Extending Single-Task Policy Distillation in Reinforcement Learning

Eli Bronstein, Zhe Zheng
{ebronstein, zhezheng@berkeley.edu}

Abstract

Advances in deep learning have allowed models with very large numbers of parameters to become feasible and very successful. However, as a result, these huge models are very time-consuming during both training and prediction. *Policy distillation* seeks to address this concern by distilling the policy from a (usually larger) teacher network to a (usually smaller) student network. Previous work on policy distillation has shown that the student network can be compressed to be as small as 7% of the teacher network size, while achieving comparable if not better performance. Thus, we focus on improving the student network's training sample complexity, or how quickly it is able to achieve maximal reward. Specifically, we analyze the following questions: (1) how does the student's exploration strategy affects its learning behavior, and (2) how can a student network efficiently learn from multiple teachers of varying and (potentially unknown) expertise. To assess potential solutions to these questions, we conduct experiments in the Pong environment.

To address the first question, we experiment with several student exploration strategies such as greedy, epsilon-greedy, boltzmann, and bayesian exploration with dropout. Our results showed that. First, for a simple task such as Pong, using a small student model size and prioritizing exploitation over exploration (deploying exploration strategies such as greedy exploration and Bayesian exploration with Dropout) is enough to distill a student model significantly faster while having no degradation in performance. However distilling a policy for a more complex task or in the multi-task setting, special care might be necessary to leverage learning from the teacher and learning from exploration. Second, a student model, that is complex enough to express a policy distilled from a deep network may fail to explore and learn the policy by itself.

To explore the second question, we pose the problem of multi-teacher single-task policy distillation as a multi-armed bandit problem, where the teachers are the arms and the rewards are the rewards the students achieved after learning from the teachers. Given this framing, we analyze two potential settings. In the non-contextual bandit setting, show that bandit algorithms such as random, epsilon-greedy, and UCB1 perform well when learning from multiple expert teachers, and UCB1 efficiently learns even when there are multiple teachers of varying quality. In the contextual bandit setting, we suggest how contextual bandit algorithms can be used to learn a holistic policy over the entire state space from teachers that are experts in subparts of the state space.

1. Introduction

In deep reinforcement learning, the deep Q-network (DQN) algorithm receives state observations and rewards. The agent chooses the action to maximize the predicted cumulative rewards at each time step. DQNs have proven to be able to deliver superhuman performance on many challenging tasks including the majority of Atari games (Mnih et al., 2015). In the case of the game of Pong, a convolutional neural network (CNN) is used to process raw pixel observations and a simple feed-forward neural network is used to learn the policy from the learned state representation. However, DQNs require long training time and massive computing power.

Distillation is an efficient technique for supervised model compression (Bucila et al., 2006). Although some commonly used model compression techniques in supervised learning such as pre-trained models weight transfer have limited success in the context of reinforcement learning (Schmitt et al., 2018), *policy distillation* can transfer one or more action policies from Q-networks (teacher networks) to an untrained network (student network) while compressing the teacher network model size by up to 15 times with no degradation in performance (Rusu et al., 2015). In addition, policy distillation can efficiently combine multiple teacher policies into a single multi-task student policy that can outperform the its teacher policies, and it can be applied as a real-time learning process as well.

In this project, we focus on how to make improvements to single-task policy distillation. In general, policy distillation has two advantages: compressing the network size with little to no degradation in performance, and being able to train a student network significantly faster and to a higher level of performance. Previous work has shown that the student network size can be as small as 7% of the expert teacher network size while achieving better performance, we focus on reducing the student network's sample complexity. More specifically, we address two primary questions: (1) how does the student network's exploration strategy affect how quickly and well learns?, and (2) how can the student network efficiently learn from multiple teachers of varying expertise?

2. Previous Work

This work is closely related to model compression using distillation, or transferring knowledge from a deep neural network to a smaller network. The idea to train new networks by matching output distributions can be found in both supervised learning and reinforcement learning. Typically in the context of reinforcement learning, one would first fine-tune and train a teacher network and then use it to train a student model using supervised learning, either through imitation learning or policy distillation.

