Appendix

A. Additional Derivations

Below is the full derivation of the objective used to motivate low-level discriminative imitation, taking inspiration from other work based on information theoretic objectives (Eysenbach et al., 2018). We start by minimizing the mutual information between morphology, M and behavior, $\mathcal{T}_t$. I denotes mutual information and H denotes entropy.

$$\begin{aligned}
&\min_{\theta_B^{lo}} I(M;\mathcal{T}_t) = \max_{\theta_B^{lo}} -I(M;\mathcal{T}_t) \\
&= \max_{\theta_B^{lo}} -(H(M) - H(M|\mathcal{T}_t)) \\
&= \max_{\theta_B^{lo}} H(M|\mathcal{T}_t) - H(M) \\
&= \max_{\theta_B^{lo}} H(M|\mathcal{T}_t) \\
&= \max_{\theta_B^{lo}} \mathbb{E}[-\log p(M|\mathcal{T}_t)] \\
&\geq \max_{\theta_B^{lo}} \mathbb{E}[-\log q_\phi(M|\mathcal{T}_t)]
\end{aligned}$$

As per the above derivation we can encourage similar behavior across agents by maximizing the entropy of the morphology given a behavior. In the fourth step we assume the distribution over morphologies is uniform, and subsequently the second term is a constant that can be omitted from the optimization. The final step applies the variational lower bound (Barber & Agakov, 2004). In practice, we try to align behavior when reaching the same goal. This can be accomplished by conditioning the original objective on the goal, $\min I(M;\mathcal{T}_t|g)$. Propagating this change through the derivation results in optimizing a goal conditioned discriminator $q_\phi(M|\mathcal{T}_t, g)$. In practice, we do this by using $(g - s_t, g - s_{t+1})$ as the discriminator input.

B. Complete Algorithm

Algorithm 2 provides the complete algorithm for using both of our purposed methods of imitation, discriminative low-level and KL-regularized high-level, to transfer a policy from agent A to agent B. Though this algorithm provides an overall training flow, note that experiments we ablate the two imitation components separately to better understand the performance contributions of each. More experimentation would be required to understand how both components interact in sequence. In general, we find that high-level KL-regularized finetuning is better for gaining performance on a specific task, whereas low-level discriminative imitation is better for boosting performance across a suite of tasks.

C. Environment Details

Below are more complete specifications of the environments used in experiments.

C.1. Navigation

For all navigation agents, the low-level reward is given by the weighted distance traveled towards the goal, with an action penalty term.

$$r_{lo} = \frac{(s_t^{hi} - s_{t-1}^{hi}) \cdot (g_t - s_{t-1}^{hi})}{||g_t - s_{t-1}^{hi}||_2} - \lambda ||a_t||_2^2$$

High-level actions are taken once every 32 steps, except on the quadruped agent where it is performed every 64. The high-level goal space is defined to be the desired change in x, y position of the agent's center, limited by a distance of four meters in either direction.

Agents: All agents observe joint positions (*qpos*), velocities (*qvel*), and the vector to the next sub-goal. All agents besides the point mass additionally observe contact forces. All agents use torque control.

- *Point Mass*: A point mass agent whose actions are forces in the cardinal directions.
- *Gym Ant*: This is the Open AI Gym *Ant* agent with its gear reduced from 150 to 125. Note that this is less modification than the *Ant* agent in HiRO (Nachum et al., 2018a).

