Curriculum Learning through Distilled Discriminators

Rahul Siripurapu
USI
rahul.siripurapu@gmail.com

Jürgen Schmidhuber*
IDSIA, USI, SUPSI
juergen@idsia.ch

Louis Kirsch*
IDSIA, USI, SUPSI
louis@idsia.ch

Abstract

Engineering reward functions to successfully train an agent and generalize to downstream tasks is challenging. Automated Curriculum Learning (ACL) proposes to generate tasks instead. This avoids spurious optima and reduces engineering effort. Recent works in ACL generate policies and tasks jointly such that the policies can be used as an initialization to quickly learn downstream tasks. However, these methods expose the agent to task distribution shifts, which may hurt its generalization capability. We automatically design a curriculum that helps overcome this drawback. First, we show that retraining the agent from scratch on a stationary task distribution improves generalization. Second, given a generated task, we construct an easier task by distilling the reward function into a narrower network reducing reward sparsity. Using this distilled task to form a simple curriculum, we obtain better generalization to downstream tasks compared to our baseline retrained agent.

1 Introduction

Nature and Nurture are both important for human intelligence. Being raised in poverty is known to impair intelligence (Mani et al., 2013). A bonobo chimp raised as a human may be the most intelligent chimp and yet be no match for an adult human (Roth and Dicke, 2005). The curricula used in our society are constantly adapted, including techniques such as spaced repetition (Landauer and Bjork, 1978) to allow for faster and efficient learning. Curricula are thus a driving force behind intelligence. This has motivated the pursuit of Automated Curriculum Learning (ACL) methods (Schmidhuber, 1991a, 2012; Portelas et al., 2020a). Essentially, these methods are designed to generate tasks in an order that expedites downstream learning, using objectives such as curiosity (Schmidhuber, 1991a), novelty (Lehman and Stanley, 2008), diversity (Lehman and Stanley, 2011), difficulty (Sukhbaatar et al., 2018), and learning progress (Schmidhuber, 1991b; Baranes and Oudeyer, 2013).

Many ACL methods jointly adapt both the agent and the task distribution (Florensa et al., 2018; Eysenbach et al., 2018; Jabri et al., 2019; Wang et al., 2019; Racaniere et al., 2020). Diversity Is All You Need (DIAYN) (Eysenbach et al., 2018), for example, learns a set of diverse skills (tasks) that allow for generalization to downstream tasks. Due to joint training, the tasks adapt to the policies and the policies adapt to the tasks. However, recent works have shown that distribution shifts during training can significantly impact generalization (Ash and Adams, 2019; Igl et al., 2020). Hence, we expect that the non-stationary task distributions in such methods would impair generalization capability. We show that this is indeed the case for DIAYN, as an agent retrained from scratch on the pre-trained and frozen task distribution generalizes much better to downstream tasks than the original DIAYN agent. This finding is related to Portelas et al. (2020b), who investigated retraining in the context of ALP-GMMs.

However, the agent retrained directly on the generated task distribution does not learn the skills as well as the original agent. This is because the original agent, apart from the multi-task training, had

*Co-Advising

Workshop on Deep Reinforcement Learning at NeurIPS 2020, Vancouver, Canada

the added benefit of the tasks being of *intermediate difficulty* (Portelas et al., 2020a), due to the joint training. Using tasks of intermediate difficulty is an important aspect of good curricula (Schmidhuber, 2002, 2012; Bengio et al., 2009; Graves et al., 2017; Justesen et al., 2018). We demonstrate a novel method to simplify the tasks by distilling the corresponding learned reward functions (LRFs) into a lesser width network. Further, we show that this method works by ameliorating the sparsity of the task reward.

Our paper makes three contributions. First, we show that retraining can improve generalization on downstream test tasks, in the context of DIAYN. Second, we show that distillation into a narrower network helps to simplify the task by ameliorating the reward sparsity of the LRF. Finally, we show that curricula can be automatically designed through such distillation. We use the retrained agent as a baseline to demonstrate that the generalization improvement comes from the curriculum, and not merely from retraining. We call this method Curriculum LeArning through Distilled discriminators (CLAD).

2 Background

In this section, we motivate our work in the context of recent results on distributional shifts and curricula, compare with related works that use distillation to improve generalization, and summarize DIAYN, the work that we build upon.

Warm starting, curricula, and anti-curricula It was shown that warm starting a network by training it on half of the dataset before training on the full dataset hurts generalization due to the shift in data distribution during training (Ash and Adams, 2019). Although warm starting or pre-training tends to help in few-shot learning settings, generalization is impaired as the training data increases. This is in stark contrast to results from Curriculum Learning, where we see that distribution shifts which correspond to a good curriculum actually improve generalization (Weinshall et al., 2018). This apparent contradiction can be reconciled as 'anti-curricula' (Weinshall et al., 2018) that show worse results than a stationary data distribution. Thus, good curricula require specific distribution shifts that improve generalization, while other distribution shifts impair it (Elman, 1993).

Distillation for improved generalization Distillation (Schmidhuber, 1992, 2018; Hinton et al., 2015) has been shown to improve generalization in multiple recent works (Mobahi et al., 2020; Igl et al., 2020; Portelas et al., 2020b). The key idea is that distilling the knowledge from a trained network would avoid the distribution shifts. Self-distillation (Furlanello et al., 2018; Mobahi et al., 2020) improves generalization even in the absence of non-stationarity as the student network gets better gradients from the soft labels of a teacher network, making learning easier. In contrast to these works, we distill the LRFs rather than the agent, to generate reward functions that are easier to optimize.