As suggested by Schmitt et al. (2018), *Kickstarting* leverages ideas from *policy distillation* (Rusu et al., 2015) and *population based training* (Jaderberg et al., 2017) and employs an auxiliary loss function which encourages the student policy to be close to the teacher policy on the trajectories sampled by the student. Importantly, there are no constraints on the architecture of the teacher or student agents and no constraints on change of weight of loss function in overall learning objective, so the student can gradually focus more on maximizing rewards instead of imitating or

distilling from the teacher. *Kickstarting* has better performances in more computationally expensive, more complex multi-task settings. However, the number of training time steps required for kickstarting in the original paper is more than 10 billion, so we focus on single-task policy distillation only.

3. Approach

Before describing our extensions to policy distillation, we begin by briefly summarizing deep Q-learning using deep Q networks (DQNs), as well as policy distillation.

3.1 Deep Q-Learning

Deep Q-learning is a state-of-the-art deep reinforcement learning approach that uses deep neural networks to estimate the average future discounted reward associated with taking a particular action a from a state s, also known as a Q-value. More specifically, suppose there is an environment with states $s \in S$, actions $a \in A$, and rewards $r \in R$. Then the goal of the DQN is to find the Q-function that maximizes the expected discounted reward for a state-action pair:

$$Q^*(s,\ a) = max_\pi\ E[\sum_{t'=t}^{T} \gamma^{t'-t} r_t | s_t = s,\ a_t = a,\ \pi],$$

where s_t is the sequence of states and actions up through time t. DQNs function by minimizing the Bellmann error using samples that are drawn from a replay buffer in order to decorrelate the samples. In order to stabilize the performance of DQNs, a separate Q-network and target Q-network are used, where the latter contains older versions of the network parameters. In addition, we utilized the double DQN extension to DQNs, which further decorrelates the Q-network and target Q-networks by using the current Q-network to predict the best action and the target Q-network to predict the value of this action.

3.2 Policy Distillation

Advances in deep learning have allowed models with very large numbers of parameters to become feasible and very successful. However, as a result, these huge models are very time-consuming during both training and prediction. Policy distillation (Rusu et al., 2015) seeks to address this concern by *distilling* the policy from a (usually larger) teacher network to a (usually smaller) student network. If the student network is able to accurately represent the teacher's policy with fewer parameters, it will be able to achieve similar performance to the teacher in a more efficient manner. This would also mean that an advanced policy does not always require a huge number of parameters, though learning to explore properly to learn such a policy might indeed require a more complex model.

We focus on single-task policy distillation, in which a large teacher trained on a single task is used to train a smaller student on the same task. Although both the teacher and student use the DQN algorithm, policy distillation can be used for other reinforcement learning algorithms. We assume there is a dataset $D^T = \{(s_i,\ q_i)\}^N_{i=0}$ from the teacher, where q_i are the Q-values of the

teacher associated with each possible action from state s_i. To train the student, we minimize a loss function between the student's Q-values and the teacher's Q-values. This loss function can either be minimized "offline", where the teacher's Q-values are recording during the teacher's training process and the student then learns from this dataset in a traditional offline supervised learning setting, or "online", where the student acts in the environment, queries the teacher for its Q-values, and then optimizes the student loss function online.

One approach for the student loss is to focus on encouraging the student to choose the same best action as the teacher $a_{i,\,best} = argmax(q_i)$, which amounts to using the negative log-likelihood (NLL) student loss:

$$L_{NLL}(D^T,\ \theta_s) = -\sum_{i=1}^{|D^T|} log\ P(a_i = a_{i,\,best}) \mid x_i,\ \theta_s\ .$$