Algorithm 2 Complete Transfer

```
  **Input:** Agents $A$ and $B$ and a goal distribution $\mathbb{G}$.
  **Initialize:** Learnable parameters $\theta_A^{lo}, \theta_A^{hi}, \theta_B^{lo}, \theta_B^{hi}$ and $\phi$ for $q_\phi$.
  $\theta_A^{lo} \leftarrow$ policyOptimizerLow($\mathcal{T}_A, \theta_A^{lo}, \mathbb{G}$)
  $\theta_A^{hi} \leftarrow$ policyOptimizerHigh($\mathcal{T}_A, \pi_A^{lo}$)
  for $i$=1,2,..$N_{\text{low}}$ do
    sample goal $g \sim \mathbb{G}$
    for $j$=1,2,..$T$ do
      collect experience $(s_t^{lo}, a_t^{lo}, s_{t+1}^{lo}, r_t^{lo})$ for $\mathcal{T}_B$ from $\theta_B^{lo}$
    end for
    for $j$=1,2,...$M_{\text{policy}}$ do
      update $r_t^{lo}$ according to eq 3
      $\theta_B^{lo} \leftarrow$ policyOptimizerLow($\{(s_t^{lo}, a_t^{lo}, s_{t+1}^{lo}, r_t^{lo})\}, \theta_B^{lo}$)
    end for
    for $j$=1,2,...$M_{\text{discrim}}$ do
      $\phi \leftarrow$ discrimOptimizer($\mathcal{T}_A, \mathcal{T}_B, \phi$)
    end for
  end for
  $\theta_B^{hi} \leftarrow \theta_A^{hi}$
  for $i$=1,2,... $N_{\text{high}}$ do
    collect experience $(s_t^{hi}, a_t^{hi}, s_{t+k}^{hi}, r_t^{hi})$ for $\mathcal{T}_B$ from $\theta_B^{hi}$
    grad$_{\text{RL}} \leftarrow$ policyOptimizerHigh($\{(s_t^{hi}, a_t^{hi}, s_{t+k}^{hi}, r_t^{hi})\}, \theta_B^{hi}$)
    grad$_B^{hi} \leftarrow$ grad$_{\text{RL}} + \alpha \nabla_{\theta_B^{hi}}$KL$(\pi_B^{hi}(a_B^{hi}|s_B^{hi})||\pi_A^{hi}(a_A^{hi}|s_B^{hi}))$
    $\theta_B^{hi} \leftarrow$ update($\theta_B^{hi}$, grad$_B^{hi}$)
  end for
```

- *3 Leg Ant*: This agent is identical to the regular ant, expect one of its legs is frozen in place.
- *2 Leg Ant*: Again identical to the Ant, expect two diagonally opposed legs are frozen in place.
- *DM Control Quadruped*: The quadruped agent is similar to the Gym Ant, expect it has an extra ankle joint on each of it's legs, making controlling it different. We do not use the same control scheme as in DM Control, and instead give it the same observations as the *Ant* agent.

Tasks:

- *Way Point Navigation*: The agent is tasked with navigating through a plane and reaching specific waypoints. As soon as the agent reaches one waypoint, another waypoint is randomly placed. The agent receives a reward for its $L2$ distance from the waypoint, and a sparse reward of 100 upon reaching the waypoint. The observation space is given by the agent's current position and the position of the waypoint. The high-level policy is trained with a horizon of 50 high-level steps.
- *Maze*: The agent must navigate through a 'U' shaped maze and reach the end. The agent only receives a sparse reward of 1000 upon reaching its final goal. During training, final goal locations are randomly sampled uniformly from the six ”blocks” of the maze path, while in evaluation the final goal is always the end of the maze. The observation space is given by just the agent's current (x, y) position and the position of the final goal.
- *Steps*: The agent has to navigate through a similarly shaped structure to that of the *Maze*, although only half the size. The height of the taller step is 0.3125 meters, while the height of the shorter step is 0.15625 meters. When the Ant agent is used for the step environment, it is given 16 rangefinder sensors and it's low level is pretrained on an environment with randomly placed steps.

C.2. Manipulation

For all manipulation tasks, low-level rewards are given by $L2$ distance to the selected sub-goal and an additional sparse reward.

$$r_{lo} = -||g_t - s_t^{hi}||_2 - \lambda||a_t||_2^2 + \gamma 1\{g_t - f(s_t) < \epsilon\}$$

High-level planning is performed every 35 steps. Again, all agents use torque control.

Agents:

- *Point Mass*: Identical to the previous point mass, just scaled to fit the environment.
- *2-Link Arm*: This is the standard reacher from the Open AI Gym set of environments, with end effector collisions enabled.
- *3-Link Arm*: A modified version of the standard 2-Link reacher with one extra degree of freedom. Each link is approximately one third the length of the arm.
- *4-Link Arm*: A modified version of the standard 2-Link arm, created by splitting each link evenly into two more links.

We found that the ant agents with fewer legs tended to be more stable and fell over less.

Tasks:

- *Block Push*: The arm agent has to push a block across the environment to a target end position. We test on variations of difficulty based on block position. Here, high-level observations include the position of the end effector and the position and velocity of the block. high-level rewards correspond to negative $L2$ distance of the block to its goal position and a sparse reward of 200 for solving the task. The high-level goal space is defined to be the desired change in the x, y position of the agent's end effector, limited by a distance of 0.07 meters in either direction. We have two different variants of the block push task, *Block Push 1*, where the block must be pushed just horizontally, and *Block Push 2*, where the block must be pushed a shorter distance, but horizontally and vertically.
- *Peg Insertion:* The agent now has a peg attached to it's end effector that it must insert into a hole. high-level observations include the position of the tip of the peg and the position of the end effector. high-level rewards correspond to negative $L2$ distance from the final desired insertion point and a sparse reward of 50 for solving the task. For peg insertion, the high-level goal space is given from the end of peg.

