# Skill Tuning in Pretrained Skill-Conditioned Policies
# CS 224R Course Project

**Robathan Harries**

## Abstract

Reward-free pretraining has emerged as a promising avenue for increased sample efficiency in task-specific learning, by reusing a pretrained policy which covers a large variety of distinct behaviors. Pretrained policies can encode information about the dynamics of an environment irrespective of specific reward functions, and can allow task-specific learning to avoid 'rediscovering the wheel'.

One recent method, Contrastive Intrinsic Control (Laskin et al., 2022), maximizes the richness of reward-free pretraining by mapping a 64-dimensional continuous skill space to maximally distinct behaviors. By learning continuous skills as opposed to a discrete set of skills, it is meant to yield a rich parametric space of distinct behaviors, maximizing coverage for downstream tasks. However, the default task adaptation method for CIC is to choose a single fixed skill vector, then fine-tune the pretrained policy to maximize task returns. Although this leads to fast task-specific adaptation, it requires changing the skills learned by the policy, rather than using them directly.

This project investigates the pretrained skill behaviors directly, testing how well they can be utilized to solve downstream tasks without any fine-tuning of the pretrained skill-conditioned policy. First, we use DDPG actor gradients to train a single skill to best solve a simple Mujoco Walker task, and find that no single skill encompasses a behavior to successfully solve this simple walking task.

Next, we extend single-skill fine-tuning with hierarchical RL, tuning a finite vocabulary of skill vectors alongside a skill selection policy, allowing different skills to specialize for different situations encountered in the task. Extending the Skill Tuning method to this hierarchical format required additions of entropy regularization terms, as well as sanity checks to ensure the skill selection policy is not collapsing to a single skill choice. This method can solve the Walking task, although it does so far slower than the default Policy Tuning used by CIC.

Finally, we test whether any of the previous skill-tuning methods can helpfully augment the default CIC Policy Tuning approach, but get mixed results. For the most part, jointly training skills alongside the skill-conditioned action policy performs comparably with the default method of tuning the policy alone. However, there are small indications that for some skill vocabulary sizes, namely 1 and 8, joint skill and policy tuning can speed up task adaptation by providing multiple avenues for updating behavior.

## 1. Motivation

Reward-free pretraining is a promising avenue for increasing the sample efficiency of task-specific reinforcement learning by using a pretrained task-general policy, which has discovered a set of skills which are generally useful for all tasks in a particular environment. These pretrained behaviors significantly speed up the exploration process in task-specific learning, by providing useful skill primitives or policy initializations which have large coverage over the state space. With skill-discovery pretraining, an agent learning a new task doesn't need to reinvent the wheel by flailing about randomly. By sharing aspects of the reinforcement learning process across all tasks in an environment, task-specific reinforcement learning can focus on finding the specific reward function that differentiates one task from another.

Contrastive Intrinsic Control (Laskin et al., 2022) is a competence-based reward-free pretraining algorithm which learns a high-dimensional continuous latent skill space, where different skill vectors map to distinct behaviors. Qualitatively, this large, continuous learned skill space exhibits many distinct modes of dynamic movement, but when adapting to downstream tasks, the method performs a minimal search through this space, and focuses primarily on fine-tuning the pretrained policy directly. Because of this, the diversity and coverage of the space of pretrained skill behaviors themselves is unclear.

This project utilizes the continuous nature of CIC skill vectors to perform gradient descent on skills directly. In this direction, the project tests 3 related methods. (1) Skill Tuning updates a single skill vector without tuning the policy directly, providing insights into the diversity and coverage of pretrained skill behaviors learned by CIC. (2) Skill Vocab Tuning extends Skill Tuning with a Hierarchical RL component, tuning a finite vocabulary of skill vectors while simultaneously training a skill selection policy that chooses individual skills from the vocabulary at each timestep. (3) Finally, hybrid methods (Skill + Policy Tuning, Skill Vocab + Policy Tuning) simply test how well each of the previous methods perform as augmentations to the original policy fine-tuning approach.

## 2. Related Work

### 2.1. Contrastive Intrinsic Control

Contrastive Intrinsic Control (Laskin et al., 2022) is an competence-based unsupervised skill-discovery algorithm that uses contrastive learning methods to maximize the mutual information between latent skill vectors and the state transitions of a skill-conditioned policy.