However, a disadvantage of this loss function is that it only emphasizes the best action, but does not retain information about the relative values of the other actions. To address this concern, we also consider the mean-squared error (MSE) between the student's and teacher's Q-values:

$$L_{MSE}(D^T,\ \theta_s) = \sum_{i=1}^{|D^T|} \|q^T{}_i - q^S{}_i\|^2{}_2\ .$$

Another loss we consider, in line with viewing the Q-values as a probability distribution over the actions to choose, is the Kullback-Leibler (KL) divergence between the student's and teacher's Q-values:

$$L_{KL}(D^T,\ \theta_s) = \sum_{i=1}^{|D^T|} softmax(\tfrac{q^T{}_i}{\tau})\ ln\tfrac{softmax(\frac{q^T{}_i}{\tau})}{softmax(q^S{}_i)}\ .$$

We used the above three loss functions as described by Rusu et al., each of which allowed the student model to learn effectively from the teacher model. As described in our results, we confirmed that the KL divergence loss resulted in the best student performance, followed by the MSE loss and then the NLL loss. In addition, we experimented with additional loss functions, such as using a linear combination of the NLL and MSE losses, as well as computing the MSE loss with respect to the student's and teacher's action probabilities (softmax of their Q-values) rather than their raw Q-values.

3.3 Processing Teacher Q-values

Rather than optimizing a loss function directly as a function of the teacher's Q-values, a preprocessing function can be applied to improve how well the policy is distilled to the student. As described in Rusu et al., the teacher's Q-values can be passed through a softmax function a lower temperature, which results in a sharpening of the Q-value distribution. The purpose of doing so is to exaggerate the difference between the Q-values in order to facilitate the distillation of the knowledge from the teacher network. We assess the effect of not processing the teacher's Q-values, sharpening them, and softening them.

3.4 Student Exploration Strategy

In order for an agent, regardless whether it is a teacher or a student, to learn to take actions that maximize the cumulative reward, it need to be exposed to an appropriate diversity of states in the

environment, which is why exploration is needed. Exploration and exploitation, which in our case includes imitating and learning from the teacher as well, need to be meaningfully balanced. In our project, we tested several exploration strategies. We denote best action as: $a_{best} = argmax(q)$, and we used 4 different action-selection methods.

1. Greedy Exploration:

$$a_t = a_{best}$$

2. ε - Greedy Exploration:

$$a_t = \begin{cases} a_{best} \text{ with probability } 1 - \varepsilon \\ a_{random} \text{with probability } \varepsilon \end{cases}$$

3. Softmax Exploration with Boltzmann Distribution:

$$\pi(a|s) = \frac{e^{\frac{q(s,a)}{\tau}}}{\sum_{a' \in A} e^{\frac{q(s,a')}{\tau}}}$$

4. Bayesian Exploration with Dropout: we utilized dropout to simulate a probabilistic network or Bayesian neural network (BNN) by randomly setting output Q-values to 0 with probability p. When taking a single sample from a network with Dropout, we are doing something that approximates sampling from a BNN.

$$a_t = argmax(\, Dropout(q)\,)$$

3.5 Multiple Teachers

We extended single-task policy distillation to the multi-teacher case, in which we are given multiple teachers of potentially varying expertise trained on a single-task. The goal is to most effectively distill the teachers' policies to the student in terms of the reward achieved by the student and the number of samples required. In this situation, the quality and generalizability of the teachers may be unknown.

A naive solution to this problem involves evaluating the performance of each teacher in the environment, choosing the one with the largest empirical reward, and reducing the problem to the single-teacher case. However, this approach upper bounds the student's performance with the best teacher's performance without allowing the student to surpass its teachers.

We frame this situation as a multi-armed bandit problem, in which the arms are the individual teachers the student can learn from, and the reward is the reward that the student receives from the environment after learning from the teacher. In this case, the reward distribution is non-stationary because as the student learns from the teachers, its performance improves and the

reward it receives increases. Various general multi-armed bandit algorithms can be used to allow the student to intelligently choose which teacher to learn from at each training iteration.

The first class of bandit algorithms we explore is non-contextual, meaning no information is available when choosing which teacher to learn from other than the history of which teachers were chosen in the past and the rewards that were received. Such algorithms include 1) randomly selecting the teacher; 2) epsilon-greedy: selecting the "best" teacher with probability $1 - \varepsilon$ and a random teacher with probability ε according to some schedule for ε; 3) adaptive epsilon-greedy strategy based on value differences (VDBE) (Tokic); 4) Upper Confidence Bounds (UCB) algorithm and its variations, such as UCB1 and Bayesian UCB; 5) and Thompson Sampling, though many other algorithms exist. These algorithms can be employed to the simple case of multi-teacher policy distillation, particularly when we only wish to use the reward achieved by the student to decide which teachers to learn from.

