HARL-A: Hardware Agnostic Reinforcement Learning Through Adversarial Selection

Lucy Jackson[1], Steve Eckersley[2], Pete Senior [2] and Simon Hadfield[1]

***Abstract*— The use of reinforcement learning (RL) has led to huge advancements in the field of robotics. However data scarcity, brittle convergence and the gap between simulation & real world environments, mean that most common RL approaches are subject to over fitting and fail to generalise to unseen environments. Hardware agnostic policies would mitigate this by allowing a single network to operate in a variety of test domains, where dynamics vary due to changes in robotic morphologies or internal parameters. We utilise the idea that learning to adapt a known and successful control policy is easier and more flexible than jointly learning numerous control policies for different morphologies.**

This paper presents the idea of Hardware Agnostic Reinforcement Learning using Adversarial selection (HARL-A). In this approach training examples are sampled using a novel adversarial loss function. This is designed to self regulate morphologies based on their learning potential. Simply applying our learning potential based loss function to current state-of-the-art already provides $\sim$ 30% improvement in performance. Meanwhile experiments using the full implementation of HARL-A report an average increase of 70% to a standard RL baseline and 55% compared with current state-of-the-art.

I. INTRODUCTION

Deep Reinforcement Learning (deep RL), named such due to the use of neural networks as powerful function approximators, has recently demonstrated huge success. Notably, beating world class players at strategic board games [1], controlling continuous robotic systems [2] and playing video games to superhuman standards [3]. However, taxing data requirements, brittle convergence and sensitivity to hyper-parameters still pose issues in the deep RL domain. One aspect of parametric sensitivity which has received limited attention, is sensitivity to the specifics of the simulator or robotic hardware used to train the agent. This creates a huge demand for data collected using real-world hardware, something that is costly, time inefficient and in some cases dangerous. Even worse, data collected in one lab using one robot cannot easily be utilised elsewhere, unless training is to occur on the exact same robot in the exact same configuration. Even then, differences in the calibration of the robots are likely to hurt reproducibility. Hardware agnostic policies would mitigate some of these issues by allowing experiences on semi-identical hardware to be pooled and used collectively, a technique widely used in the supervised learning community. On top of this it would make the sharing, reproduction and testing of results much easier, aiding in the development of advanced intelligent systems [4]. Semi-identical is defined in this paper as a robot of the same fundamental structure (e.g. Degrees of Freedom and action space), but with varying dynamics due to differences in size, shape, construction or calibration.

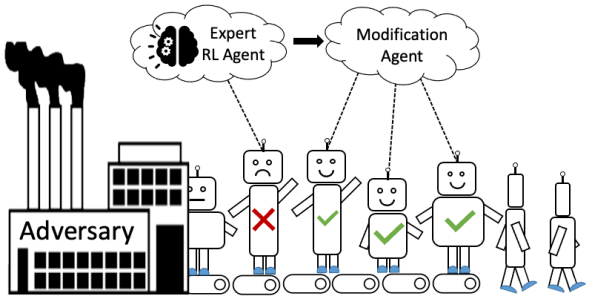


Fig. 1: Representation of HARL-A. Where a single agent can successfully operate on a range of different robots, whose configuration has been determined by an adversary.

The idea of hardware agnostic policies has not been tackled explicitly in prior work, but transfer learning provides a promising start. Transfer learning has been extensively explored in a cross-task setting [5] [6] and more recently in bridging the 'reality-gap': the inability of agents trained in simulation to operate on real world hardware [7]. These methods aim to overcome inter-dependence of hardware properties and learnt policies. This is often considered a form of multi-robot transfer learning, which allows a primary agent to utilise data collected by a second, similar agent carrying out the same task [8]. The source and target domains are not drawn from a distribution at test or training time, but are instead prior knowledge, relating to one or two morphologies. The definition of hardware agnostic policies proposed in this paper is somewhat comparable to that of multi-robot transfer learning but with broader applicability. In this paper training and target domains will be drawn from a continuous distribution of hardware parameters, rather than a discrete number of pre-determined possibilities.

HARL-A is a novel system to train hardware agnostic policies, a representation of which can be seen in Figure 1. It is comprised of a pre-trained/expert policy, a modification policy and an adversary network. The modification policy is conditioned on the change in morphology and learns to adjust the output of a pre-trained 'expert' policy based on the changing hardware. The adversary network is trained using a novel loss function, derived from the agent's ability to learn on each robotic morphology, hence named learning potential. This allows for morphologies on which the agent can never learn to be removed from the training distribution, shifting

*This work was supported by Surrey Satellite Technology Ltd.

