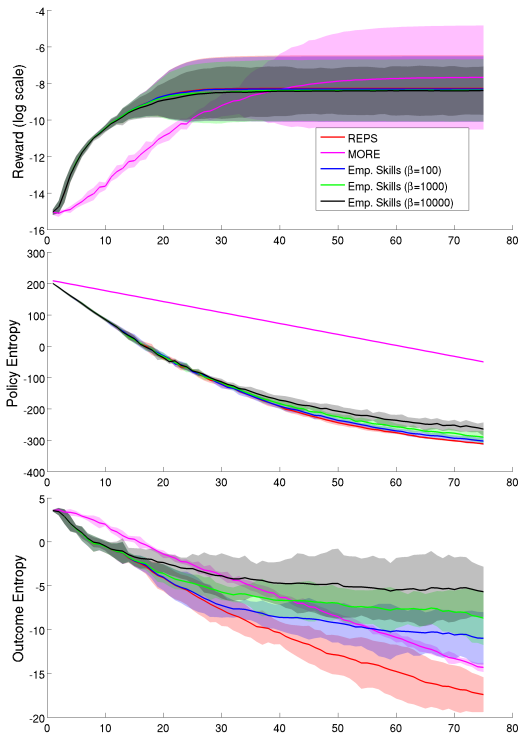

Fig. 1. Evolution of outcome entropy, policy entropy and reward over 75 iterations in the reaching task experiments. The figure shows that our algorithm manages to find policies with higher outcome entropy at the expense of a slight reward decrease.

β	Policy Entropy	Outcome Entropy	Avg. Reward
100	-302.47 ± 6.28	-10.980 ± 1.496	-4032 ± 3329
1000	-289.72 ± 10.04	-8.675 ± 1.533	-4316 ± 3363
10000	-264.36 ± 10.87	-5.677 ± 1.429	-4406 ± 2480
REPS	-311.48 ± 1.44	-17.394 ± 0.989	-3900 ± 3270
MORE	-49.56 ± 0.00	-14.291 ± 0.262	-2144 ± 1946

TABLE I
Results of the Reaching Task Experiments.

The table shows the results of our reaching task experiments. The experiments were run with 5 trials of 75 iterations per setting. We include REPS [21] and MORE [24] results as baseline comparison. Of note are the similar policy entropy and average reward, the high standard deviation on the reward as well as the distinct differences in outcome entropy which show that our algorithm is able to find more diverse solutions.

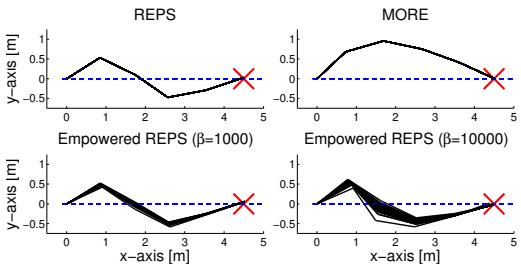

Fig. 2. Final arm configuration of 50 trajectories sampled for different algorithms and values of β. Policies returned by standard RL algorithms exhibit no diversity (top row). For our algorithm (bottom row), increasing β increases the behavioral diversity while only slightly decreasing the reward.

The returned policy of each algorithm can be seen in Figure 2 which shows for different settings of β the robot arm configurations in the last time step as well as the target point, indicated by the red cross. For this we plotted the arm configuration for all the 50 sample trajectories on top of each other. The results make it clear that the solution found by our algorithm is similar to that found by REPS which it is based on. But where REPS finds a single solution, our algorithm exploits the robot's structure to locally increase diversity in the target joint. Choosing higher values for β leads to increased diversity in posture.

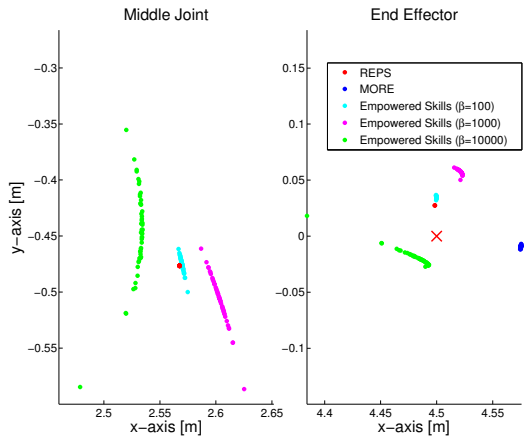

Fig. 3. Middle Joint position (left, outcome) and End Effector (right, reward) of 50 trajectories sampled for different algorithms and values of β illustrating the trade-off between behavioral diversity and reward.