The second class of bandit algorithms we consider is contextual, in which we are given a context vector that can be used to decide which teacher to learn from. Specifically, for contextual bandit settings we assume there is a dataset of samples (x, a, r, p) where x is the context, a is the chosen action (i.e. which teacher was chosen to learn from), r is the reward, and p is the probability that a was chosen given s for that sample. Contextual bandit algorithms would be particularly useful for multi-teacher policy distillation in situations where the teachers are experts in different parts of the state space. Using the state of the environment as the context, the student can learn a relationship between the context and which teacher is likely to allow the student to subsequently achieve larger rewards. In the ideal case, the student would perfectly learn which part of the state space each teacher is best at. Several contextual bandit algorithms that can be useful in this setting are 1) training a regressor to predict the reward given the context x and the chosen teacher a; 2) using importance-weighted classification to predict the action a using the context x with importance $\frac{r}{p}$; among others.

4. Results and Discussion

4.1 Experimental Setup

We chose the Atari game Pong as our task. State-of-the-art DQN models can achieve an average award of approximately 21, which is the maximum achievable reward, after 4M training steps. Our teacher achieved the same performance using similar configurations as suggested by Mnih et al. in 2013. Our teacher model has a total of 6,716,748 parameters, while for the student model, we reduced the number of CNN kernels and the size of the fully connected layers, resulting in 432,492 parameters, merely 6.4% of the teacher network size.

4.2 Student Loss Functions

We experimented with using several different student loss functions, including the KL divergence, MSE, and NLL losses presented in Rusu et al. As described in Rusu et al., we confirmed that using the KL divergence for the student loss results in the best student

performance in terms of how quickly the student reaches maximum average reward, followed by the MSE loss and then the NLL. Fig. 4.2-1 shows the students achieving the maximum reward after approximately 1.1 million timesteps, which is significantly faster than the approximately 2.5 million timesteps required by the teacher to the achieve maximum reward. In addition to those three losses, we tested combinations of those losses, such as a linear combination of the MSE and NLL losses. However, these additional losses were not significantly better than the KL divergence loss.

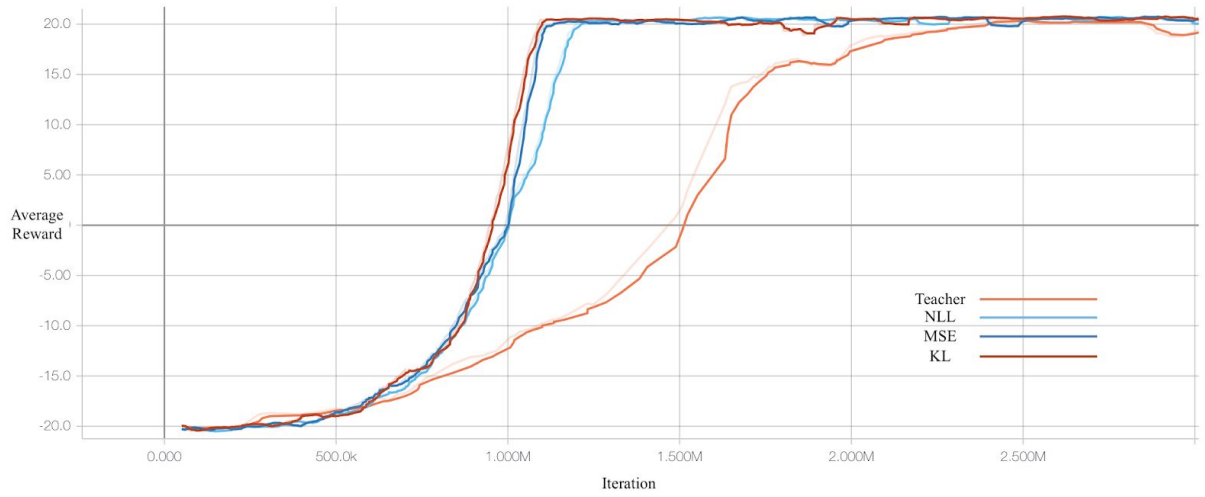

Figure 4.2-1 Learning curves of the teacher network (orange) and three of its student networks on the PongNoFrameskip-v4 environment. The students were trained with the KL divergence (red), MSE (dark blue), and NLL (light blue) losses.