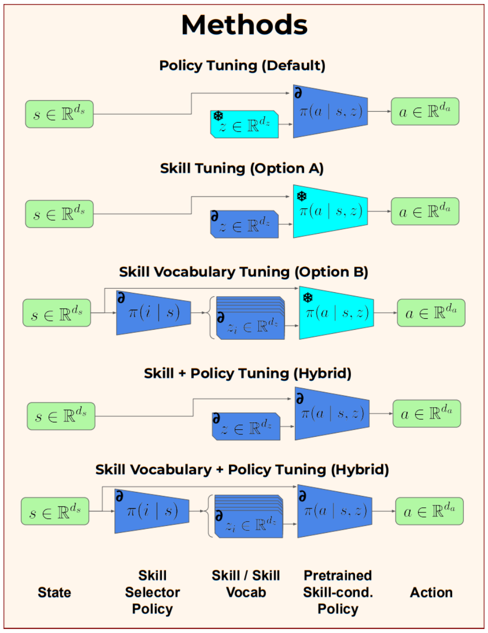

*Figure 1.* Actor architectures for downstream task adaptation methods tested in this project, compared to the default method used in CIC. Light-blue modules indicate that the relevant parameters are frozen during training. The actor is trained using DDPG actor-critic loss for all methods.

#### 2.1.1. CIC Pretraining

During pretraining, CIC trains a skill encoder and a state transition encoder, using Noise Contrastive Estimation (NCE) Loss to maximize the relative similarity between matching pairs of skill, state-transition compared to a random batch of unmatched pairs of skill, state-transition. These encoder training maximizes this objective, where $g_{\psi_1}$ and $g_{\psi_2}$ are the encoders for the state-transition $\tau_i$ and skill vector $z_i$, respectively:

$$F_{NCE}(\tau) = \frac{g_{\psi_1}(\tau_i)^\top g_{\psi_2}(z_i)}{\|g_{\psi_1}(\tau_i)\| \|g_{\psi_2}(z_i)\| T} - \log \frac{1}{N} \sum_{j=1}^{N} \exp\left( \frac{g_{\psi_1}(\tau_j)^\top g_{\psi_2}(z_i)}{\|g_{\psi_1}(\tau_j)\| \|g_{\psi_2}(z_i)\| T} \right)$$

At the same time, the skill-conditioned policy is trained using DDPG actor critic to maximize an intrinsic reward based on an APT particle entropy estimate, which measures the difference between the current state-transition embedding and the $k$ nearest state-transition embeddings in the replay buffer. The intrinsic entropy reward has the following form, where $h_i$ is the embedding for state-transition $\tau_i$:

$$\mathcal{H}_{particle}(\tau) \propto \frac{1}{N_k} \sum_{h_i^\star \in N_k}^{N_k} \log \|h_i - h_i^\star\|$$

### 2.1.2. CIC Downstream Task Adaptation

To adapt to downstream tasks, CIC first spends 4,000 initial timesteps trying out a set of candidate skill vectors, saving the returns experienced for each one, then selecting the best-performing skill vector and freezing it. Following this, the algorithm fine-tunes the action policy conditioned on this frozen skill vector, using the same DDPG actor-critic algorithm as during pretraining. This generally achieves good task adaptation performance after 100,000 steps.

The set of candidate skills are evenly-spaced along the diagonal of the skill space, from $< 0, ..., 0 >$ to $< 1, ..., 1 >$, with 40 skills tested for 100 timesteps each. During DDPG actor-critic training, both the actor and the critic modules are initialized from the pretrained policy, although both are updated during fine-tuning.

### 2.2. Context Optimization

The baseline method for this project, Skill Tuning, draws inspiration from Context Optimization (Zhou et al., 2021), a method for few-shot image recognition using pretrained Vision-Language Models. A pretrained Vision-Language Model computes the similarity encodes text and images into a shared latent space, and can perform zero-shot image classification by comparing the similarity of a given image embedding to the class label embeddings for each class.

To extend this zero-shot classification paradigm to a few-shot learning setting, CoOp trains a language prompt (in continuous token embeddings) that is prepended to each class label input to minimize the classification loss on few-shot examples. These continuous prompt embeddings are trained using gradients propagated through the pretrained model.

## 3. Technical Approach

An overview of all methods, including the default CIC Policy Tuning, is shown in Figure 1.

### 3.1. Skill Tuning

The Skill Tuning method requires very little adjustment compared to the default Policy Tuning method used by CIC. The first 4,000 steps of training are still used to explore 40 different possible skills along the diagonal of the skill space from $< 0, ..., 0 >$ to $< 1, ..., 1 >$. The best-performing skill is set as the initialization for the skill vector, and during the next 96,000 training steps, the same DDPG actor-critic algorithm is used to update the skill vector and the critic module, leaving the pretrained actor policy unchanged.