D. Training Details

When training low-level policies, we only reset environment occasionally after selecting a new low-level goal to allow the agent to learn how to perform well in long-horizon settings. Low level policies are trained over longer horizons than the exact number of steps in between high level actions. For high-level training on top of pre-trained low-levels, we collect samples only when the high-level policy sets a new sub-goal. We include hyper-parameters for all low level training in Table 4 and hyperparameters for all high level training in Table 5.

When training the discriminator for low level imitation, we anneal the learning rate linearly from its initial value to zero over the first "stop" fraction of training timesteps. This allows the agent to learn against an increasingly fixed target. Additionally, we anneal the discriminator weight in the reward function from it's initial value to 0.1 linearly over the first 90% of training timesteps. Full parameters for the discriminators can be found in Table 6. Additionally, we tested online and offline data collection. In offline data collection, transitions are randomly sampled from agent A's low level policy. In online data collection, we align the goals of the two agents, such that we collect transitions of agent A reaching goal g when agent B is attempting to reach the same g. The results presented in the main paper body are exclusively from offline data collection.

For KL-regularized fine-tuning, we use the same parameters across almost all experiments. We add the KL-divergence between Agent B's policy and Agent A's policy at every timestep. For the Waypoint task and all manipulation tasks, we use a KL weight coefficient of 1 in the loss, a learning rate of 0.01, and linearly anneal the weight of the KL loss to zero during the first 50% of training. For the Maze Task, we lowered the learning rate to 0.001 and the KL loss coefficient to 0.01. We performed a search over learning rates for regular fine-tuning, and found the original learning rate of the policy tended to perform best and as such used it for comparison.

Agent	Timesteps	Learning Rate	Batch Size	Layers	Horizon	Reset Prob	Buffer Size	DM
PM (Nav)	200000	0.0003	64	64 64	35	0.1	200000	4
Ant(s)	2500000	0.0008	100	400 300	100	0.1	1000000	4
Quadruped	2500000	0.0008	100	400 300	150	0.1	1000000	4
Manipulation	1200000	0.0003	100	128 96	45	0.25	250000	0.07

Table 4. Hyperparameters for low level policy training. “DM” stands for goal delta max, or the size of the goal space in each dimension sampled from during training.

Task	Timesteps	Learning Rate	Batch Size	Layers	Horizon	Buffer Size
Waypoint	200000	0.0003	64	64 64	50	50000
Maze	400000	0.0003	64	64 64	100	50000
Block Push	500000	0.0003	64	64 64	60	50000
Insert	500000	0.0003	64	64 64	50	50000

Table 5. Hyperparameters for high level policy training.

E. Extended Zero-shot Results

In our navigation experiments we also considered an additional agent, the *Two-Leg* Ant. *Waypoint* results for the *Two-Legged Ant* can be found in Table 7 which contains complete zero-shot results with more precision. *Maze* results can be found in Table 8.

Zero-shot results for the *2-Link* arm were withheld from Table 1 for consistency with the *PegInsert* task, which the two*2-Link* arm was unable to complete due to its limited range of motion. Zero-shot results for the *2-Link* on *BlockPush* can be found in Table 9.

F. Extended Discriminative Imitation Results

In our initial experiments we considered both an online and offline data collection scheme used for training the discriminator. In the online version of data collection, roll-outs are collected from each agent running on the same goal g, ensuring the discriminator is trained on the same goals from both agents. Initial experiments showed that offline data collection, as described in section 4.2 was as good or better than online data collection in most cases. A possible explanation is that online data collection made the discriminator's task too easy. In the main body of the paper, we only report results from offline data collection. Here, Table 10 and Table 11 contain results from all the online vs. offline comparisons we ran.

G. Extended KL-regularized Finetuning Results

We ran finetuning experiments on the *Waypoint* task that were not included in the main body of the paper due to their similarity to the included curve for the *Ant* agent. Finetuning results for the *3-Leg Ant* and the *Quadruped* on the waypoint task are included in Figure 8.