4.3 Process Teacher Q-values

Rather than computing the student loss directly with respect to the raw teacher's Q-values, we experimented with processing the teacher's Q-values by applying a softmax function with a different temperature to sharpen or soften the Q-value distribution. We empirically found that sharpening the Q-values with a temperature $\tau = 0.01$ led to the best student performance. This is likely because sharpening the Q-values allows the student network to better learn the relative values of different actions better, thus distilling knowledge from the teacher network more effectively.

4.4 Student Exploration Strategy Results

We experimented with using several different student exploration strategies, including Greedy, Epsilon-Greedy, Softmax with Boltzmann distribution, and Bayesian exploration with Dropout. We used MSE losses, softmax teacher Q-values processing with $\tau = 0.01$ and only one teacher

model (reached maximum average reward and trained for 20M timesteps) for all our student exploration strategy experiments.

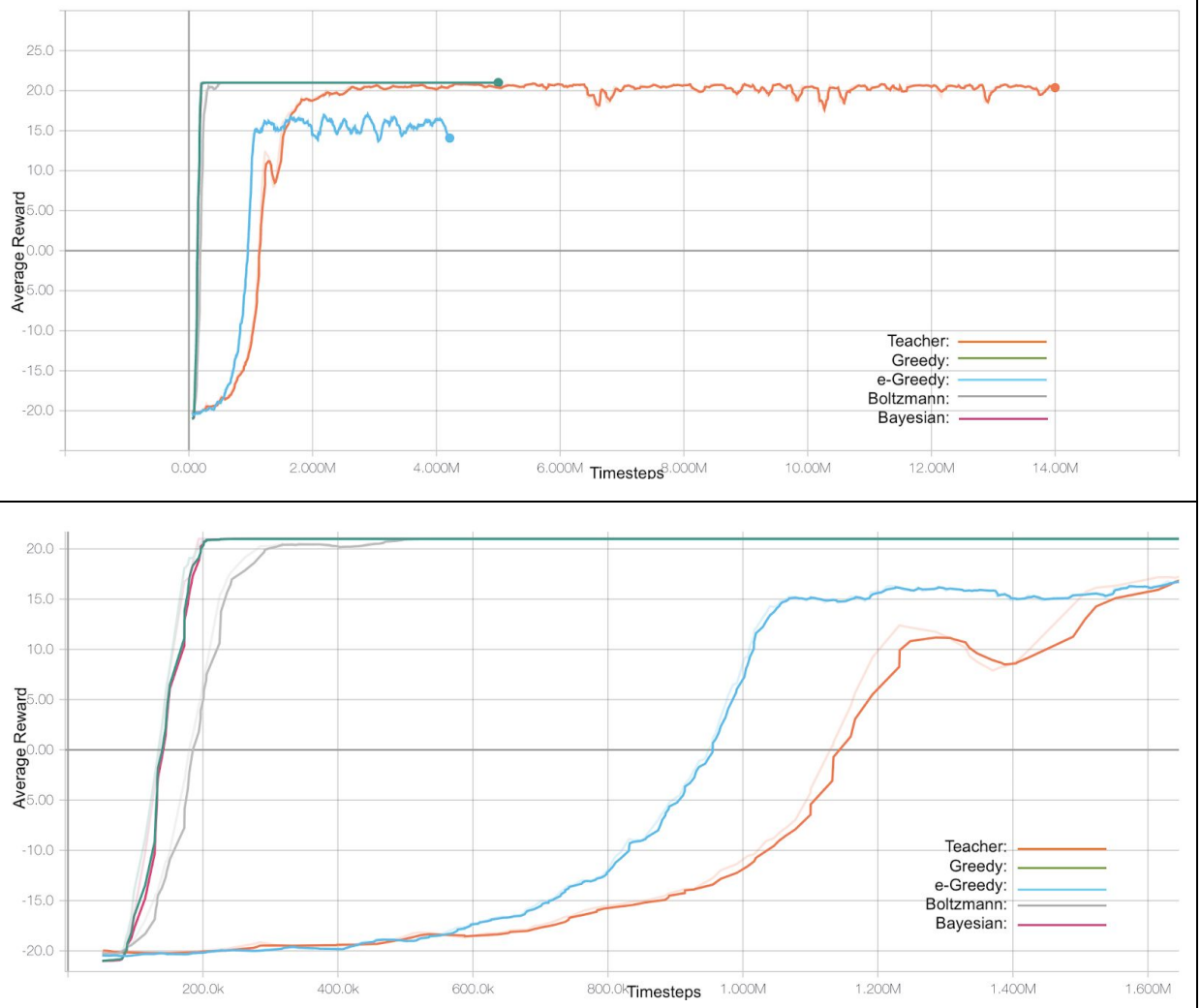

Figure 4.4-1 Learning curves of 4 students employing different exploration strategies trained on 1 teacher.

Student Exploration Strategy	Maximum Average Reward Reached	Timesteps Taken to Reach Maximum Average Reward
Greedy	**21**	**250K**
Epsilon-Greedy	16	5M
Boltzmann	**21**	400K
Bayesian	**21**	**250K**

Teacher with ε - Greedy	**21**	4M

Surprisingly, only Epsilon-Greedy exploration strategy failed to reach average reward of 21 while Greedy and Bayesian exploration reaching the maximum in 250K timesteps (including 50K exploration steps). We believe this is because our student model's network size is not large enough to properly learn and explore such a policy. Also, since Pong is a rather simple task, the pre-trained teacher model could be regarded as a perfect solver for the task and its Q-value function is a close enough approximation of the true action-value function. So, in order for the student model to learn faster and better, learning from the teacher is a better option than learning from exploration.