This can be confirmed with a look at Figure 3 which displays the end effector and the middle joint in the last time step of the last iteration. The position distributions seem to be arc shaped in the runs of our algorithm. This suggests that the algorithm exploits the configuration of the robot to locally increase diversity. The arcs are evidence of variation in few, specific joints. Variation in many joints dissolves the arc-shape-constraint imposed by a single link rotating around a single joint.

We measured a sizable increase in the diversity of the outcomes. For REPS the middle joint positions cover an area smaller than $0.01mm^2$. For Empowered Skills ($\beta = 100$) this area is more than 220 times larger[5].

The statistical results of the experiment runs can be seen in Table I. While the policy entropy for our algorithms is only marginally increased in comparison to REPS and a lot lower than in the MORE results, the outcome entropy is distinctly higher.

B. Robot Table Tennis

In the table tennis setting there is a simulated robot arm with a racket attached to its end effector. The robot's objective is to return a ball to the opponent's side of the table. The robot arm has six joints and is suspended over the table from a floating base which itself has 3 linear joints

[5]The size of the markers exaggerates the area covered by REPS due to visibility constraints.

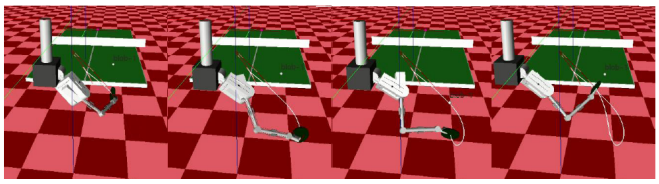

Fig. 4. Images of the robot table tennis performing a forehand strike using a policy learned by our algorithm.

to allow small 3D movement. A trajectory is parameterized by the goal position and velocities of a DMP, yielding an 18 dimensional continuous action space (9 positions and 9 velocities).

In this setting we conduct several experiments in which we choose different characteristics as outcomes. In the first experiment we use the location of the ball when landing on the table upon being returned by the robot. This gives us a two dimensional continuous outcome space. In the second experiment we use the speed of the ball at the moment of impact instead, that gives a one dimensional outcome space. Because in these sets of experiments the outcomes are not properties of the robot, the relation between policy and outcomes depends on the environment (in this case the physics of ball movement).

We use imitation learning to initialize the weight parameters of the DMP and optimize for the goal position and velocity. The initial trajectory used for imitation learning was generated using a hand coded player following [25]. Once the policy is initialized, we run 100 iterations of each of the RL algorithms. The reward optimized by these algorithms is inversely proportional to the distance between a target point[6] and the location where the ball first enters the table plane after being returned by the robot. If the robot doesn't hit the ball, the reward is the negative minimum distance between the racket and the ball throughout the trajectory. In all the experiments we ran 5 trials for each algorithms for 100 iterations using 50 samples per iteration.

Experiment 1 - Comparison to a State-of-the-Art RL Algorithm

The first set of experiments again pits our algorithm against a state-of-the-art RL algorithm and explores the

[6]The target is marked by a red cross in our plots.

β	Policy Entropy	Outcome Entropy	Avg. Reward
10	-217.805 ± 1.463	-3.705 ± 0.891	17.694 ± 0.190
100	-216.498 ± 2.951	-2.898 ± 0.616	17.412 ± 0.503
1000	-214.432 ± 2.184	-1.654 ± 0.830	10.536 ± 5.394
REPS	-216.863 ± 1.591	-8.119 ± 1.765	18.124 ± 0.017

TABLE II
RESULTS OF THE TABLE TENNIS EXPERIMENTS.

These experiments were run in 5 trials with 100 iterations and 50 samples per iteration. Displayed are mean values over the 5 trials. The results show how higher settings of β lead to an increased outcome entropy while having a limited effect on policy entropy and reward except for $beta = 1000$. In this experiment the outcomes are the 2D location where the ball impacts the table after being returned by the robot.

influence of an increased β on the shape of the solutions. The outcomes in this case are the position of the ball in the table plane at the moment the ball enters this plane.

