Dual-Observation Self-paced Policy Co-Training

Kordel K. France KFRANCE8@JH.EDU
Whiting School of Engineering
Johns Hopkins University
3400 N. Charles St.
Baltimore, MD 21218, USA

Editor: None

Abstract

A paramount component behind many successful robotic navigation systems is reinforcement learning. Imperfect knowledge of the environment limits the circumstances behind which a reinforcement learning agent can approximate a policy it is able to use to navigate the optimal path. The difficulty behind this problem is further exacerbated with limited experience to construct a policy from. To help reconcile scenarios of this type, we propose Dual-Observation Self-paced Policy Co-Training (DOSPCoT) in which we demonstrate the ability of two agents to collaboratively navigate an unknown environment with very minimal training data. We hypothesize that self-paced co-training can allow multiple agents with imperfect knowledge to consolidate knowledge from each of their imperfect policies in order to approximate an optimal policy of a degree more robust than that which would be achieved if each agent acted alone. Additionally, we show how the performance of co-training reinforcement learning agents can be maximized through the use of Expected SARSA as the policy learner by strategically exploiting the self-paced tuning parameters. We demonstrate the superior ability of the DOSPCoT framework to navigate an unknown environment by analyzing its performance over a 2-dimensional navigation task. Finally, it is shown how DOSPCoT, while proposed for traditional reinforcement learning techniques, can be naturally extended to inverse reinforcement learning and imitation learning for a multi-model co-trainer.

1. Introduction

The generally accepted notion of "two heads are better than one" suggests that multiple minds working on the same difficult problem can surmise a solution faster than if each mind was otherwise working alone. Collaborative problem solving implies having multiple agents observe the problem in order to offer different angles of insight. Reinforcement learning is an optimization technique commonly used in control problems that leverages the use of rewards in order to approximate a solution to a problem. It has garnered much attention in its ability to out-play humans in board games and navigate vehicles. One of the difficulties of successfully training a reinforcement learning agent is obtaining full observability over all possible states that the agent can encounter. This lack of observability could be due to the cost of obtaining the knowledge itself, the ambiguity of differentiating states in the observation space, or a host of other reasons. In a pathfinding problem, an agent trained by reinforcement learning must decide how to navigate such conditions given its policy.

Consider a scenario of robotic navigation in which there are multiple robotic vehicles whose job is navigate from one end of a course to another. Each agent only has partial knowledge of the entire course map, and perhaps even different domains of knowledge. Through collaborative learning, each agent can infer the action that will achieve the highest approximate reward by observing states it does have knowledge of and evaluating the rewards received from applying its

policy together with another agent in a collaborative fashion. We hypothesize that self-paced co-training can allow multiple agents with imperfect knowledge to consolidate knowledge from each of their imperfect policies to approximate an optimal policy to a degree higher than that which would be achieved if each agent acted alone. In doing so, they construct *an* optimal policy for navigating from one end of the course to the other and effectively determine *an* optimal path, assuming one exists.

In this paper, we present Dual-Observation Self-paced Policy Co-Training (DOSPCoT) a co-training framework for learning more efficient policies to reason about minimally observable environments. We leverage this framework to help confirm our hypothesis and show that learning from two observations of state-action trajectories can improve policy construction over learning from a single observation alone. DOSPCoT extends the concept proposed by Ma et al. (2017, 2020) in which they utilize self-paced co-training to train classifiers on two or more different views and exchange pseudo labels of unlabeled instances. Effectively, they propose a method that allows for the classification of unlabeled data based on minimal training data. The paper focused exclusively on using this technique among classification problems, but there is a natural extrapolation to accommodate Markov decision processes (MDPs) and reinforcement learning. To illustrate, we analyze state-action pairs, where each state-action pair is sampled from a map that is whose rewards are mostly unobservable. In other words, the rewards for most of the state-action pairs are unknown or masked. Instead of learning labels of image or text data, one can learn rewards and apply "pseudo-rewards" to unobservable states instead of "pseudo-labels" to unlabeled instances as a means of learning a policy versus a classifier. In the context of multi-view co-training, a "view" in classification may be interpreted as being synonymous with an "observation" in a MDP-based pathfinding problem.