Retraining from scratch in ACL Portelas et al. (2020b) demonstrate the generalization gaps in the context of Curriculum Learning based on procedural task generation and absolute learning progress (LP). They claim that during automatic curriculum generation using ALP-GMMs (Portelas et al., 2019), the initial exploration phase for discovering the curriculum may have a 'cluttering effect' that impairs the agent. They distill the curriculum by using LP to guide the task distribution order and selection. In our work, we design the curriculum by constructing easier tasks. Adopting the terminology from Portelas et al. (2020a) the former approach combines environment generation based on LP in the data collection phase, with environment selection based on LP in the data exploitation phase. Our approach combines LRFs based on diversity in the collection phase with subsequent LRF distillation and ordering based on difficulty in the exploitation phase.

DIAYN In our experiments, we use DIAYN to analyze the benefits of retraining and curriculum distillation in ACL. DIAYN learns a task distribution by maximizing an objective related to empowerment (Salge et al., 2013), a measure of control an agent has over the environment. The approach defines a skill-conditioned policy and a discriminator-based reward function that are trained cooperatively. Skills are discrete latent variables that influence the policy behavior. The discriminator is optimized to predict the skill, given a state, while the policy is rewarded to visit states that are more discriminable. Each skill has to occupy a different part of the state space to be easily discriminable leading to a soft partition of the state space. Using the resulting policy as an initialization improves

learning on downstream tasks. DIAYN uses the Soft Actor Critic framework (Haarnoja et al., 2018) to optimize the policy. An illustration of the algorithm can be found in appendix A.

3 Methods

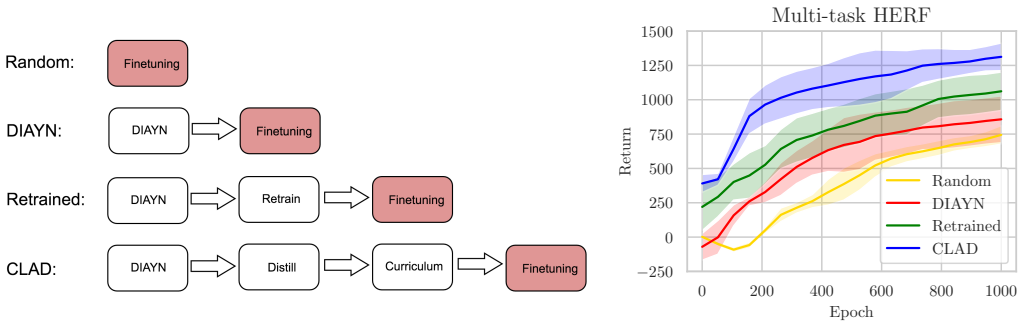

Figure 1: (Left) **Training process:** The training steps that occur prior to the finetuning step in the four different variants we compare. (Right) **Generalization via curriculum:** A comparison of the generalization advantage on a multi-task Human Engineered Reward Function (HERF) benchmark consisting of 25 different tasks in the cheetah environment. The tasks correspond to eight combinations of running/flipping forwards/backwards and standing/hopping on the front/back foot. Policies are conditioned on the tasks and trained on a randomly sampled task in each epoch. The curves plot the average episodic return over all of the 25 tasks. We see that the CLAD agent generalizes better than the retrained agent which in turn outperforms DIAYN. All curves are 5 seed averages.

Retraining to avoid distribution shifts In DIAYN, the discriminator based reward function is trained jointly with the agent. The skill state-distribution that the discriminator fits, changes as the agent learns, producing a changing task distribution. We can overcome this by taking the fully trained discriminator and then retraining a randomly initialized agent in a manner similar to DIAYN but with the discriminator frozen. We expect that this *retrained agent* would learn faster than the DIAYN agent on downstream test tasks.

Distillation to ameliorate reward sparsity In DIAYN, as in the retraining, the multi-task training provides a curriculum. But in DIAYN, the discriminators are jointly trained with the agent, with the task difficulty adjusted to the agent's capability. We lose this advantage during retraining and consequently, we observe that the retrained agent fails to learn certain skills as well as the original agent (See appendix C). We hypothesize that the DIAYN discriminator produces a sparse reward, making the task difficult to learn for an agent starting from scratch. Hence, we propose a simple distillation method to construct a reward function that provides the agent with better gradients to speed up learning. To this end, we train a lesser-width discriminator on the DIAYN skill state-distribution. See appendix D for additional details.

Curriculum Learning Through Distilled Discriminators By first training the agent on the lesser-width distilled discriminator and then subsequently on the full-width discriminator, we obtain a simple curriculum. We show that our CLAD agent generalizes better than the retrained agent on the finetuning task, confirming that the advantage is specifically due to the curriculum. Figure 1 (left) illustrates the training process involved in producing the four agents we compare. Figure 1 (right) compares the generalization capabilities on a multi-task benchmark.

4 Experiments and Results

We adopt the same experimental methods as in DIAYN and demonstrate our advantage on continuous control tasks from OpenAI Gym (Brockman et al., 2016) based on MuJoCo (Todorov et al., 2012). We consider the Half-Cheetah, Ant, and Hopper environments and evaluate the generalization advantage by how quickly the reward is maximized on these tasks. We expect the retrained agent to outperform DIAYN and the CLAD agent to outperform pure retraining.

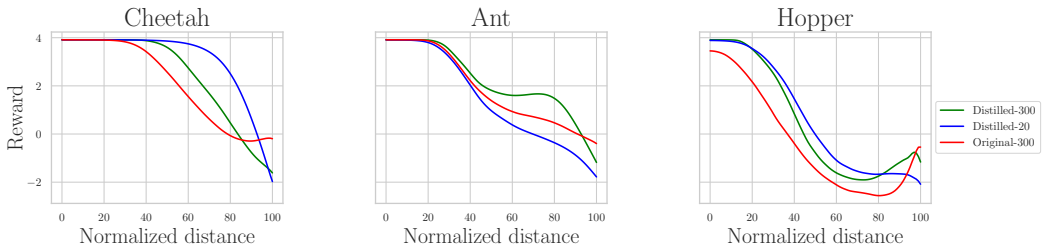

Figure 2: **Discriminator reward profile:** A comparison of the average discriminator reward profile along a cross-section with the initial state at 100 and the skill state at 0. In all environments the 300-wide DIAYN discriminator (red) has the sparsest reward (regions of low gradient or valleys before reaching the maximum). The 20-wide distilled discriminator (blue) has a more consistent gradient, making it easier to optimize (Experimental validation in appendix G). The 300-wide distilled discriminator (green) as a control, shows the advantage comes from reducing the width and not from the distillation.