In general, when distilling a policy in a simple single-task context, we would recommend using a small student model size and prioritizing exploitation over exploration. We also hypothesize that when distilling a policy in a complex single-task context, choosing a student model size that is complex and large enough to learn the policy on its own and an exploration strategy with enough exploration could lead to the student model surpassing the teacher model in performance.

4.5 Multiple Teachers Results

We explore two different bandit settings: non-contextual and contextual bandit. For the non-contextual setting, at each training iteration the student is tasked with choosing which teacher to learn from. The student only possesses information about which teachers it chose in the past and the rewards it received after learning from each teacher. For the experiments, we used the 4 different teachers shown in Fig. 4.2-1: the large teacher and its 3 students, which were trained with the KL divergence, MSE, and NLL losses. As can be seen in Fig. 4.5-1, we trained 3 different students, each using a different multi-armed bandit algorithm: random, epsilon-greedy, and UCB1. The performance of these students is practically identical because they learned from 4 expert teachers that likely had very similar policies. Thus, choosing a random teacher to learn from at each iteration is as good of a learning strategy as any other.

To better analyze the effect of the bandit algorithm, we conducted an experiment with 12 different teachers. However, due to how time-consuming training 12 different teachers would be, we utilized the same 4 teachers as in the previous experiment to create 8 additional artificial teachers. By adding random Gaussian noise to the teacher's Q-values, we can create a teacher that is arbitrarily worse. Using this method, we gathered 4 expert teachers (the original 4 teachers with no noise), 4 "good" teachers (the teachers with $N(0, 1)$ noise), and 4 "mediocre" teachers (the teachers with $N(0, 2)$ noise). As shown in Fig. 4.5-2, this student is able to achieve maximum reward as quickly as the students trained only on the expert teachers, suggesting that the UCB1 algorithm is able to effectively choose which teachers to learn from in the presence of multiple teachers of varying quality.