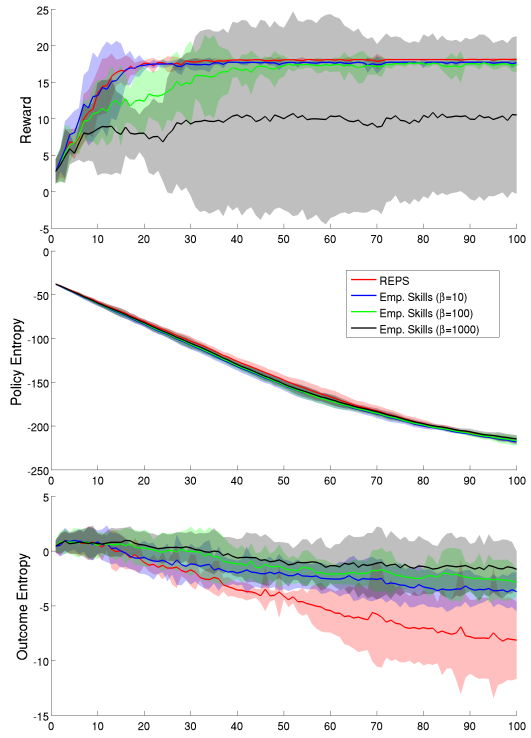

Fig. 5. Evolution of outcome entropy, policy entropy and reward over 100 iterations of the table tennis experiment. The figure shows the similarities in the policy entropy and reward while the outcome (bounce locations) curves are more distinct. For $\beta = 1000$, reward decreases due to balls hitting the net.

Figure 5 shows how outcome entropy, policy entropy and reward develop over 100 iterations of running the algorithms. The biggest differences are visible in the outcome entropy. While the policy entropy is comparable for all the runs over all iterations, the outcome entropy clearly decreases when a smaller β is chosen. The reward converges slower for higher β but does reach a similar score for all but the run with the highest β. Our algorithms seems to be able to keep outcome entropy and reward relatively high while decreasing the policy entropy. The results for the last iteration are shown again in Table II.

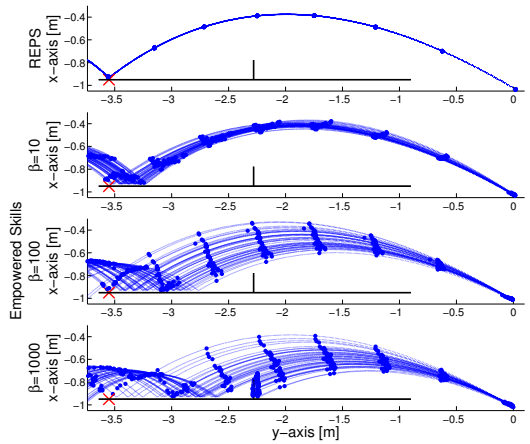

Fig. 6. Sample trajectories from REPS and Empowered Skills for different values of β. REPS although learning a probabilistic policy always shoots to the target. Our algorithm is able to simultaneously learn to shoot to different areas of the table.

The final ball trajectories can be seen in Figure 6. The figure shows 50 ball trajectories for each of the algorithms. While the trajectories for REPS are indistinguishable and those with $\beta = 10$ are still very similar, for higher values of β the trajectories cover a much wider area. The figure makes it clear that the Empowered Skills algorithm is able to learn a policy that represents multiple solutions to the posed problem while the policy learned by REPS just yields a single behavior.

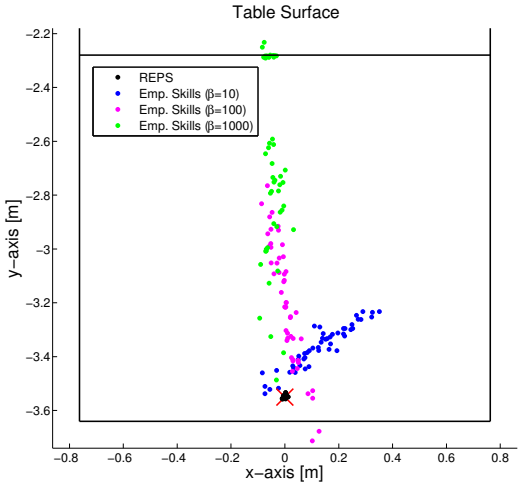

Fig. 7. A top view of the table. Marked on it are the impact points of the ball for 50 sample trajectories per algorithm. Higher values of β lead to a larger spread of outcomes. The mean of the outcomes is different as well, and gets further away from the table's edge.