The actor loss function has this form:

$$\mathcal{L}_{\text{actor}}(z) = \mathbb{E}_s\left[-Q^\phi\left(s, \pi^\theta(s,z), z_c\right)\right]$$

Like the default CIC Policy Tuning, both the actor and critic modules are initialized from the pretrained modules, both of which are skill-conditioned. The critic is passed a frozen dummy skill vector $v_c =< 0.5, ..., 0.5 >$, to avoid training the skill vector based off the pretrained state of the critic, especially as the critic is further updated based on samples from a single skill, making its skill gradients $(\frac{\partial}{\partial z} Q^\phi(s,a,z))$ particularly inconsistent with the actual task.

### 3.2. Skill Vocab Tuning

Skill Vocab Tuning extends Skill Tuning to tune a finite vocabulary of skill vectors simultaneously. At each timestep, a skill selection policy chooses which skill to enact from this finite set. Theoretically, the skill selection policy allows skill tuning to prioritize skill tuning for different circumstances, which may be helpful for learning stabilization procedures alongside consistent locomotion behaviors. By learning skill vectors alongside a hierarchical skill-selection policy, this method applies hierarchical learning to a continuous skill space.

The skill selection policy is also trained using the actor gradients propagated through the DDPG actor loss, with the following form, where $\pi^\gamma(i \mid s)$ is the skill selection policy, and $i \in [n]$ is the index of the selected skill in the vocab of size $n$:

$$\begin{aligned}\mathcal{L}_{\text{actor}}(z,\gamma) \quad = \quad & \mathbb{E}_s\left[-\sum_{i=1}^{n} \pi^\gamma(i \mid s) Q^\phi(s, \pi^\theta(s, z_i))\right] \\ & +\lambda \mathbb{E}_s\left[\mathcal{H}(\pi^\gamma(\cdot \mid s))\right]\end{aligned}$$

The first term computes the expected critic value over potential indices selected by the skill selection policy. The second term is an entropy regularization term, which ensures that the algorithm doesn't immediately collapse to only selecting and optimizing a single policy. After sweeping possible values, all experiments used an entropy regularization constant of 0.01.

### 3.3. Hybrid Methods

As an auxiliary experiment, this project also tests whether Skill Tuning and/or Skill Vocab Tuning can effectively augment the default Policy Tuning method used in CIC. The only change to the previous two methods is to enable the actor parameters to be optimized alongside the skill vectors and skill selection policy parameters.

# 4. Empirical Results

## 4.1. Environment and Task

The environment and task used for these tests is the Mujoco Walker_Walk task (Figure 2), in which a two-dimensional two-legged walker is rewarded for achieving a specified horizontal velocity while standing with an upright and elevated torso segment. The reward function in this task has the following form:

$$r = \left(\frac{3}{4} r_{\text{torso\_elevated}} + \frac{1}{4} r_{\text{torso\_upright}}\right)\left(\frac{1}{6} + \frac{5}{6} r_{\text{forward\_vel}}\right)$$

Each of the reward components $r_{\text{torso\_elevated}}$, $r_{\text{torso\_upright}}$ and $r_{\text{forward\_vel}}$ scale from 0.5 to 1.0 in a margin around a desired value, and are 0 outside that margin.

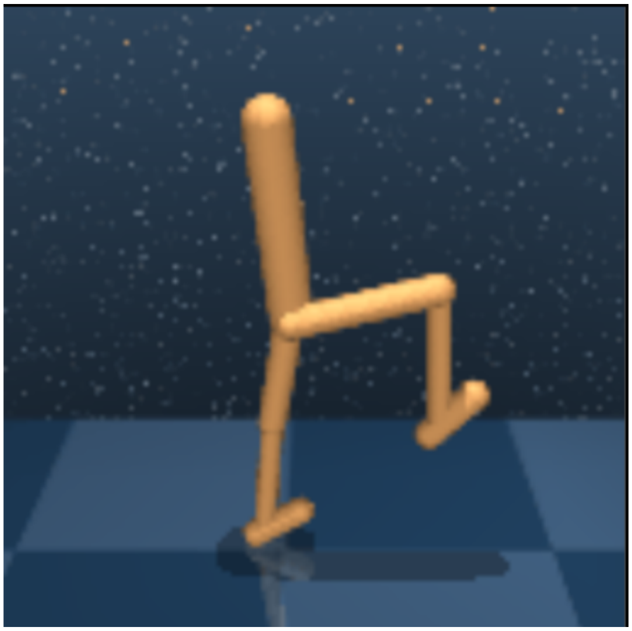

*Figure 2.* Rendered frame from the Mujoco Walker_Walk task. The agent is rewarded for velocity forward with an upright and elevated torso.