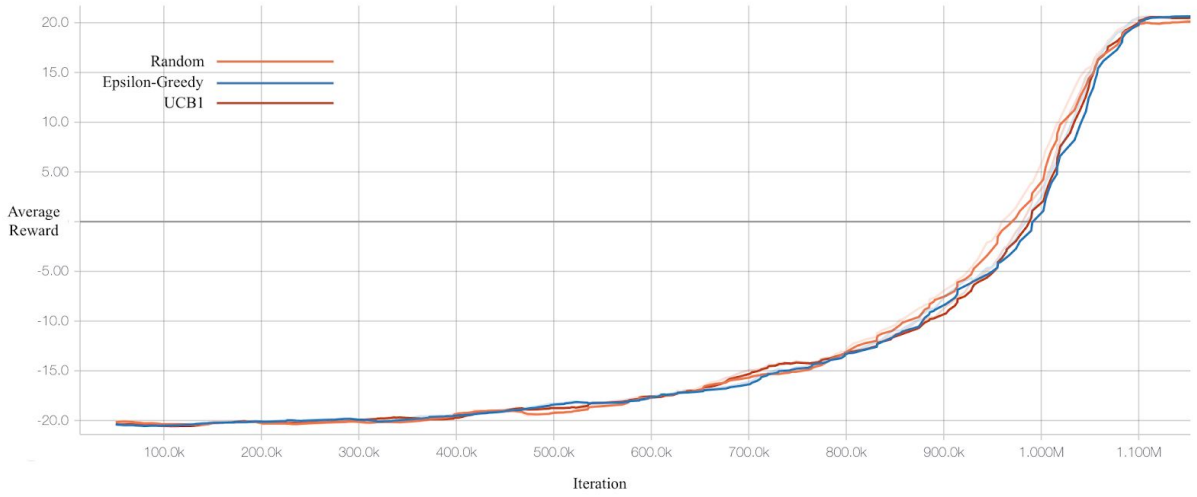

Figure 4.5-1 Learning curves of students employing different multi-armed bandit algorithms trained on 4 teachers: a large teacher and its 3 students, which were trained with the KL divergence, MSE, and NLL losses. The learning performances of these different bandit algorithms are equivalent because all 4 teachers are experts and likely use very similar policies.

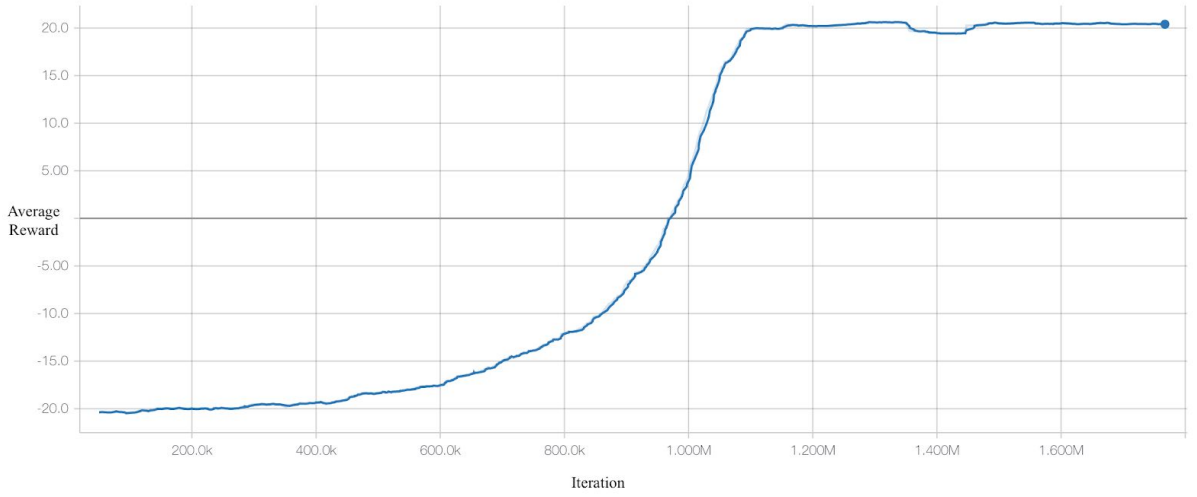

Figure 4.5-2 Learning curve of student using the UCB1 algorithm, trained on a total of 12 teachers: 4 expert teachers (the original 4 teachers with no noise), 4 "good" teachers (the teachers with $N(0,\ 1)$ noise), and 4 "mediocre" teachers (the teachers with $N(0,\ 2)$ noise).