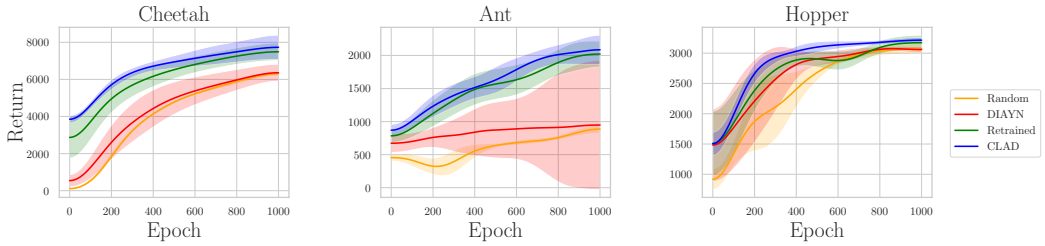

Figure 3: **Generalization via curriculum:** A comparison of the generalization advantages of CLAD on downstream tasks with the average return of 5 seeds across three environments. A consistent ordering of the randomly initialized agent (orange), the DIAYN agent (red), the retrained agent (green), and the CLAD agent (blue) can be observed. This suggests that the changing task distribution due to the joint training of the discriminator in DIAYN hurts the agent's generalization capabilities. In addition to retraining, the distilled discriminator curriculum can further improve generalization to downstream tasks.

Retraining Our retrained agent uses the same experimental setup and hyperparameters as in DIAYN (Refer to appendix B). We train the agent from scratch for 1000 epochs using the trained DIAYN discriminator. Compared to DIAYN, this retrained agent generalizes much better to downstream tasks on all three environments as shown in figures 1 and 3. We note that the comparisons in the DIAYN paper are produced after training DIAYN for 10k epochs while we train for a much shorter duration. Further, they compare the maximum rather than the mean. Refer to appendix E for further analysis.

Network width effects To make the tasks easier in terms of reward sparsity, we distill the 300-wide, trained DIAYN discriminator into a 20-wide network. Figure 2 shows the reward profile of the distilled discriminator. We sample 1000 paths from a random start state to a random high scoring skill state and we plot the average reward profile of all the paths. A similar interpolation method has been previously used (Goodfellow et al., 2015). We see that the 20-wide discriminator has a more consistent gradient compared to the other variants. Empirical verification of the ease of maximization of the distilled reward function can be found in appendix G.

CLAD We train the agent first on the distilled discriminator for half the epochs (500) followed by the original DIAYN discriminator for the remaining half. In figures 1 and 3, we see that this improves generalization on downstream tasks in multiple environments when compared with the performance of the retrained agent. Thus we can conclude that the generalization advantage can be attributed specifically to the use of a curriculum with the task distribution ordered by difficulty. Further ablation can be found in appendices H, I and J.

5 Conclusions

Curricula are special distribution shifts that improve learning on downstream tasks. We demonstrated that the implicit curriculum generated by DIAYN also generates non-stationarity that impedes future learning. First, by retraining a new agent on the generated tasks we obtain better downstream performance. Second, we introduced simple curricula for learned reward functions by distilling them into narrower networks. This distillation ameliorates the reward sparsity of the learned reward function and subsequently helps in outperforming the retrained agent.

To understand the ease of maximization of reward functions, the approach to visualising high-dimensional reward functions by interpolating rewards directly in state-space is useful but only serves as a heuristic. There is no guarantee that the intermediate states are actually reachable in the given environment dynamics. Further, distilling reward functions into lower width networks is a crude but simple and effective way to reduce reward sparsity. Dirichlet energy minimisation may be an interesting future direction (Rudin et al., 1992; Drucker and Le Cun, 1992; Finlay et al., 2019).

Other ACL methods (Sharma et al., 2020; Jabri et al., 2019) also use learned reward functions based on neural networks, and hence they may also benefit from the use of CLAD. We leave this to future work.

References

Ash, J. T. and Adams, R. P. (2019). On the Difficulty of Warm-Starting Neural Network Training. *arXiv:1910.08475 [cs, stat]*.

Baranes, A. and Oudeyer, P.-Y. (2013). Active Learning of Inverse Models with Intrinsically Motivated Goal Exploration in Robots. *Robotics and Autonomous Systems*, 61(1):49–73.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pages 1–8, Montreal, Quebec, Canada. ACM Press.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). OpenAI Gym. *arXiv:1606.01540 [cs]*.

Drucker, H. and Le Cun, Y. (1992). Improving generalization performance using double backpropagation. *IEEE Transactions on Neural Networks*, 3(6):991–997.

Elman, J. L. (1993). *Learning and Development in Neural Networks: The Importance of Starting Small*.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. (2018). Diversity is All You Need: Learning Skills without a Reward Function. In *International Conference on Learning Representations*.

Finlay, C., Calder, J., Abbasi, B., and Oberman, A. (2019). Lipschitz regularized Deep Neural Networks generalize and are adversarially robust. *arXiv:1808.09540 [cs, math, stat]*.

Florensa, C., Held, D., Geng, X., and Abbeel, P. (2018). Automatic Goal Generation for Reinforcement Learning Agents. *arXiv:1705.06366 [cs]*.