## 4.2. Results: Skill Tuning and Skill Vocab Tuning

Figure 3 shows the task-adaptation learning curves for Skill Tuning and Skill Vocab Tuning compared to default CIC Policy Tuning. Skill Tuning shows converges at a low return, and cannot solve the task. Skill Vocab Tuning succeeds at the task (albeit slowly), with a skill vocab of size 4 and 8.

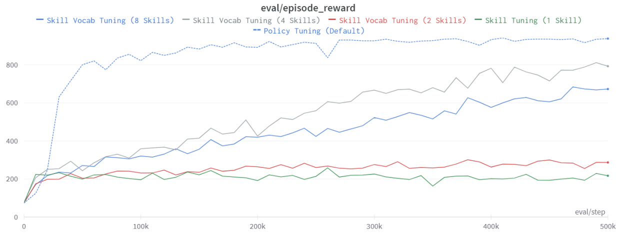

*Figure 3.* Task adaptation training curves for Skill Tuning and Skill Vocab Tuning alongside default CIC Policy Tuning. Skill Tuning very quickly converges to a low return policy and cannot get beyond that. Skill Vocab Tuning shows no benefit for a vocab of size 2, but shows consistent training for vocab sizes 4 and 8.

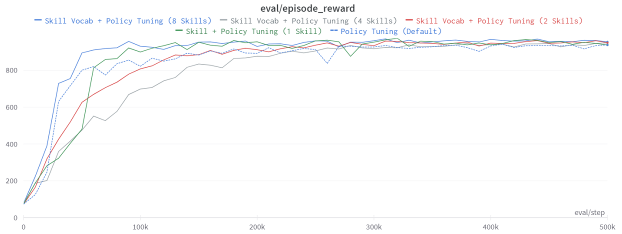

*Figure 4.* Skill and Skill Vocab Tuning + Policy Tuning, as compared to the default CIC Policy Tuning. Overall, the methods are generally comparable to Policy Tuning, without significant improvement.

To ensure that Skill Vocab Tuning remains meaningfully different from single-skill Skill Tuning, I recorded metrics which confirmed that while the skill selection policy does tend towards low entropy at each step (with max probabilities around 0.99), it never collapses to selecting only a single skill; the skill selection policy always puts high probability on at least two different skills within each batch, measured by checking the difference between the range of probabilities for each skill along the batch.

## 4.3. Results: Hybrid Methods

Combining Skill Tuning and Skill Vocab Tuning with Policy Tuning yields similar performance to Policy Tuning alone, and can inconsistently lead to small improvements over Policy Tuning. In Figure 4, both the single-skill method and the 8-skill method yield slight improvements over the default method, although not by a significant amount.

# 5. Discussion

In this project, we investigate alternate methods for adapting CIC skill-discovery pretrained policies to downstream tasks, tuning skill vector inputs rather than updating the skill-conditioned policy itself. This could hold potential for regularizing large pretrained network policies, and in the meantime provides insight into the richness of the skill space learned by Contrastive Intrinsic Control (Laskin et al., 2022). Since Skill Tuning with a single skill was unable to learn the Walker_Walk task, it's likely that the skill space learned by unsupervised CIC pretraining did not encompass a consistent walking behavior. However, tuning a finite vocabulary of skills alongside a hierarchical skill selection policy does allow for viable task adaptation without any updates on the pretrained skill-conditioned actor network. Further work could attempt to isolate the lower bound of required skill vocab size in order to successfully complete tasks, yielding a richer estimate of the behaviors covered by CIC pretraining.

Augmenting default CIC Policy Tuning with these Skill Vocab Tuning methods yields minor and inconsistent benefits, indicating that the hierarchical policy trained through backpropagated action loss is often outweighed by the simpler updates in the actor module directly. However, for both single-skill and 8-skill hybrid methods, the algorithm reaches convergence before the default Policy Tuning method, indicating that Skill Tuning may be a useful avenue for augmenting the adaptation abilities of continuous-skill pretrained policies.

## References

Achiam, J., Edwards, H., Amodei, D., and Abbeel, P. Variational option discovery algorithms, 2018.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function, 2018.

Gregor, K., Rezende, D. J., and Wierstra, D. Variational intrinsic control, 2016.

Laskin, M., Liu, H., Peng, X. B., Yarats, D., Rajeswaran, A., and Abbeel, P. Cic: Contrastive intrinsic control for unsupervised skill discovery, 2022.

Nam, T., Sun, S.-H., Pertsch, K., Hwang, S. J., and Lim, J. J. Skill-based meta-reinforcement learning, 2022.

Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. Dynamics-aware unsupervised discovery of skills, 2020.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Learning to prompt for vision-language models, 2021.