However, the multiple teachers could be experts in a specific part of the state space. Agents that are trained in different instances of the same type of general environment might have this characteristic. For example, autonomous vehicle agents trained in urban and suburban environments are both helpful for agent learning to drive autonomously in any area, but the two

environments could result in very different policies. In this situation, having context about the state, or some other useful information, might help the student choose which teacher to learn from in a given situation. To simulate this scenario, we trained two different Pong agents. The first "top" agent was only trained on states in which the ball was in the top half of the screen, while the second "bottom" agent was trained on states in which the ball was in the bottom half of the screen. The "top" agent performed decently well in its state subspace of expertise, consistently achieving the maximum reward in later episodes and an average reward of up to 15.02. The "bottom" agent did not perform as well in its state subspace of expertise, only achieving a maximum reward of 2. Naturally, both of these agents performed poorly when operating in the entirety of the state space.

Unfortunately, due to time constraints we were unable to complete the contextual bandit algorithm experiments. However, there are many interesting avenues for future work, which we explore in the following section.

5. Discussion and Future Work

Our project has explored some options to improve policy distillation in the single-task context. We aim to minimize the student model's size while allowing the student model to achieve a performance that is at least as good as that of the teacher models. We confirm the results of previous work on policy distillation, showing that the KL divergence student loss and teacher Q-value sharpening leads to the best student performance. In addition, we analyze the effect of different student exploration strategies on the student's learning efficiency, showing that greedy and Bayesian exploration strategies are the best. Moreover, we address the situation of single-task multi-teacher policy distillation by framing it as a multi-armed bandit problem, and show some promising results about the use of bandit algorithms in this setting. We believe, those experiments could open pathways for speeding up the training process of deep reinforcement learning models for industrial deployment where computation power is limited and real-time action is needed.

To extend this work, we aim to further explore how multi-armed bandit algorithms can be used for multi-teacher single-task policy distillation. For the non-contextual bandit case, more extensive experiments can be conducted with teachers of a greater variety of expertise, and additional bandit algorithms can be employed such as adaptive epsilon-greedy strategy based on value differences (VDBE), Bayesian UCB, and Thompson Sampling. For the contextual bandit case, we wish to test contextual bandit algorithms on various scenarios in which each teacher is an expert on its own subpart of the state space to assess how well multiple disparate teacher policies can be learned and fused into one holistic and smaller student policy.

Works Cited

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *KDD*, pages 535–541. ACM, 2006.

Jaderberg, Max, Dalibard, Valentin, Osindero, Simon, Czarnecki, Wojciech M., Donahue, Jeff, Razavi, Ali,Vinyals, Oriol, Green, Tim, Dunning, Iain, Simonyan, Karen, Fernando, Chrisantha, and Kavukcuoglu, Koray. Population based training of neural networks. CoRR, abs/1711.09846, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. *Playing atari with deep reinforcement learning. Deep Learning Workshop*, NIPS, 2013.

Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, Raia Hadsell. *Policy distillation*. ICLR, 2015

Simon Schmitt, Jonathan J. Hudson, Augustin Zidek, Simon Osindero, Carl Doersch, Wojciech M. Czarnecki, Joel Z. Leibo, Heinrich Kuttler, Andrew Zisserman, Karen Simonyan, S. M. Ali Eslami. *Kickstarting Deep Reinforcement Learning*, March 2018

Tokic, Michel. "Adaptive ε-greedy exploration in reinforcement learning based on value differences." Annual Conference on Artificial Intelligence. Springer, Berlin, Heidelberg, 2010.