Much of the same notation from the original SPaCo paper by Ma et al. (2017) can be directly leveraged in order to accomplish this. However, there are some adjustments needed to both the process and methodology in order to map the problem from a classification one into one based on reinforcement learning, which shall be noted in the succeeding section. We maintain the idea of self-paced regularization and attributing weights to each sample, except in this case a "sample" is synonymous with a state-action "trajectory". The self-paced aspect of co-training shall also be maintained from the original SPaCo work; in addition, however, the greediness ϵ of the reinforcement learning algorithm shall be a function of the self-paced hyperparameter λ. Expected SARSA shall be the specific RL algorithm trained by co-training. Sutton and Barto (2018) demonstrate that, by allowing for the greediness to be tuned throughout policy learning, Expected SARSA converges to Q-learning by the end of the training process. We intend to leverage this principle by increasing greediness ϵ in direct proportion to λ.

In much modern co-training research regarding classification, there is concern around independence between views. In the context of this paper, this would translate to a concern between observations of state-action sequences satisfying the Markov property. A key assumption made with the algorithms proposed in this paper is that the Markov property is satisfied, and that one state within the navigational map is not changed by an action taken within another state. In a case of collaborative problem solving with two navigation agents occupying the same map, there are some conditions in which the Markov property is unsatisfied. Namely, if one state is occupied by one agent, the other agent cannot occupy that same state at the same time, limiting the latter agent's action space if such an action is available. While we do not directly address this problem in this paper, we acknowledge its importance and leave it as an item of interest for future work.

The rest of the paper will be organized as follows. A summary of related work along with its relevance to the methods proposed here is given in Section 2. Section 3 presents the DOSPCoT framework and illustrates its relevance in pathfinding. In Section 4, the optimization methods regarding the DOSPCoT framework are proposed. Section 5 describes the experiments performed

and quantifies their results. In Section 6, we discuss opportunities to extend DOSPCoT and our intentions for future work, after which we provide concluding remarks in Section 7.

2. Background and Related Work

With the our proposed DOSPCoT framework, our intention is to investigate a derivative of Ma's 2017 work with SPaCo but within the context of reinforcement learning. Our analysis of related work contains many references to algorithms that ultimately act as the building blocks of SPaMCo (Ma et al. 2020), but it is to our knowledge that no prior art exists surrounding the exact methodology proposed in this paper.

2.1. Reinforcement Learning

Among the reinforcement learning umbrella exist imitation learning and inverse reinforcement learning. In their paper, Abbeel and Ng (2004) define inverse reinforcement learning (IRL) as the problem of deriving a reward function from observed behavior of a supposed "expert". Conversely, Hester (2018) defines imitation learning (IL) as deriving a policy from demonstrations of expert behavior. While parallels of each can be drawn to the method proposed here, both IL and IRL suggest that the model to be imitated demonstrates perfect expert behavior, whereas the behavior of our co-trained policy is highly unobservable and therefore can not qualify as an expert to imitate. IRL requires traditional model-based reinforcement learning in order to approximate the reward function, and imitation learning can be reduced to the same traditional methods for deterministic agents as Ciosek (2022) demonstrates. Later we show how the DOSPCoT framework can be slightly retuned to accommodate both IL and IRL as subroutines.

2.2. Co-Training and Co-Regularization

The notion of co-training was proven tractable in 1998 under the initial assumption that instances from different views are independent contingent on the co-trainer's classifier making useful predictions on unlabeled data (Blum and Mitchell, 1998). Brefeld and Sheffer (2004) implemented a co-training algorithm incorporating a naive Bayes classifier with SVM improving upon prior work accomplished by Nigam and Ghani (2000).

