A DIAYN illustrated

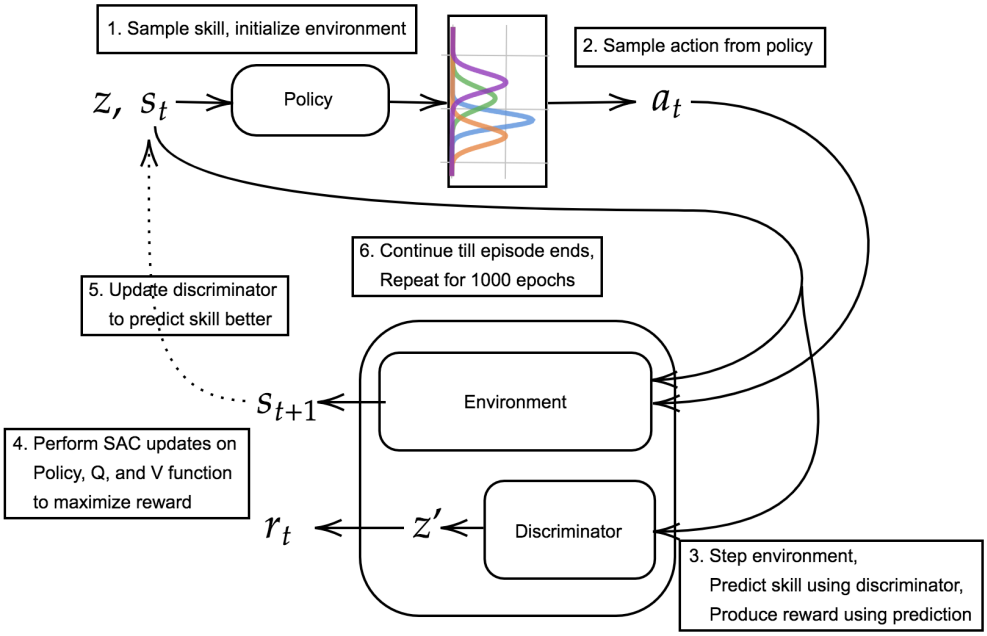

Figure 4: **The DIAYN algorithm:** The skill z is a one-hot 50-bit vector that the policy is conditioned on. The discriminator predicts the skill given the state. Essentially, the skill represents a discriminable partition of the state space. The policy is optimized to reach discriminable states.

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.

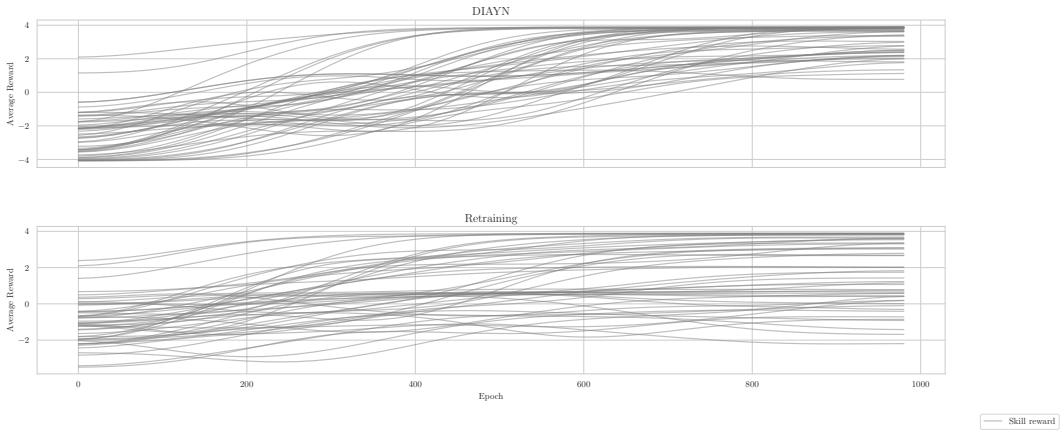

Figure 5: **Average skillwise discriminator reward:** Average reward obtained per time step, for each of the 50 skills, during DIAYN training (Top) and retraining (Bottom).

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.