[1] University of Surrey, GU2. l.jackson@surrey.ac.uk, s.hadfield@surrey.ac.uk

[2]Surrey Satellite Technology Limited. Tycho House, GU2. s.eckersley@sstl.co.uk, p.senior@sstl.co.uk

focus to areas which are both challenging, and possible. The result is a general policy that is capable of operating on semi-identical robots with zero-shot transfer, meaning no fine-tuning is required when the policy is rolled out on a new robot. Unlike standard approaches to multi-robot transfer learning, the proposed method increases the scope of the agents ability during the learning phase, as oppose to utilising it after data collection.

To summarise, the contributions of this paper are: 1) The first work to explore training of hardware-agnostic RL agents. 2) A novel modification-network architecture for multi-robot learning. 3) A novel learning criteria and adversarial sample selection approach to mitigate sample scarcity in RL training.

II. RELATED WORK

Transfer learning is a popular field of study for RL, and the majority of the literature in this field focuses on the transfer of policies between tasks [5], [6], [9]–[11], known as multi-task transfer learning. Prior work has also looked into transfer between control policies [12], dynamics [13], non-stationary environments [14], and visual inputs [15].

Yu et al. [16] propose using multiple policies with varying behaviors, where each is optimized for a particular dynamic vector. At the search stage, the highest performing policy is selected using co-variance matrix adaptation. Devin et al. [17] also explore the idea of having multiple trained networks that they draw from based on the task at hand. They decompose neural network policies into 'task-specific' and 'robot-specific' modules, where 'task-specific' modules are shared across tasks and 'robot-specific' modules across different robots. The recombination of these modules after training allows for different tasks to be carried out on different robots, however this requires training across options simultaneously, and tasks cannot be learnt in sequence.

Finn et al. [18] move away from the idea of training and selectively using multiple expert networks and instead use meta-learning. They identify network parameters that are most sensitive to changes in a task. This allows a good initialisation of network parameters to allow for 'few-shot' transfer. Nagabandi et al. [19] built on this idea and increased network generalization by treating each time-step as a new task, breaking away from the episodic nature of RL. Again, this method, along with most meta-learning approaches, requires experience in the new setting before the task can be completed. This idea of adapting or fine tuning is something our method does not need.

Chen et al. [20] propose a method that is considered current state-of-the-art in the context of hardware agnostic policies. They augment a physical parameter vector with the environmental observation and use this as the agents observed state. During training the physical parameter vector is randomly selected from a discrete distribution of 7 possibilities. Testing is then carried out on these 7 possibilities plus 1 more. While they report strong results compared to DDPG+HER, the scope of operation of their agent is limited across the training distribution. On top of this, some policies are fine-tuned in the test environment. While this is not as laborious as re-training, zero-shot transfer is required for policies to be truly hardware agnostic. They also show that their explicit method falls short in situations when the task policy is dependent on agent dynamics.

Adversarial networks are common in other areas of machine learning, but are a lesser explored concept within RL. Rajeswaran et al. [21] implemented an adversarial inspired idea in the context of RL to increase robustness of policies in Ensemble Policy Optimization (EPOpt). This dual-step approach consists of policy optimisation using samples from a batch of tasks whereby the distribution of tasks is updated using the lowest performing experiences. Evaluation of the algorithm shows that performing batch optimisation on the worst performing subset of tasks leads to more robust policies.

Pinto et al. [22] advance this idea by using a separate adversarial network to apply perturbation forces to the system during training. Their system, named RARL (robust adversarial reinforcement learning) is set up as a two player zero-sum discounted game, where the adversary's reward is the negative of the agents reward, and the loss functions are derived accordingly. Though this method shows improvements over a baseline, the application of adversarial forces during training does not affect the dynamics of the system. Shioya et al. [23] propose two modifications to the RARL architecture, the first introduces a penalty term to the adversary's loss function for sampling training examples that fall far from the current state. While the second utilises sampling from a memory bank of adversary networks. Both of these methods aimed to tackle the problem outline by Bansal et al. [24], whose experiments showed that always using the hardest environment can hinder the training of the agent. Our method addresses this problem via the novel loss function derived from the learning potential. This ensures that the agent is provided with low performing samples on which it can definitely learn.

III. METHODOLOGY

The goal of this work is to learn a robust, hardware agnostic policy. More specifically a single learnt policy that successfully works on all morphology vectors drawn from the continuous distribution: $v \in \mathcal{V}$. The system follows the structure shown in Figure 2, where an adversary network samples a value of v_T. This is combined with the environment observations, and the action that a canonical expert network would have taken given that observation. This extended state is used as an input to the modification network, which outputs a modified action, suitable for the robot with morphology v_T.