Later attempts to provide a more resilient co-regularization function were published by Sindhwani and Li, in which they attempt to encode prediction dependencies among views into a single co-regularization term (Sindhwani et al. 2005; Li et al., 2012). The resulting objective function turned out to be difficult to optimize and orthogonal to the fundamental approach to co-training. Many attempts up to harmonize technical nuances behind co-training up to this point leveraged only two-shot cases and gave unclear performance measures. In 2019, Zhou published work on using pseudo labels and abductive learning to improve classification of unlabeled data, building on his work with Zhang in 2011 that prefaced this strategy with neighboring graphs. Ma et al. proposed SPaCo (self-paced co-training) in 2017 that established a more generalizable objective function and self-paced learning technique over a pseudo-supervised co-training algorithm. However, this model still only complied with two-view cases. Building on top of this work, Ma et al. began developing methods into SPaCo that allowed for resiliency against false negative samples, accommodated multi-view cases, and improved on co-regularization that manifested into what eventually became SPaMCo (Ma et al, 2020). SPaMCo introduced co-training such that it could be used beyond the two-view scenario. A natural extension of the work presented in this paper is to apply the SPaMCo technique to allow for more than two agents to learn from each other via multi-observation co-training of the reinforcement learning policy. Similar works to SPaMCo followed with Huang et al. (2021) showing multi-view co-training in clustering that built upon the works of Xu, Tao, and Xu (2015); and Xia et al. demonstrated similar effects over image segmentation in 2020, among others. In 2021, Wang and Peng et al. introduced the concept of self-paced *and* self-consistent co-training over image segmentation.

2.3. Co-Training and Reinforcement Learning Theory

From our research, works attempting to apply co-training with the multi-view or self-paced flavor to reinforcement learning have been scarce. Ziyu et al. (2016) shows how two "dueling" models can collaboratively achieve great performance, but they do not use co-training and suggest to split the state-value and action functions which is not the intent here. Amelia and Lin (2021) demonstrate how co-training can be used to train a reinforcement learning agent on when to select an action by learning a temporal policy. Wu, Li, and Wang (2018) propose training a Q-learning agent to learn a policy for data selection and then exploit that policy to train co-training classifiers automatically. In this context, RL was used as a means to train a supervisor over a classification problem, but a RL-based agent was not actually trained with co-training itself. The work of Song et al. (2020) demonstrate how co-training can be used to enable a RL agent to learn a policy in settings with multiple state-action representations. This last work seems to align closely with the research proposed here in regard to RL being combined with co-training, but they fail to incorporate the self-paced element as intended to here; additionally, they demonstrate learning a co-trained policy as a means for more efficient policy learning, not as a technique for learning policies whose domains have little known data. While Song and company did use a Q-learning algorithm, they do not demonstrate the ability of Expected SARSA to converge to Q-learning as a function of the self-paced parameter we intend to here.

3. Algorithm

In prose form, the DOSPCoT model may be interpreted as two policy learner's collaboratively minimizing regret. Muthukumar defines regret as the loss between the reward received under the current policy versus the reward received by the action taken under the optimal policy. The two policy learners (two Expected SARSA agents) are iteratively trained through the exchange and refinement of predicted *pseudo rewards* assigned to state-action pairs, or *trajectories*.

3.1. Self-Paced Learning

Let the ground truth reward for the i^{th} trajectory be denoted as y_i where $i \in [0,N]$ and N denotes the number of trajectories available in the observable domain. Similarly, let the i^{th} input state-action sequence—or *trajectory*—be denoted as $\phi_i \in \Phi$, the vector of model parameters be denoted as $\theta \in \Theta = \{\theta^{(1)}, \theta^{(2)}\}$, and g denote the decision function that estimates the pseudo reward. Let the loss function that calculates the cost between the ground truth reward and the estimated reward be denoted as

$$L(y_i, g(\phi_i, \theta, \epsilon))$$

where $g(\phi_i, \theta, \epsilon)$ denotes the estimated reward as a function of the input sequence, model parameters, and greediness ϵ (later we will show that this reward function takes the form of Expected SARSA). Let $f(v, \lambda, \epsilon)$ denote the self-paced regularization function where v relates to the latent weights of unrewarded instances, λ is the age parameter that moderates learning within the self-paced model, and ϵdenotes the current greediness of the policy. The above loss function may be used in expressing the self-paced learning model in the following manner:

$$\min_{\theta, v \in [0,1]^n} E(\theta, v; \lambda; \epsilon) = \sum_{i=1}^{n} (v_i L(y_i, g(\phi_i, \theta)) + f(v_i, \lambda, \epsilon)) \quad (1)$$

Note that the self-paced regularization term for the j^{th} view of the i^{th} trajectory sample can be written as $\lambda^{(j)} v_i^{(j)}$ as shown in Jiang et al. (2014a). The adjustment of the age parameter λ coupled with the joint learning of the model parameters and the latent weights allow for the model to moderate the number of training samples during each iteration of the algorithm as a function of sample complexity.