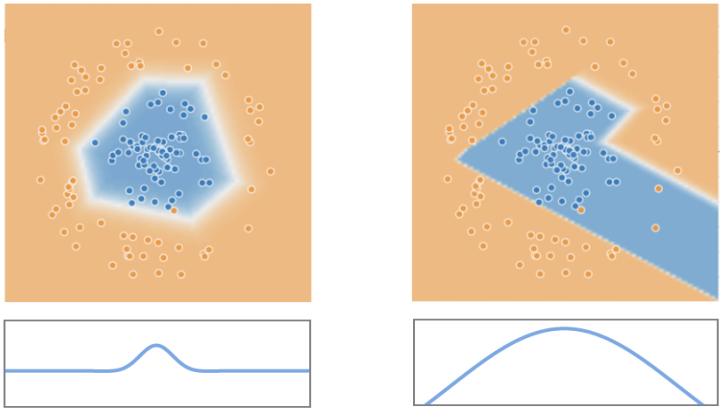

Figure 6: **Classification boundary comparison:** On the top left is the boundary produced by a 1 hidden layer, 3-wide ReLU network, on the top right is the boundary produced by a 6 hidden layers, 2-wide ReLU network. We see that the width is crucial for fitting the data perfectly. This is a consequence 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.

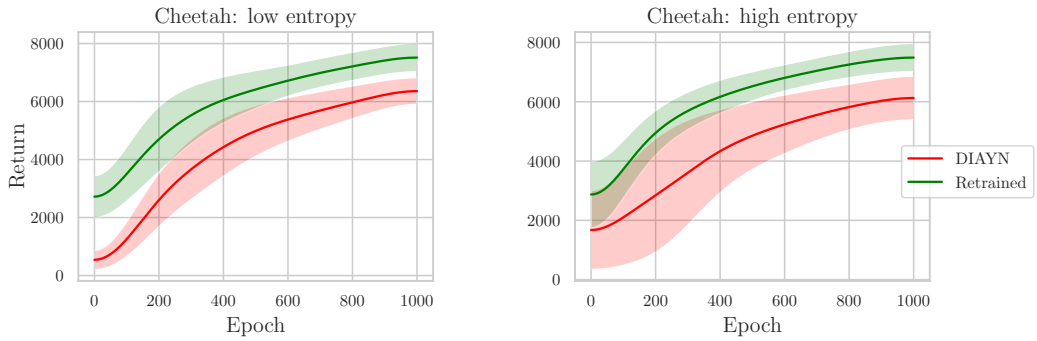

Figure 7: **Effects of entropy coefficient on generalization:** A comparison of the DIAYN (5 seed average) agent with the retrained agent with both agents previously trained using the same entropy coefficient of 0.1 on the left and 1.0 on the right. This shows that the generalization advantage is due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

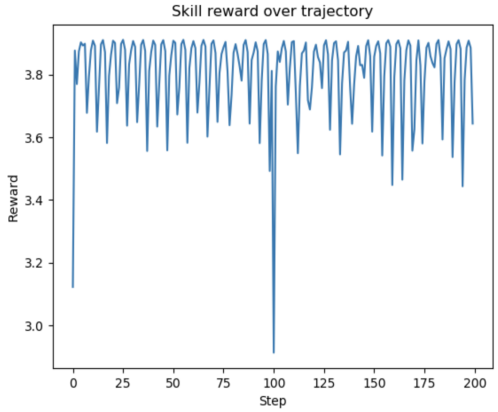

Figure 8: **Retrained agent learns unintended running skill:** Reward-per-time-step as defined by the DIAYN discriminator on the retrained agent over two episodes of 100 steps each in the cheetah environment. We see that the running skill learned by the agent shows oscillations suggesting it does not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

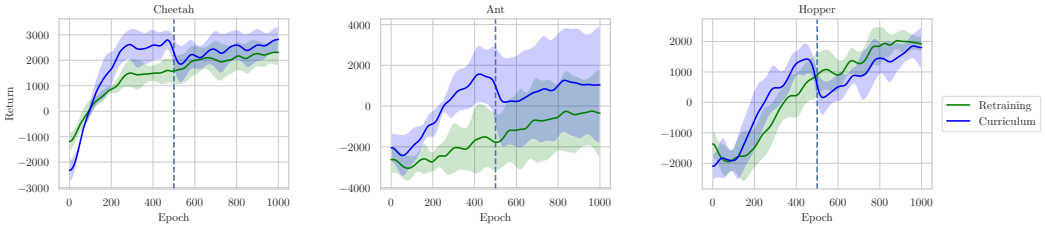

Figure 9: **Retraining vs curriculum:** Plot shows the discriminator based returns (average of a sampled subset of 50 skills) during retraining (green) and curriculum training (blue). The variations in the curve are due to the sampling of the skills. Dotted line represents the point when we switch from 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.