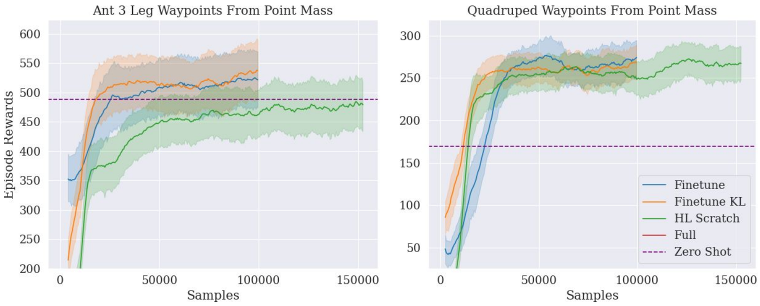

Figure 8. Comparison of performance finetuning from *PointMass* for *Ant Waypoint*, and *Quadruped Waypoint* respectively.

Agent A	Learning Rate	Batch Size	Layers	Update Freq	Weight	Stop
Nav PM	0.0002	64	42 42	8	0.3	0.5
PM, 2Link	0.0003	64	42 42	8	0.4	0.5
3Link	0.0005	64	42 42	8	0.4	0.5

Table 6. Hyperparameters for discriminator training.

	Point Mass High	Ant High	Ant3 High	Ant2 High	Quadruped high
Point Mass Low	1021.49 ± 43.25	602.56 ± 33.82	716.61 ± 58.12	593.18 ± 60.09	576.65 ± 69.6
Ant Low	482.72 ± 38.96	476.42 ± 19.44	472.85 ± 50.68	417.59 ± 27.48	406.96 ± 35.63
Ant3 Low	488.62 ± 74.35	483.59 ± 71.67	499.19 ± 64.99	471.24 ± 65.42	432.29 ± 64.59
Ant2 Low	353.56 ± 39.33	371.11 ± 27.15	388.38 ± 31.56	420.81 ± 31.24	373.99 ± 34.52
Quadruped Low	169.43 ± 33.36	182.33 ± 21.55	219.57 ± 26.61	267.12 ± 18.95	257.13 ± 18.83

Table 7. Zero-Shot transfer for the way-point navigation task.

Maze Task	PM High End	Ant High End	PM High Sample	Ant High Sample
Ant2 Low	$.74 \pm .17$	$.14 \pm .13$	$.87 \pm .09$	$.60 \pm .05$

Table 8. Zero-shot results for *Two-Legged Ant* on *Maze*

Block Push 1	PM HL	2Link HL	3Link HL	4Link HL
Ant2 Low	$.36 \pm .14$	$.97 \pm .02$	$.22 \pm .07$	$.39 \pm .11$

Table 9. Zero-shot results for *2-Link Arm* on *Block Push 1*

Task	Waypoint	Maze Sampled	Maze End
Transfer	PM→Ant	PM→Ant	PM→Ant
Zero-Shot	482.72 ± 38.96	0.3 ± 0.08	0.65 ± 0.04
Discrim Online	467.22 ± 20.61	0.37 ± 0.1	0.63 ± 0.04
Discrim Offline	546.06 ± 14.78	0.55 ± 0.13	0.72 ± 0.03

Table 10. Discriminative imitation zero-shot results for the Ant.

Task	Block Push 1				Block Push 2			
Transfer	PM→3Link	2Link→3Link	2Link→4Link	3Link→4Link	PM→3Link	2Link→3Link	2Link→4Link	3Link→4Link
Zero-Shot	$0.17 \pm .08$	$0.2 \pm .09$	$0.07 \pm .04$	$0.1 \pm .05$	$0.24 \pm .12$	$0.61 \pm .16$	$0.19 \pm .12$	$0.39 \pm .16$
Discrim On	$0.34 \pm .1$	$0.43 \pm .15$	$0.17 \pm .08$	$0.23 \pm .13$	$0.43 \pm .12$	$0.42 \pm .11$	$0.46 \pm .13$	$0.44 \pm .14$
Discrim Off	$0.35 \pm .13$	$0.49 \pm .12$	$0.28 \pm .14$	$0.15 \pm .04$	$0.43 \pm .15$	$0.42 \pm .11$	$0.41 \pm .16$	$0.41 \pm .15$

Table 11. Discriminative imitation zero-shot results for various manipulation configurations.

H. Resources

Our code can be found at `https://github.com/jhejna/hierarchical_morphology_transfer` and videos depicting results of our experiments can be found at `https://sites.google.com/berkeley.edu/morphology-transfer`.