Vector v can be of any size, dependant on the environment and each element represents the parameterisation of a different robotic property. Examples include the dimensions or mass of any limb or part and friction coefficients in joints. During training, as illustrated in Figure 2, one v is selected per episode roll out, hence v_T. This value is not time dependant within any single episode. The per-

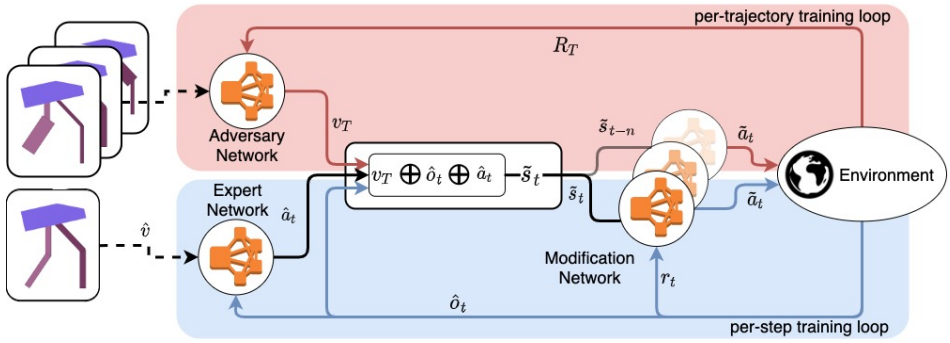


Fig. 2: Flow diagram showing training loops used with HARL-A . The per-trajectory loop is shown in red, and the per-step loop in blue.

trajectory training loop is shown in red, in which whole episodes are condensed into a single event and these are used to optimise the adversary network. In blue is the per-step training loop, in which every step is stored and used to update the modification network. The weights in the expert network are not updated at any point.

A. Reinforcement Learning Formulation

We formulate this as an RL problem in a fully observable environment, modeled as finite horizon Markov Decision Process ($\mathcal{M}$). This is represented by the tuple: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where $s \in \mathcal{S}$ is a point in a continuous state space, $a \in \mathcal{A}$ is a continuous action vector, $\mathcal{P}(s_{t+1}|s_t, a_t)$ is the state transition function, $r(s_t, a_t)$ is the reward function and γ is the reward discount rate. In this setting, the goal of a network is to find parameters θ that maximise the expected return, $R = \sum_{t=0}^{T} \gamma^t r(s_t, a_t)$, over trajectories induced by the policy $\pi_\theta(a_t|s_t)$, which takes action a_t, given state s_t at time step t.

B. Modification Network

The overarching aim of the modification network is to learn a policy, $\widetilde{\pi}_{\widetilde{\theta}}(a_t|s_t)$, parameterised by $\widetilde{\theta}$ (note $\widetilde{}$ indicates parameters relate to the modification network), that is capable of maximising the expected reward for any morphology drawn from distribution $v \in \mathcal{V}$. To achieve this, the network is conditioned on v_t: $\widetilde{\pi}_\theta(a_t|s_t, v_t)$. In order to condition the policy on the robotic morphology, the state space of the modification network differs from that of a standard RL problem. In addition to morphology conditioning, information is leveraged from a pre-trained expert network which already has knowledge of the environmental dynamics. To achieve both of these aims, an observation ($o_t \in \mathcal{O}$) is obtained from the environment and is used by the expert network to predict action a_t. o_t and a_t are then concatenated with the current morphology vector v_T, giving the state space for the modification network as: $\widetilde{s}_t = v_T \oplus o_t \oplus a_t$, where $\oplus$ denotes concatenation.

For a given parameter vector and corresponding transition function, $\mathcal{P}(o_{t+1}|o_t, a_t, v_T)$, also now dependant on v_t the optimal policy parameters can be learnt such that the expected reward is maximized:

$$\widetilde{\theta}^* = \underset{\widetilde{\theta}}{arg\,max} \; \mathop{\mathbb{E}}_{\widetilde{s}_0} \left[\sum_{t=0}^{T} \gamma^t r(\widetilde{s}_t, \widetilde{a}_t) \middle| \mathcal{P}(o_{t+1}|o_t, \widetilde{a}_t, v_T), \widetilde{\pi}_\theta(\widetilde{a}_t|\widetilde{s}_t) \right] \quad (1)$$

Different to standard RL settings, equation 1 shows that the expected return of the modification network is conditioned on the transition function since this varies with v_T. Given this definition, the desired policy should maximise its expected return across all possible transition functions:

$$\widetilde{\theta}^* = \underset{\widetilde{\theta}}{arg\,max} \mathop{\mathbb{E}}_{v_T} \left[\mathop{\mathbb{E}}_{\widetilde{s}_0} \left[\sum_{t=0}^{T} \gamma^t r(\widetilde{s}_t, \widetilde{a}_t) \middle| \mathcal{P}(o_{t+1}|o_t, \widetilde{a}_t, v_T), \widetilde{\pi}_{\widetilde{\theta}}(\widetilde{a}_t|\widetilde{s}_t) \right] \right] \quad (2)$$

There are multiple approaches to optimising equation 2, dependant on the sampling of v. In this paper, first propose sampling v randomly from a pre-defined design space. We refer to this approach as Hardware Agnostic Reinforcement Learning (HARL). One obvious limitation of HARL is that a significant amount of time may be wasted training on samples that already perform well. This leads us to propose the full HARL-A approach, which includes adversarial selection of challenging, but achievable, training samples.