Furlanello, T., Lipton, Z. C., Tschannen, M., Itti, L., and Anandkumar, A. (2018). Born Again Neural Networks. *arXiv:1805.04770 [cs, stat]*.

Goodfellow, I. J., Vinyals, O., and Saxe, A. M. (2015). Qualitatively characterizing neural network optimization problems. *arXiv:1412.6544 [cs, stat]*.

Graves, A., Bellemare, M. G., Menick, J., Munos, R., and Kavukcuoglu, K. (2017). Automated Curriculum Learning for Neural Networks. *arXiv:1704.03003 [cs]*.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *arXiv:1801.01290 [cs, stat]*.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the Knowledge in a Neural Network. *arXiv:1503.02531 [cs, stat]*.

Igl, M., Farquhar, G., Luketina, J., Boehmer, W., and Whiteson, S. (2020). The Impact of Non-stationarity on Generalisation in Deep Reinforcement Learning. *arXiv:2006.05826 [cs, stat]*.

Jabri, A., Hsu, K., Eysenbach, B., Gupta, A., Levine, S., and Finn, C. (2019). Unsupervised Curricula for Visual Meta-Reinforcement Learning. *arXiv:1912.04226 [cs]*.

Justesen, N., Torrado, R. R., Bontrager, P., Khalifa, A., Togelius, J., and Risi, S. (2018). Illuminating Generalization in Deep Reinforcement Learning through Procedural Level Generation. *arXiv:1806.10729 [cs, stat]*.

Landauer, T. and Bjork, R. (1978). Optimum rehearsal patterns and name learning. /paper/Optimum-rehearsal-patterns-and-name-learning-Landauer-Bjork/120dca3871cc78991dcee4e6faa3484138039763.

Lehman, J. and Stanley, K. (2008). Exploiting Open-Endedness to Solve Problems Through the Search for Novelty. In *ALIFE*.

Lehman, J. and Stanley, K. O. (2011). Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218.

Mani, A., Mullainathan, S., Shafir, E., and Zhao, J. (2013). Poverty Impedes Cognitive Function. *Science*, 341(6149):976–980.

Mobahi, H., Farajtabar, M., and Bartlett, P. L. (2020). Self-Distillation Amplifies Regularization in Hilbert Space. *arXiv:2002.05715 [cs, stat]*.

Portelas, R., Colas, C., Hofmann, K., and Oudeyer, P.-Y. (2019). Teacher algorithms for curriculum learning of Deep RL in continuously parameterized environments. *arXiv:1910.07224 [cs, stat]*.

Portelas, R., Colas, C., Weng, L., Hofmann, K., and Oudeyer, P.-Y. (2020a). Automatic Curriculum Learning For Deep RL: A Short Survey. *arXiv:2003.04664 [cs, stat]*.

Portelas, R., Hofmann, K., and Oudeyer, P.-Y. (2020b). Trying AGAIN instead of Trying Longer: Prior Learning for Automatic Curriculum Learning. *arXiv:2004.03168 [cs, stat]*.

Racaniere, S., Lampinen, A. K., Santoro, A., Reichert, D. P., Firoiu, V., and Lillicrap, T. P. (2020). Automated curricula through setter-solver interactions. *arXiv:1909.12892 [cs, stat]*.

Roth, G. and Dicke, U. (2005). Evolution of the brain and intelligence. *Trends in Cognitive Sciences*, 9(5):250–257.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268.

Salge, C., Glackin, C., and Polani, D. (2013). Empowerment – an Introduction. *arXiv:1310.1863 [nlin]*.

Schmidhuber, J. (1991a). Curious model-building control systems. In *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*, pages 1458–1463 vol.2.

Schmidhuber, J. (1991b). Curious Model-Building Control Systems. In *Proceedings of the International Joint Conference on Neural Networks, Singapore*, volume 2, pages 1458–1463. IEEE press.

Schmidhuber, J. (1992). Learning Complex, Extended Sequences Using the Principle of History Compression. *Neural Computation*, 4(2):234–242.

Schmidhuber, J. (2002). Optimal Ordered Problem Solver. *arXiv:cs/0207097*.

Schmidhuber, J. (2012). POWERPLAY: Training an Increasingly General Problem Solver by Continually Searching for the Simplest Still Unsolvable Problem. *arXiv:1112.5309 [cs]*.

Schmidhuber, J. (2018). One big net for everything. *arXiv preprint arXiv:1802.08864*.

Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. (2020). Dynamics-Aware Unsupervised Discovery of Skills. *arXiv:1907.01657 [cs, stat]*.

Sukhbaatar, S., Lin, Z., Kostrikov, I., Synnaeve, G., Szlam, A., and Fergus, R. (2018). Intrinsic Motivation and Automatic Curricula via Asymmetric Self-Play. *arXiv:1703.05407 [cs]*.

Todorov, E., Erez, T., and Tassa, Y. (2012). MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033.

Wang, R., Lehman, J., Clune, J., and Stanley, K. O. (2019). Paired Open-Ended Trailblazer (POET): Endlessly Generating Increasingly Complex and Diverse Learning Environments and Their Solutions. *arXiv:1901.01753 [cs]*.

Weinshall, D., Cohen, G., and Amir, D. (2018). Curriculum Learning by Transfer Learning: Theory and Experiments with Deep Networks. *arXiv:1802.03796 [cs]*.