Figure 7 displays the outcomes of 50 sample trajectories drawn from the final policy for different settings of β. We can see that the impact area grows with β. For the second highest choice of β two shots miss the table and for the highest β some of the shots end in the net. Interestingly with increasing outcome entropy the center of outcomes moves to the middle of the opponent's side of the table. Had it stayed in the same place (the target point marked by the red cross) approximately half of the outcomes would have missed the table and with it the objective. This shows that our algorithm finds a compromise between getting a high reward and producing outcomes with higher entropy.

Experiment 2 - The Speed Test

The second set of experiments changes the outcomes to the speed at which the ball impacts the table. This leads our algorithm to learn a different set of policies.

Figure 8 shows the distribution of outcomes for REPS and our algorithm at different settings of β. The diversity of impact speeds increases with growing β. Again the algorithm learns policies with different mean, shifting away from the

table border with increasing β. This shift is a lot less pronounced than in the experiments with position outcomes in Section III-B.

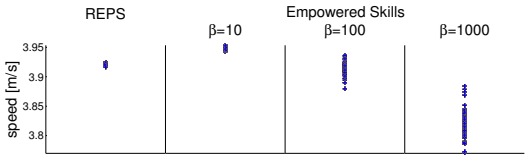

Fig. 8. Speed outcomes of 50 trajectories sampled for different values of β and REPS. Here the outcome is the balls impact speed on the table. The figure shows how increasing β leads to more diverse impact speeds that are distributed around different means.

With a look at Figure 9 it becomes also clear that the diversity of impact locations is much lower than in the previous experiment. We can thus choose which characteristics are important to us and learn fitting policies by choosing appropriate outcomes.

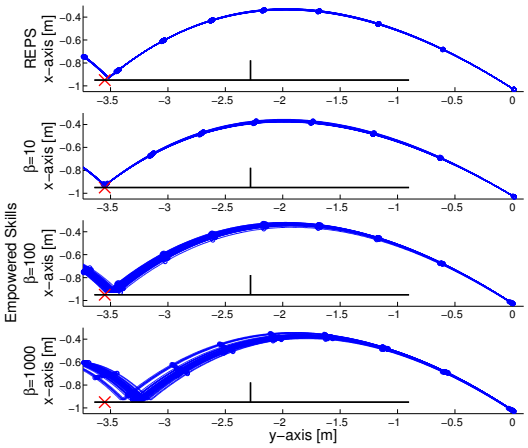

Fig. 9. Speed outcomes of 50 trajectories sampled for different values of β and REPS. The solutions found by our algorithm for the speed experiment show a distinctly reduced spread on the table compared to those for the position experiment (Sec. III-B).

IV. Discussion

In this paper we added an outcome entropy based intrinsic motivation term to a state-of-the-art objective driven Policy Search algorithm. Our new algorithm is capable of solving problems with continuous action and outcome spaces and is easily extendable to the contextual case. We tested the algorithm in two experimental settings, a reaching task and a robot table tennis task. The experiments show how the Empowered Skills algorithm proposed in this paper is able to learn policies which exhibit higher behavioral diversity in high reward areas of the task. The main limitation of our current setting is the simple form of our Gaussian policy. In order to further increase outcome diversity, our future perspective is to study the integration of this intrinsic motivation term in more complex distributions such as mixture models. Another perspective is to learn a higher level policy to select the most appropriate skill in a cooperative or adversarial setting such as robot table tennis, from a library of skills discovered by intrinsic motivation.

References

[1] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," IJRR, vol. 32, no. 11, pp. 1238–1274, 2013.

[2] S. Levine, N. Wagener, and P. Abbeel, "Learning Contact-Rich Manipulation Skills with Guided Policy Search," in IEEE ICRA, 2015, pp. 156–163.

[3] P. Kormushev, S. Calinon, and D. G. Caldwell, "Robot motor skill coordination with EM-based reinforcement learning," in IEEE/RSJ IROS, no. September 2016, 2010, pp. 3232–3237.