C. Adversarial Network

The role of the adversary is to sample morphology vectors for the modification network to train on. It learns policy $\overline{\pi}_{\overline{\theta}}(v|\widetilde{\theta})$, parameterised by $\overline{\theta}$ (note $\bar{}$ indicates parameters relate to the adversary network) and conditioned on the current version of $\widetilde{\pi}_{\widetilde{\theta}}$. Intuition says that the modification network should train on the lowest performing morphology vectors, in order to improve overall behaviour. However, when dealing with a large continuous design space there are inevitably morphologies on which the agent can never learn to 'complete the task'. In a standard adversarial set-up, since these samples have the lowest performance, they would continually be selected for training, wasting valuable resources. Instead we introduce the concept of learning potential (l_p).

This is calculated as the ratio between performance in the environment following policy ($\widetilde{\pi}_{\tilde{\theta}_i}$) and performance in the environment following the same policy after K updates, ($\widetilde{\pi}_{\tilde{\theta}^{i+K}}$):

$$l_p(v_T) = \frac{R_{\widetilde{\pi}_{\tilde{\theta}^i}}\lambda}{R_{\widetilde{\pi}_{\tilde{\theta}^{i+k}}} + \lambda} \tag{3}$$

where λ is a constant used to ensure that division by zero never occurs. $l_p(v_T)$ is interpreted as the agents ability to learn a more effective policy for that morphology.

D. Implementation of HARL-A

Algorithm 1 outlines the system training. The modification network is trained using a standard a PPO algorithm implemented in PyTorch. The objective function used when training the adversary is a variation on the standard PPO algorithm [25], where the l_p is used as opposed to the normalised advantage:

$$\mathcal{L} = min\left(-l_p\nabla_t, -l_p clip(\nabla_t, 1-\epsilon, 1+\epsilon)\right) - \alpha H(v_T) \tag{4}$$

Here ∇_t represents the ratio between policy updates, $H(v_T)$ is the entropy of the adversary, calculated over its output. ϵ is the clipping factor and α is the entropy coefficient. l_p is negated since the objective function is minimised during training.

Training alternates between the modification network and the adversary network. Prior to training the adversary a copy is made of the modification network weights ($\widetilde{\theta}$) and this is used to determine l_p. $\widetilde{N}$ and $\overline{N}$ are the number of agent and adversary updates that are carried out before training switches. $\mathcal{Z}$ is the number of steps per update for the modification network, and K is the number of updates used to calculate l_p. For all experiments in this paper $K = 10$.

IV. RESULTS AND DISCUSSION

We demonstrate the performance of HARL and HARL-A in three environments:

CartPole (CP): This is a standard OpenAI Gym control environment [26]. The task is to balance a pole above a cart, it is attached through an un-actuated joint and the system can apply a force of ± 1 to the cart which slides over a frictionless track. Experiments are carried out varying the length and mass of the pole and the mass of the cart. A point is awarded for each time step the pole remains above the cart ($\pm 15°$), a score of above 1000 is considered as solving the problem. The expert network is trained for a pole length of 1 and a cart mass of 1.

BipedalWalker (BW): This is another standard environment from the OpenAI Gym control package. Here a bipedal robot aims to move forwards with a reward earned based on efficiency over distance travelled. If the robot falls it gets -100 points and applying a motor torque costs points. A more efficient walking pattern therefore earns a higher final score. In this paper a score of above 250 is deemed as successful. In our experiments either the size of the entire walker is varied: $v_T \in \mathbb{R}^1$, or the length and width of each leg segment is varied independently: $v_T \in \mathbb{R}^8$. For both