A DIAYN illustrated

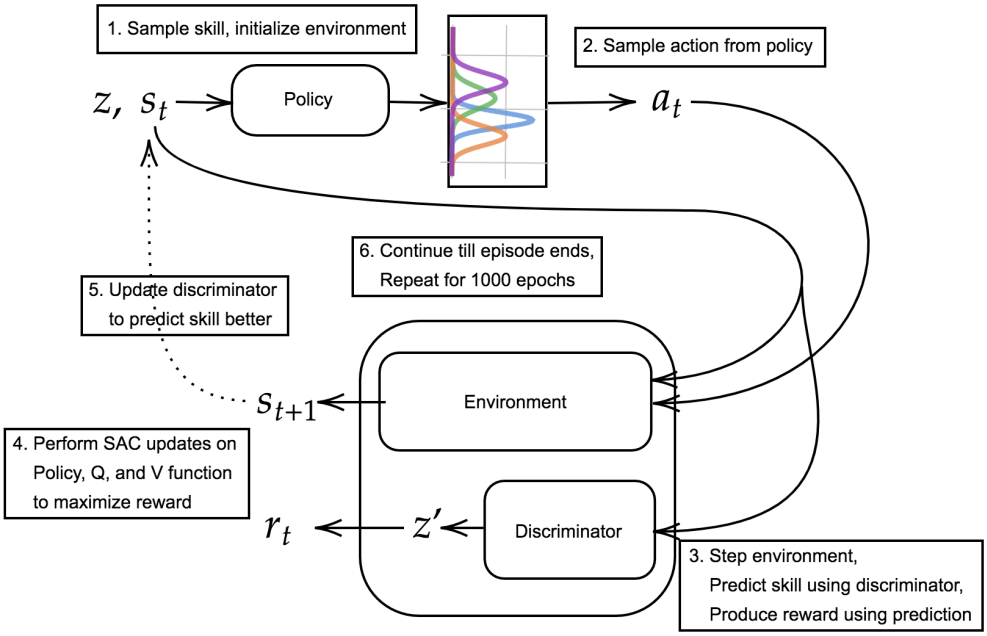

Figure 4: **The DIAYN algorithm:** The skill z is a one-hot 50-bit vector that the policy is conditioned on. The discriminator predicts the skill given the state. Essentially, the skill represents a discriminable partition of the state space. The policy is optimized to reach discriminable states.

The policy, the value functions Q and V and the discriminator are instantiated as 2 hidden layers, 300-wide ReLU networks. The policy outputs the parameters of a 4-component gaussian mixture model. The number of skills, represented by one-hot vectors, is fixed to 50. The skill vector is appended to the state vector and is used as input to the policy and value functions. The reward is given by

$$r(z, s_t) = \log q_\phi \left(z \mid s_t \right) - \log(1/50) \tag{1}$$

where q_ϕ is the discriminator and z is the skill.

B Experimental details

We use the hyperparameters given in Table 1 and Table 2 in all experiments.

Table 1: Common hyperparameters across all environments and algorithms

Hyperparameter	Value
no. of mixtures	4
batch size	128
discount	0.99
epoch length	1000
layer width	300
learning rate	0.0003
max path length	1000
num skills	50
seeds	1,2,3,4,5
τ (Polyak)	0.01

Table 2: Hyperparameters which are not the same for all environments and algorithms. Note that we train DIAYN for a much shorter period than was done in the original work (10k epochs). We only train until the discriminator reward saturates. We use a reward scale of 3.0 for ant, as done in the original DIAYN paper.

Hyperparameter	DIAYN	Retrain	Curriculum	Finetuning
num epochs	3k (hopper), 2k (ant), 1k (cheetah)	1k	1k	1k
α (Entropy)	0.1	1.0	1.0	1.0
switch epoch	-	-	500	-
reward scale	1.0	1.0	1.0	1.0 (hopper, cheetah), 3.0 (ant)
discriminator width	300	300	10-300 (ant, hopper), 20-300 (cheetah)	-

C Advantage of joint training in DIAYN

DIAYN performs joint training of the discriminator and the agent. This ensures that the skills are of "intermediate difficulty" for the agent as the discriminator is fit to the skill state-distribution of the agent. Hence the DIAYN agent is able to maximize the discriminator rewards easily. In retraining however, the discriminator is pre-trained and has a fixed difficulty for each skill. This makes it much harder for the agent to maximize the reward for some of the skills. Hence, we see in figure 5 (top) that the reward for all 50 generated skills is on average higher than in the plot below.

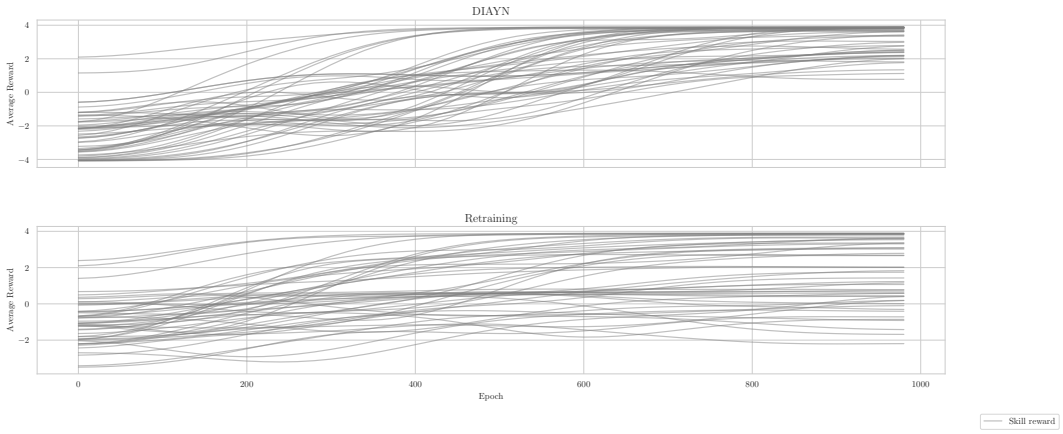

Figure 5: **Average skillwise discriminator reward:** Average reward obtained per time step, for each of the 50 skills, during DIAYN training (Top) and retraining (Bottom).

D Network width affects ease of optimization

In figure 6 we can see an example of how reducing the width of the discriminator can produce a decision boundary that is easier for the agent to cross. In the limiting case of a one-neuron-wide network, the decision boundary would be a simple hyperplane.