[4] M. P. Deisenroth, G. Neumann, and J. Peters, "A Survey on Policy Search for Robotics," Foundations and Trends in Robotics, vol. 2, no. 2011, pp. 1–142, 2011.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," CoRR, 2013.

[6] C. Daniel, G. Neumann, O. Kroemer, and J. Peters, "Hierarchical Relative Entropy Policy Search," JMLR, vol. 17, pp. 1–50, 2016.

[7] A. Jain, B. Wojcik, T. Joachims, and A. Saxena, "Learning Trajectory Preferences for Manipulators via Iterative Improvement," in NIPS, 2013, pp. 575–583.

[8] P. Y. Oudeyer and F. Kaplan, "What is intrinsic motivation? A typology of computational approaches," Frontiers in Neurorobotics, vol. 1, no. Nov, p. 6, 2009.

[9] T. Jung, D. Polani, and P. Stone, "Empowerment for Continuous Agent–Environment Systems," Adaptive Behavior, vol. 19, pp. 16–39, 2011.

[10] A. Baranes and P.-Y. Oudeyer, "R-IAC: Robust Intrinsically Motivated Exploration and Active Learning," IEEE TAMD, vol. 1, no. 3, pp. 155–169, 2009.

[11] J. Schmidhuber, "A Possibility for Implementing Curiosity and Boredom in Model-Building Neural Controllers," in SAB, vol. 1, 1991, pp. 222–227.

[12] M. Lopes, T. Lang, M. Toussaint, and P.-Y. Oudeyer, "Exploration in model-based reinforcement learning by empirically estimating learning progress," in NIPS, 2012, pp. 206–214.

[13] G. Baldassarre and M. Mirolli, "Deciding Which Skill to Learn When: Temporal-Difference Competence-Based Intrinsic Motivation (TD-CB-IM)," in Intrinsically Motivated Learning in Natural and Artificial Systems, G. Baldassarre and M. Mirolli, Eds. Springer, 2013, pp. 257–278.

[14] A. Baranes and P. Y. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," Robotics and Autonomous Systems, vol. 61, no. 1, pp. 49–73, 2013.

[15] S. Forestier and P.-Y. Oudeyer, "Modular active curiosity-driven discovery of tool use," in IEEE/RSJ IROS, 2016, pp. 3965–3972. [Online]. Available: http://ieeexplore.ieee.org/document/7759584/

[16] M. Rolf, J. J. Steil, and M. Gienger, "Goal babbling permits direct learning of inverse kinematics," IEEE TAMD, vol. 2, no. 3, pp. 216–229, 2010.

[17] P. Delarboulas, M. Schoenauer, and M. Sebag, "Open-Ended Evolutionary Robotics: An Information Theoretic Approach," in PPSN, 2010, pp. 334–343.

[18] J. Lehman and K. O. Stanley, "Abandoning objectives: evolution through the search for novelty alone." Evolutionary computation, vol. 19, no. 2, pp. 189–223, 2011.

[19] A. S. Klyubin, D. Polani, and C. L. Nehaniv, "All Else Being Equal Be Empowered," in ECAL, 2005, pp. 744–753.

[20] V. Kumar, E. Todorov, and S. Levine, "Optimal Control with Learned Local Models : Application to Dexterous Manipulation," in IEEE ICRA, 2016, pp. 378–383.

[21] J. Peters, K. Mülling, and Y. Altün, "Relative Entropy Policy Search," in AAAI, 2010, pp. 1607–1612.

[22] A. Kupcsik, M. P. Deisenroth, J. Peters, A. P. Loh, P. Vadakkepat, and G. Neumann, "Model-based Contextual Policy Search for Data-Efficient Generalization of Robot Skills," Artificial Intelligence, 2015.

[23] J. Schulman, S. Levine, M. Jordan, and P. Abbeel, "Trust Region Policy Optimization," in ICML, 2015, pp. 1889–1897.

[24] A. Abdolmaleki, R. Lioutikov, J. R. Peters, N. Lau, L. P. Reis, and G. Neumann, "Model-Based Relative Entropy Stochastic Search," in NIPS, 2015, pp. 3523–3531.

[25] K. Mülling, J. Kober, and J. Peters, "Simulating human table tennis with a biomimetic robot setup," in SAB, 2010, pp. 273–282.