Algorithm 1 Implementation of HARL-A

1: Randomly initialise networks $(\widetilde{\theta}, \overline{\theta})$
2: **for** $i = 1, 2, \ldots$ total number of updates **do**
   # Modification Network Update
3: $\quad\widetilde{M} \leftarrow \varnothing$
4: $\quad$**for** $j = 1, 2, \ldots, \widetilde{N}$ **do**
5: $\quad\quad v_T^j \sim \overline{\pi}_\theta$
6: $\quad\quad o_0^j \sim \mathcal{O}$
7: $\quad\quad$**for** $t = 1, 2, \ldots, \mathcal{Z}$ **do**
8: $\quad\quad\quad a_t^j \sim \pi(o_t^j)$
9: $\quad\quad\quad \widetilde{s}_t^j = (v_T^j \oplus o_t^j \oplus a_t^j)$
10: $\quad\quad\quad \widetilde{a}_t^j \sim \widetilde{\pi}_\theta(\widetilde{s}_t^j)$
11: $\quad\quad\quad \widetilde{o}_{t+1}^j \sim \mathcal{P}(o_t^j, \widetilde{a}_t^j, v_T^j)$
12: $\quad\quad\quad \widetilde{M} \leftarrow \widetilde{M} \cup (o_t^j, \widetilde{a}_t^j, r_t^j, v_T^j, o_{t+1}^j)$
13: $\quad\quad$**end for**
14: $\quad\quad$Sample a random mini batch from $\widetilde{M}$
15: $\quad\quad$Optimise $\widetilde{\theta}$ according to equation 2
16: $\quad$**end for**
   # Adversary Network Update
17: $\quad\overline{M} \leftarrow \varnothing$
18: $\quad\widetilde{\theta}^{i\prime} \leftarrow \widetilde{\theta}^i$
19: $\quad$**for** $J = 1, 2 \ldots, \overline{N}$ **do**
20: $\quad\quad v_T^j \sim \overline{\pi}_\theta$
21: $\quad\quad$**for** $k = 1, 2, \ldots K$ **do**
22: $\quad\quad\quad o_0^j \sim \mathcal{O}$
23: $\quad\quad\quad$Optimise $\widetilde{\theta}'$ over a full roll out (lines 7 to 15)
24: $\quad\quad$**end for**
25: $\quad\quad$Calculate l_p according to equation 3
26: $\quad\quad\overline{M} \leftarrow \overline{M} \cup (v_T^j, l_p^j)$
27: $\quad\quad$Sample a random mini batch from $\overline{M}$
28: $\quad\quad$Optimise $\overline{\theta}$ using equation 4
29: $\quad$**end for**
30: **end for**
31: **return** $\widetilde{\theta}$, $\overline{\theta}$

situations the expert network is trained for a walker of scale or leg size 1.

SpaceRobot (SR): This is a novel environment developed for use with this work [1]. It comprises of a small, 10 Degree of Freedom free-flying space robot aiming to reach its end effector to a payload, the position of which is randomly assigned at the start of each episode. A reward of -10 is given if the joint limits of the robotic arm are exceeded, 100 points are awarded for successful capture of the payload if the base has a sufficiently low velocity, 40 points are given if the payload is captured but the base's velocity is above the threshold value, any form of capture is considered a success. Lastly, a reward of -50 is given if any part of the robot, excluding the end effector, collides with the payload. This environment is used as it demonstrates a complex control problem in three dimensions, using a sparse reward where changes to the morphology have a huge impact on the system dynamics [2]. Therefore showing the ability of our method to deal with complex control algorithms and design spaces. For experiments the sizes of each link, on the four link robot, are varied individually. The expert network was trained

[1]Code available at `https://gitlab.eps.surrey.ac.uk/lj00304/spacerobot_v1`

[2]The micro-gravity operating environment means that every joint actuation creates an equal and opposite reaction upon the base. This is a phenomenon called dynamic coupling that makes control very challenging.

TABLE I: Information on morphology vectors used during experiments. A, B and C represent different experiments of varying difficulty level.

Parameters			A	B	C
CP	Pole	$v_T \in \mathbb{R}^1$			[0.5,10]
	Cart, Pole	$v_T \in \mathbb{R}^2$			[0.5,10]
SR	Individual	$v_T \in \mathbb{R}^4$		[0.25,0.5]	[0.5,0.75]
BW	Whole	$v_T \in \mathbb{R}^1$	[1,1.25]	[0.75,1]	[0.75,1.25]
	Individual	$v_T \in \mathbb{R}^8$	[1,1.25]	[0.75,1]	[0.75,1.25]

with link lengths, $0.25m$, $0.25m$, $0.25m$ and $0.05m$. Due to the environment's sparse reward, l_p is calculated for this environment with R approximated as the $\%$ success over 50 episodes.

Balancer: We also demonstrate the successful application of HARL-A on a physical balance robot, whose task is to remain upright by actuating a single wheel-axle. Videos of its performance under a single policy on a range of sizes can be found in the supplementary material [3].

In this paper, a range of experiments were run to evaluate HARL-A, these use the range of morphologies quoted in Table I. Each experiment was run 3 times with 3 different seeds, for $1e^7$ steps in the corresponding environment and all results are an average of the 3 runs. To fairly evaluate performance over these ranges, a consistent test distribution was used for each experiment. This consists of 500 random morphology vectors from the defined range. We evaluate our method against current state-of-the-art for across morphology control - hardware conditioned policies (HCP) [20], and a standard PPO baseline.