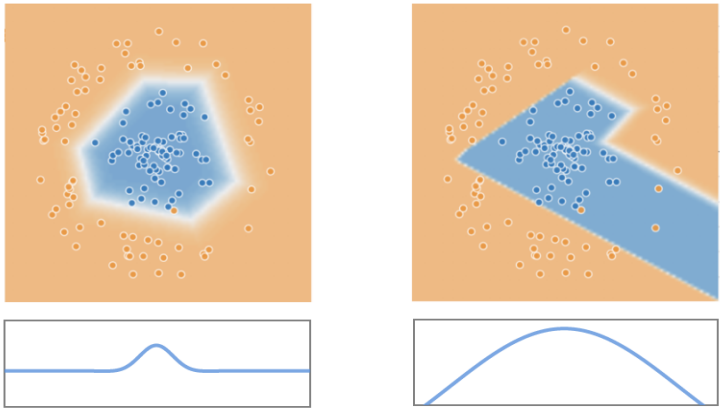

Figure 6: **Classification boundary comparison:** On the top left is the boundary produced by a 1 hidden layer, 3-wide ReLU network, on the top right is the boundary produced by a 6 hidden layers, 2-wide ReLU network. We see that the width is crucial for fitting the data perfectly. This is a consequence of the fact that less wide networks are less expressive, producing simpler functions, with lesser variability. Naively, using this effect we can modify a classifier-based reward function that looks like the one on the bottom left to look more like the one on the bottom right. The latter is easier to learn from, as the reward gradient is far more informative. An empirical visualization of this effect in CLAD can be found in figure 2.

E Entropy coefficient does not explain generalization

We note that the DIAYN algorithm uses an entropy scaling coefficient of 0.1, whereas finetuning is done using an entropy coefficient of 1.0 (Original DIAYN paper). Hence while retraining, we choose an entropy coefficient of 1.0. All else being equal, an agent with a high-entropy policy should generalize better. However, to ensure that the generalization advantage we get is purely from the retraining and not due to the entropy coefficient, we show a comparison of DIAYN and the retrained agent, trained using 0.1 on the left and 1.0 on the right in figure 7. Both plots correspond to the finetuning task using an entropy coefficient of 1.0.

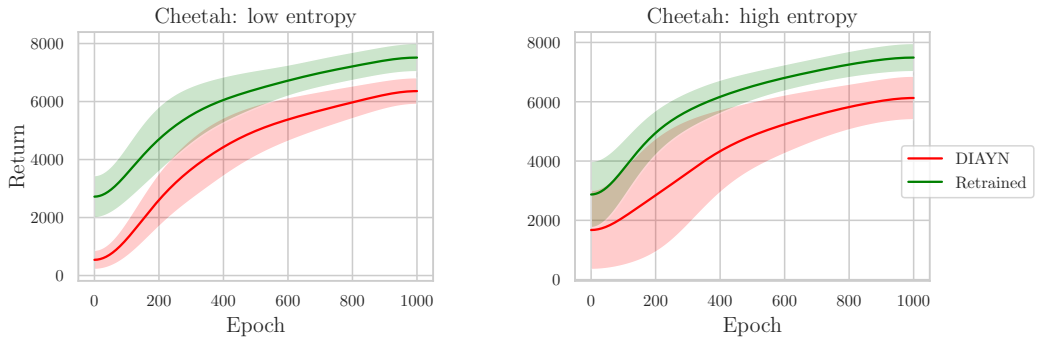

Figure 7: **Effects of entropy coefficient on generalization:** A comparison of the DIAYN (5 seed average) agent with the retrained agent with both agents previously trained using the same entropy coefficient of 0.1 on the left and 1.0 on the right. This shows that the generalization advantage is due to retraining and is less affected by the entropy coefficient.

F Skill-dependent advantages in retraining bias the results

Upon observation of the learned skills, one finds that agents retrained on the DIAYN discriminator learn running skills even when the original DIAYN agent doesn't. We analyze the discriminator reward for one such running skill in the cheetah environment in figure 8.

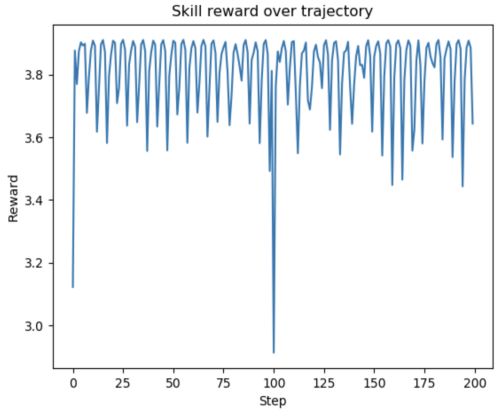

Figure 8: **Retrained agent learns unintended running skill:** Reward-per-time-step as defined by the DIAYN discriminator on the retrained agent over two episodes of 100 steps each in the cheetah environment. We see that the running skill learned by the agent shows oscillations suggesting it does not actually correspond to the intended discriminator skill.

We observe that the running skill can be achieved via oscillations around a fixed pose. Let's assume that one of the skills corresponds to the cheetah taking up a pose (Most DIAYN skills correspond to poses). Let's assume that the agent matches the pose perfectly apart for the hind leg being slightly off. It moves the hind leg to correct it, achieving the discriminator maximum, but consequently the front leg slips out of position. Now it corrects the front leg, only to get the hind leg out of position again. This oscillating behaviour can produce a running skill. Hence the retrained agent starts off with a very high reward on the running task (figure 7) in comparison with the DIAYN agent. We suspect that such effects mask the actual advantages of retraining and curricula, particularly in running tasks (figure 3). Therefore we test on a multi-task HERF benchmark that has flips and stands which cannot be achieved by such oscillations.