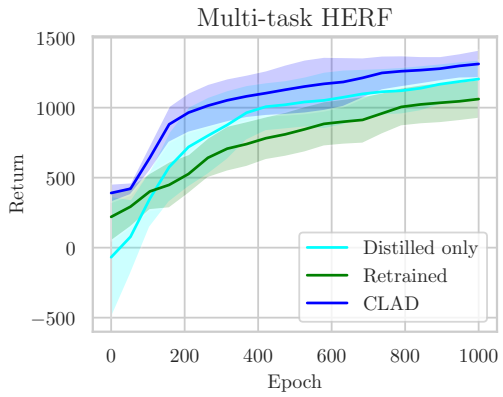

Figure 10: Comparison of the generalization advantage on the multi-task HERF benchmark in the cheetah environment. The curves plot the average episodic return over the 25 tasks. We can see that the agent trained only using the distilled discriminator (cyan) does not perform as well as the curriculum 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.

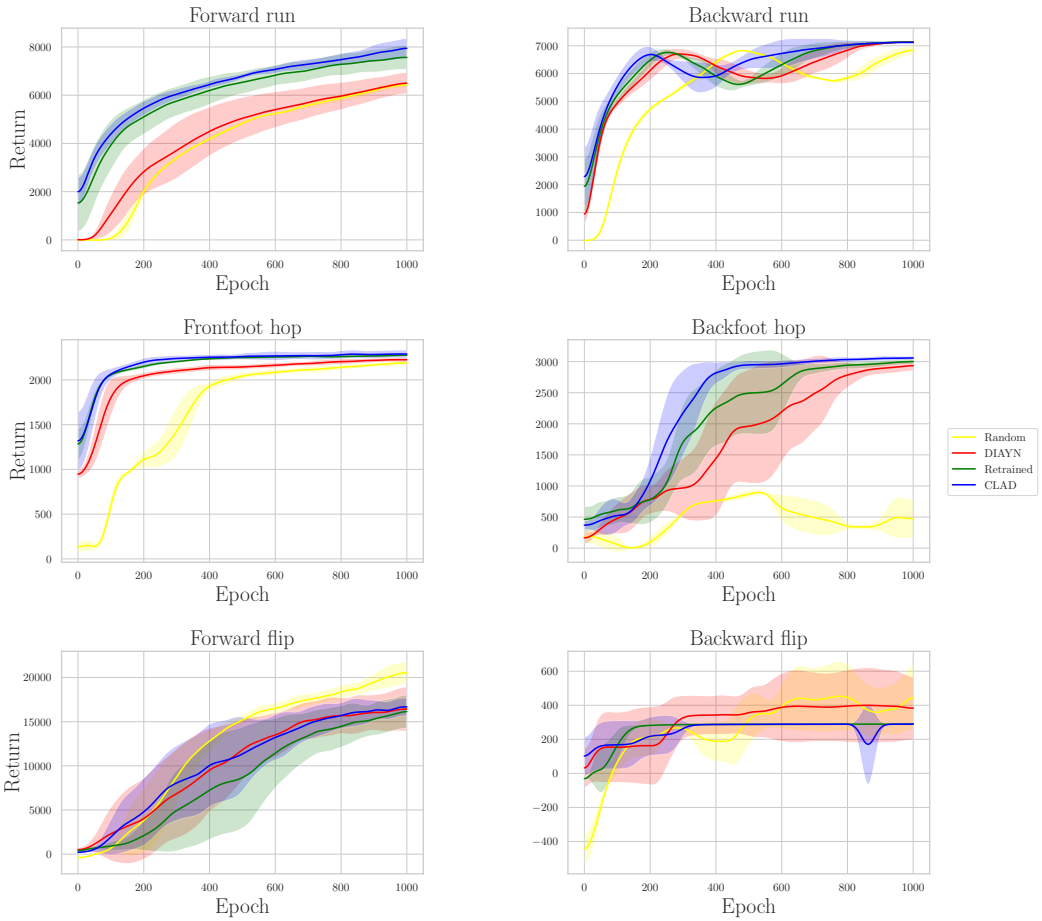

Figure 11: An individual comparison (5 seed averages) on 6 different tasks selected from the HERF benchmark set. In all tasks, the curriculum outperforms or matches the performance of the retrained agent.