A. *Modification Network vs. Direct Learning*

We first show the training progress of HARL compared to an RL baseline, where the RL agent is trained using a random sample from $\mathcal{V}$ for each T. Figure 3 shows performance as a function of training iteration for BW-whole C. The policies learnt using HARL reach a higher performance at a quicker rate demonstrating that these tasks cannot be easily solved with traditional hardware-dependent agents. Comparison to HCP, also in Figure 3 shows how learning to modify the output of a pre-trained network in contrast to learning a single robust policy leads to higher performance at a quicker rate. More detailed results can be seen in Table II. In all cases the performance of HARL improves on both the PPO and HCP baseline, validating our initial motivation. On top of this, in almost all cases the improvement seen between HCP and HARL is much larger than PPO and HCP. In fact, for experiment SR-individual C neither PPO or HCP learnt to reach the payload.

Figure 4 shows how the policies learnt for BW-whole C perform as a function of the walker scale for all control policies evaluated. Proving the increased generalisation capability of HARL compared with the PPO and HCP. Here it can be seen that performance of HARL also leads to increased performance outside of the training range, shown by the black dotted line.

[3] https://www.youtube.com/watch?v=oxuByrlig2M

TABLE II: Comparison between proposed method HARL, HCP and standard PPO. Quoted are the $\%$ success rates over test distributions.

Experiment			PPO	HCP	HARL
CP	Pole	C	29	41	**44**
	Pole and Mass	C	25	31	**37**
SR	Individual	B	69	61	**88**
		C	0	0	**35**
BW	Whole	A	62	68	**71**
		B	27	52	**74**
		C	39	44	**70**
BW	Individual	A	43	43	**72**
		B	29	31	**38**
		C	24	20	**26**

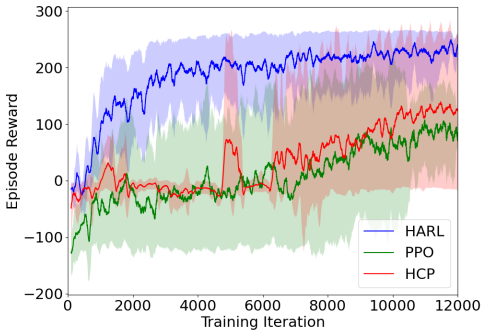


Fig. 3: Reward against training iteration for BW-whole C for HARL, HCP and PPO (maximum, minimum and mean).

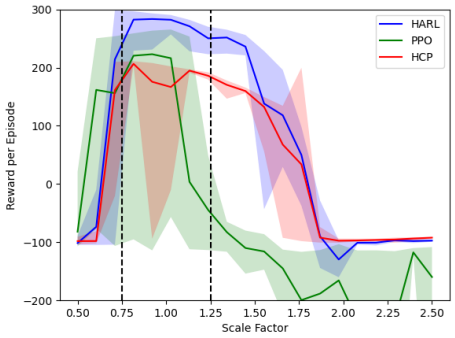


Fig. 4: Reward as a function of BW (whole C) scale for HARL, HCP and PPO (maximum, minimum and mean). The black dotted line shows the range of scales used during training.

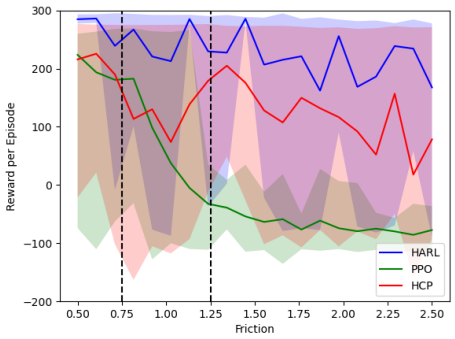


Fig. 5: Comparison of reward as a function of the coefficient of friction at the hip and knee joints of the BW for HARL, HCP and PPO (maximum, minimum and mean results).

TABLE III: % success rates over test distributions for HARL-A and HCP-A. Non-adversarial counterparts are included for comparison.

Experiment			HCP	HCP-A	HARL	HARL-A
CP	Pole	C	41	*54*	44	**56**
	Pole and Mass	C	31	35	**37**	*35*
SR	Individual	B	61	87	*88*	**90**
		C	0	0	*35*	**36**
BW	Whole	A	68	*73*	71	**87**
		B	52	73	*74*	**82**
		C	44	*77*	70	**84**
BW	Individual	A	43	*76*	72	**85**
		B	31	*39*	38	**46**
		C	20	*30*	26	**45**

Having shown that our method improves performance over changing physical morphologies we now show that it can have the same level of improvement when internal parameters are varied. A new experiment is introduced here, whereby the friction coefficient at the joints of the hip and knee of the BW are varied. Figure 5 shows how performance of the walker varies for HCP, PPO and HARL. Again, this shows that the implementation of our method drastically improves the performance of a single policy over a range of morphologies with varying internal parameters.