G Distilled discriminator based reward is easier to optimize

In figure 9, we see that the distilled discriminator based reward is easier to optimize. Further, the learning on the distilled discriminator directly transfers to the original discriminator, helping the agent achieve a higher score in cheetah and ant. This is not the case for hopper, but the blue curve catches up with the green curve upon further training. This suggests that the improvement in generalization is due to the ease of training rather than the final reward achieved on the discriminator.

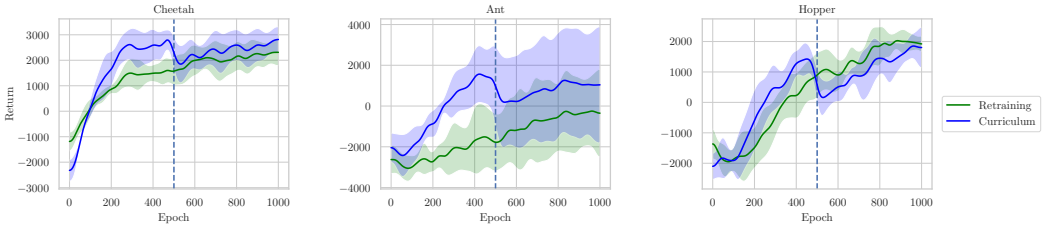

Figure 9: **Retraining vs curriculum:** Plot shows the discriminator based returns (average of a sampled subset of 50 skills) during retraining (green) and curriculum training (blue). The variations in the curve are due to the sampling of the skills. Dotted line represents the point when we switch from the distilled discriminator to the original discriminator in the curriculum training. As evidenced by the sharper rise in performance (blue) in all three environments, we see that the distilled discriminator based reward is much easier for the agent to maximize.

Table 3: **Final discriminator return:** We see that for all seeds, the training on the distilled discriminator for 1000 epochs on cheetah achieves a much higher return than on the original DIAYN discriminator. This shows that the distilled discriminator based reward is much easier to optimize.

Seed	Distilled discriminator	Discriminator
1	**3677**	2838
2	**3048**	1614
3	**3386**	2133
4	**3549**	2430
5	**2389**	1414
Average:	**3210**	2086

H Is distillation sufficient by itself?

Prior work on self-distillation (Furlanello et al., 2018) has shown that ease of learning itself can improve generalization. The distilled LRF is easier to learn from, so it is possible that this explains the generalization advantage and there is no need for a curriculum. Hence, we compare the generalization advantage obtained from the curriculum with that obtained purely from the distillation on the multi-task HERF benchmark in figure 10. We see that although the distillation itself helps, the curriculum does better. In table 4 the curriculum agent achieves a much higher score on the DIAYN discriminator than the agent trained only on the distilled discriminator.

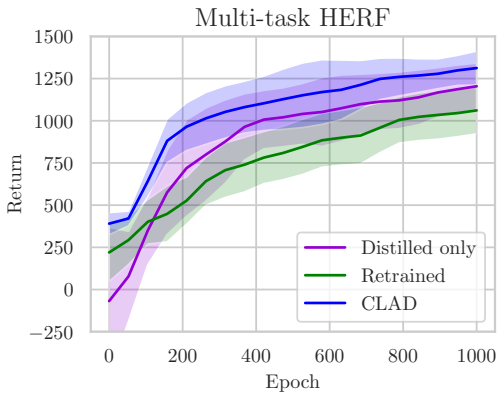

Figure 10: Comparison of the generalization advantage on the multi-task HERF benchmark in the cheetah environment. The curves plot the average episodic return over the 25 tasks. We can see that the agent trained only using the distilled discriminator (violet) does not perform as well as the curriculum trained agent (blue). This shows that it is actually the curriculum that helps and not just training on the distilled discriminator which only makes the rewards easier to maximize.

Table 4: **Discriminator return comparison:** Table below shows the return obtained on the DIAYN discriminator by training only on the distilled discriminator vs using the curriculum on the cheetah environment

Seed	Distilled discriminator only	Curriculum
1	2917	**3329**
2	1364	**2071**
3	2230	**2576**
4	2115	**2821**
5	1158	**1689**
Average:	1957	**2497**

I Multi-task HERF benchmark

As mentioned in appendix F, certain tasks may bias the results. Hence we constructed a multi-task Human Engineered Reward Function (HERF) benchmark consisting of 25 tasks as listed in the table below.

Table 5: Final returns achieved in the multi-task HERF benchmark

S.No	Task	Random	DIAYN	Retrain	CLAD
1	Forward run 8 m/s	406.92	542.7	2090.23	**3779.62**
2	Forward run 6 m/s	391.06	531.68	2065.69	**3774.72**
3	Forward run 4 m/s	307.92	545.45	2001.62	**3306.56**
4	Forward run 2 m/s	456.61	226.26	1173.1	**1299.4**
5	Backward run 2 m/s	1146.64	970.21	**1351.14**	1329.81
6	Backward run 4 m/s	3242.48	3133.82	**3292.25**	3260.17
7	Backward run 6 m/s	3445.94	3600.09	3772.64	**4039.64**
8	Backward run 8 m/s	3462.13	3603.9	3750.46	**4063.38**
9	Forward flip 8 rad/s	326.61	**1040.62**	509.48	438.51
10	Forward flip 6 rad/s	331.28	**965.81**	509.21	446.18
11	Forward flip 4 rad/s	319.82	**688.69**	446.37	406.83
12	Forward flip 2 rad/s	294.64	**444**	223.42	292.66
13	Backward flip 8 rad/s	-9.59	110.16	169.65	**170.24**
14	Backward flip 6 rad/s	-5.75	113.78	173.68	**174.1**
15	Backward flip 4 rad/s	-4.39	113.7	170.3	**170.67**
16	Backward flip 2 rad/s	-9.82	13.9	107.16	**130.08**
17	Stand 90°	1347.48	1328.36	1385.78	**1390.32**
18	Stand 60°	921.98	947.74	**950.3**	925.86
19	Stand 120°	1386.31	1355.31	1410.27	**1444.82**
20	Stand -90°	-41.44	**13.4**	-38.38	-19.78
21	Stand -60°	**-41.45**	-42.84	-44.02	-41.91
22	Frontfoot hop 90°	1307.76	1315.52	**1343**	1335.74
23	Frontfoot hop 60°	969.1	971.66	988.17	**1036.79**
24	Backfoot hop -90°	-23.11	**43.82**	-17.36	9.03
25	Backfoot hop -60°	-34.22	-40.59	-11.73	**5.02**
	Average	795.8	901.49	1110.9	**1326.74**

Experimental details We use the same training process as DIAYN (i.e. train on one randomly sampled skill every epoch). But since the DIAYN skills do not match the 25 tasks that we have defined, we perform a skill matching before training. We identify the best DIAYN skill for each task, and we switch the weights corresponding to the latent skill input in the network to activate that skill when learning that task. We design the reward functions as triangular hills centered at the target goal (position or velocity) with a reward of 0 at the origin. Hops use a combination of the Stand task reward function with the absolute vertical velocity added in. We use a reward scale of 1 for all tasks except flips, where we use 5. This is done because all agents fail to learn full-flips in single-task training of flips with a reward scale of 1.

The CLAD agent outperforms the other agents, on a majority of tasks except forward flips, where DIAYN consistently outperforms others. This seems to be a result of the fact that flips are difficult to learn when training on the frozen discriminator, but easier during joint training as done in DIAYN. We conclude that not all tasks benefit from the retraining or our curriculum.

J Multi-task HERF benchmark: single-task evaluation

We evaluate on some individual tasks from the multi-task HERF benchmark. This is done to disentangle inter-task dependencies that may arise in multi-task training. All tasks use a reward scale of 1 except flips, which use 5. This is necessary as all agents fail to learn with a lower reward scale. We suspect this is because the action penalty outweighs the task rewards for flips. In all tasks, the curriculum outperforms or matches the performance of the retrained agent. In flips, the DIAYN agent outpeforms the other agents since flipping directly corresponds to one of the learned DIAYN skills and the retrained/curriculum trained agents find it difficult to learn, particularly multiple consecutive flips. We hypothesize the random agent outperforms DIAYN as it and its derivatives, all learn to depend on the angular position, which needs to be ignored to learn continuous flips.

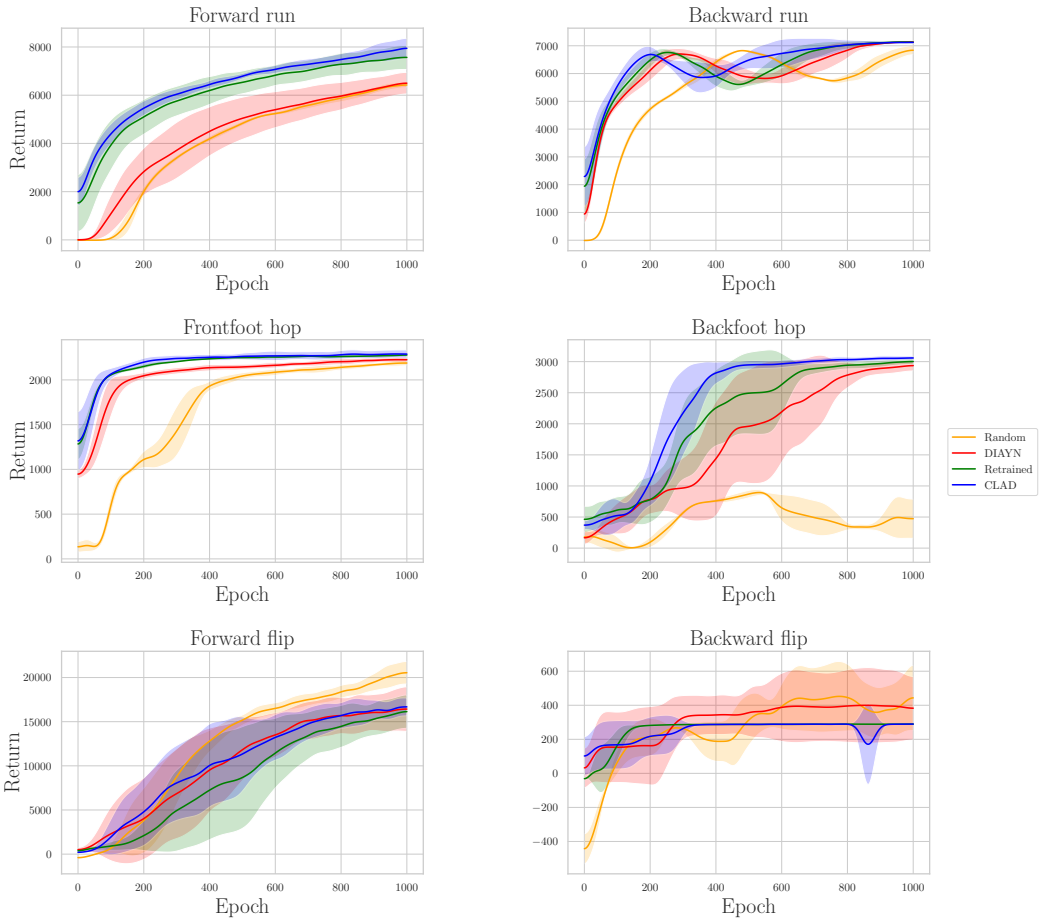

Figure 11: An individual comparison (5 seed averages) on 6 different tasks selected from the HERF benchmark set. In all tasks, the curriculum outperforms or matches the performance of the retrained agent.