B. Adversarial Learning Evaluation

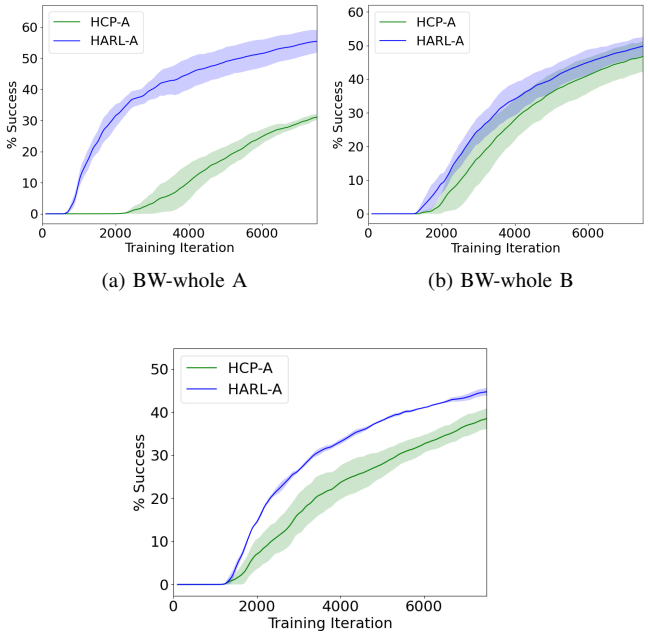
Next we demonstrate the benefits of the adversarial training using l_p over both the standard HARL and HCP systems, named HARL-A and HCP-A respectively. Results are shown in Table III. For all experiments, except one, HARL-A shows the highest level of performance. In CP-pole and mass HARL shows the highest performance. However, all results for CP, other than the benchmark are similar. It is thought that this is due to the limited effect that changing the physical parameters has on the dynamics of the simple problem, as well as the fact that the system is near saturation.

The performance of HCP-A is second best in 6 out of 10 cases, proving the importance of the novel l_p loss. In fact, excluding CP-pole and mass and SR-individual C, all experiments where the adversary is included show better results than their non-adversarial counterparts. It is thought that the highly non-linear and coupled dynamics of the SR meant HCP/HCP-A was unable to learn to reach the payload.

Since all HARL-A experiments for BW and CP shown here were carried out using an expert network trained using PPO, the same set of experiments were run again using TRPO [27] and DDPG [2]. In all cases, results matched that of HARL-A + PPO, showing that HARL-A is also agnostic to the model used to train the expert policy, provided it gives a high level of performance. Figure 6 serves to demonstrate the continued benefit of the modification network in the presence of the adversarial training regime. It shows a comparison between HCP-A and HARL-A for all BW-whole experiments. Here it can be seen that in all cases HARL-A begins to learn quicker and at a faster rate throughout training, as well as reaching a higher final performance.

C. Use of Learning Potential

The intuition behind sampling using l_p as opposed to traditional hard negative mining is now evaluated. The idea



(a) BW-whole A (b) BW-whole B

(c) BW-whole C

Fig. 6: Running percentage performance during the first 7500 training iterations. Graphs show the mean, maximum and minimum.

TABLE IV: Comparison between HARL used with hard negative mining and HARL-A. All experiments are run with BW-whole.

BW-whole	HARL	HARL-NM	HARL-A
A	71	*79*	**87**
B	74	*76*	**82**
C	70	*77*	**84**
D	32	*41*	**53**

of hard negative mining is to select the lowest performing morphologies from $\mathcal{V}$ to train the system. We compare our results from HARL-A to those collecting using HARL with hard negative mining (HARL-NM). Due to issues with the scalability of NM experiments were only computed on BW-whole A-C, the results are shown in Table IV. A fourth set of design space bounds is introduced here (BW-whole D), where scales range from 0.5 to 2, in order to demonstrate how HARL-A works over larger parameter spaces. For all cases it can be seen that the performance of HARL-NM improves that of HARL, but HARL-A shows the best results. For experiment BW-whole A, the change from HARL-NM to HARL-A leads to a 10% increase in performance, where as in experiment D a 25% increase is seen. This disparity is due to the fact that in the easier problems all of the morphologies can be learnt on, meaning that selection of the lowest performing examples is acceptable. Whereas, when the scale of the walker is > 1.8, as in experiment BW-whole D, the learning of a policy is much less stable. With the use of HARL-A, the adversary learns to stop presenting the agent with those morphologies, allowing it to spend valuable time learning at other values of v.

It is also worth re-iterating that a naive hard negative mining approach is intractable for any system of reasonable complexity. Most of the experiments in this paper would

take hundreds of years to complete a single iteration of hard negative mining.

V. CONCLUSIONS

In this paper we presented HARL-A: a system to train a new type of hardware agnostic policy using a form of adversarial selection. This system aims to tackle the problem of generalisation in domains with varying dynamics, i.e across similar robots with different morphologies. Tackling this problem will allow for the sharing of training data across labs and improvements in the reproducability of results. We first presented the idea of a modification network architecture to enable multi-robot learning. This lead to performance improvements in current state-of-the-art of around 45%. We then presented the idea of adversarial selection using learning potential. This is a novel metric that quantifies the ability of a policy to train on a particular robot morphology. The implementation of this metric in the adversarial network used in conjunction with the modification network (the full HARL-A system) lead to $\sim 70\%$ increase in the ability of a single policy to act on a range of semi-identical robots. In the future this architecture could be advanced to allow for generalisation across robots with different degrees of freedom, or larger variations in morphology. This would advance the field of RL and be a huge step towards the over arching goal of general AI.

VI. ACKNOWLEDGEMENTS

This work was partially supported by Surrey Satellite Technology Ltd and the UK Engineering and Physical Sciences Research Council (EPSRC) grant agreement EP/S035761/1 'Reflexive Robotics'.

REFERENCES

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature (London)*, vol. 529, no. 7587, pp. 484–489, 2016.

[2] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. M. O. Heess, T. Erez, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Proc. in 4th International Conference on Learning Representations (ICLR)*, San Diego, Ca, USA, 7-9, May San Juan, Puerto Rico, 2015.

[3] Y. Lui, "Deep reinforcement learning: An overiview," *CoRR*, vol. abs/1701.07274, 2017.

[4] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *32nd AAAI Conference on Artificial Intelligence (AAAI)*, New Orleans, USA, 2018.

[5] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *Journal of Machine Learning Research*, vol. 10, no. 56, pp. 1633–1685, 2009.

[6] J. Fu, S. Levine, and P. Abbeel, "One-shot learning of manipulation skills with online dynamics adaptation and neural network priors," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea, 2016, pp. 4019–4026.

[7] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, 2019, pp. 8973–897.

[8] M. K. Helwa and A. P. Schoellig, "Multi-robot transfer learning: A dynamical system perspective," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, 2017, pp. 4702–4708.

[9] L. Pinto and A. Gupta, "Learning to push by grasping: Using multiple tasks for effective learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Marina Bay Sands, Singapore, 2017, pp. 2161–2168.

[10] T. T. Um, M. S. Park, and J. Park, "Independent joint learning: A novel task-to-task transfer learning scheme for robot models," Hong Kong, China, 2014, pp. 5679–5684.

[11] A. Barreto, D. Borsa, J. Quan, T. Schaul, D. Silver, M. Hessel, D. Mankowitz, A. Zidek, and R. Munos, "Transfer in deep reinforcement learning using successor features and generalised policy improvement," in *Proceedings of Machine Learning Research*, vol. 80, 2018, pp. 501–510.

[12] A. Murali, L. Pinto, D. Gandhi, and A. Gupta, "Cassl: Curriculum accelerated self-supervised learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May Brisbane, QLD, 2018, pp. 6453–6460.

[13] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," *CoRR*, vol. abs/1710.06537, 2017.

[14] M. Al-Shedivat, T. Bansal, Y. Burda, I. Sutskever, I. Mordatch, and P. Abbeel, "Continuous adaptation via meta-learning in nonstationary and competitive environments," *CoRR*, vol. abs/1710.03641, 2017.

[15] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, 2017.

[16] W. Yu, C. K. Liu, and G. Turk, "Policy transfer with strategy optimization," *CoRR*, vol. abs/1810.05751, 2018.

[17] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine, "Learning modular neural network policies for multi-task and multi-robot transfer," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Marina Bay Sands, Singapore, 2017, pp. 2169–2176.

[18] C. Finn, P. Abbel, , and S. Levine, "Model-agnostic meta-learning for fast adapation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia, 2017, pp. 1126–1135.

[19] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn, "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning," *CoRR*, vol. arXiv:1803.11347, 2018.

[20] T. Chen, A. Murali, and A. Gupta, "Hardware conditioned policies for multi-robot transfer learning," *CoRR*, vol. abs/1811.09864, 2018.

[21] A. Rajeswaran, S. Ghotra, S. Levine, and B. Ravindran, "Epopt: Learning robust neural network policies using model ensembles," in *Proc. 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[22] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *Proc. of the 34th International Conference on Machine Learning (ICML)*, Sydney, Australia, 2017.

[23] H. Shioya, Y. Iwasawa, and Y. Matsuo, "Extending robust adversarial reinforcement learning considering adaptation and diversity," *Proc. 6th International Conference on Learning Representations (ICLR)*, Vancouver, BC, 2018.

[24] T. Bansal, J. Pachocki, S. Sidor, I. Sutckever, and I. Mordatch, "Emergent complexity via multi-agent competition," in *Proc. 6^{th} International Conference on Learning Representations (ICLR)*, Vancouver, BC, 2018.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.

[26] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *CoRR*, vol. abs/1606.01540, 2016.

[27] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul Lille, France, 2015, pp. 1889–1897. [Online]. Available: http://proceedings.mlr.press/v37/schulman15.html