3.2. Soft Co-Regularizer

The DOSPCoT framework allows for a weight of each unlabeled training instance to coordinate learning order; weights are added to rewarded sequences after enough sequences have been assigned rewards. These weights help determine the pseudo rewards to apply to sequences during training. The policy realized via co-training learns by analyzing two observations of the same state-action sequence and exchanging the pseudo rewards associated with that sequence. A soft co-regularization harmonizes this to aid the self-paced learning capability of the model. Different observations maintain common knowledge of the confidence in the pseudo rewards assigned to samples. Due to this, the inner product of the weights for both observations enforces the weight of the reward penalizing the loss of one trajectory observation similar to another. The co-regularizer ensures that the learned reward estimations from multiple observations of unrewarded trajectories are consistent throughout training, while simultaneously working to moderate the complexity of observations trained. The co-regularization term R for the p^{th} and q^{th} observation where $p, q \in [1,2]$, $q = N_u - p$, $p < q$, and N_u denotes the total number of unrewarded state-action sequences is defined as:

Co-Regularization Term:
$$R(V) = \gamma \sum_{p=1}^{N_u} (v^{(p)} - v^{(q)})^T (v^{(p)} - v^{(q)}) \tag{2}$$

where $v^{(p)}$ contains the weights of unrewarded sequences in the p^{th} observation and γ denotes the age hyper parameter that controls the association degree between different observations.

For a single trajectory weight, the term may be more simply expressed as:

Co-Regularization Term of Single Instance:
$$r_s(v) = \gamma (v^{(1)})^T v^{(2)} \tag{3}$$

The self-paced learning element of the SPaCo allows the model to incrementally grow its confidence in reward predictions with each successive iteration, allowing it to learn with increasingly complex sequences. The selection for proper values of the age parameter λ shall be defined in the section 5 to demonstrate the ability of self-paced learning to handle sequences with noisy rewards.

3.3. Expected SARSA

Q-learning is an off-policy temporal difference algorithm that learns to directly approximate the optimal action-value function independent of the policy being followed. It learns a policy that selects the next state-action pairs that produces the maximum reward. It is one of the most commonly used temporal difference algorithms in reinforcement learning due to its computational efficiency and good approximation of the total cumulative reward. However, consider the following: during the initial iterations of co-training, the confidence in assigning rewards to state-action pairs is low; one can effectively consider rewards as being assigned to state-action pairs at random (alluding to the expected performance of regular SARSA). Due to

this, it is obvious that immediately seeking the state-action pair that returns the highest cumulative reward may produce high volatility in the loss for incorrect reward assignments. Thus, it is better to start with an algorithm that seeks the state-action trajectory that returns the *expected* reward over the *maximum* reward in order to control loss volatility. The regularization terms defined above do provide some protection against this, but they are not entirely comprehensive. Expected SARSA, initially proposed by Sutton and Barto (2018) provide an easy vehicle for this and it is for the reasons above that it is employed as the RL algorithm of choice for DOSPCoT

As Ma et al. demonstrated in their 2017 SPaCo and 2020 SPaMCo with image classification, we show that, as confidence in reward assignment grows through successive iterations, DOSPCoT becomes more accurate in correctly attributing rewards to state-action trajectories. Therefore, the Expected SARSA policy learner may be allowed to become more greedy. As ϵ approaches 1, it begins to select the action that results in the highest cumulative reward and evolves into exactly Q-learning (Sutton and Barto, 2018). Expected SARSA will allow the algorithm to take advantage of both worlds by moderating the learning of its policy as a function of correct reward assignment and λ. In this fashion, early iterations of DOSPCoT will begin with a policy that approximates SARSA moderating the high loss, and then evolve into approximate greedy Q-learning as the policy becomes more refined. This also affords the algorithm to be used as both an online and offline policy.

SARSA:

$$Q^*(s,a) = Q(s,a) + \alpha[\ R(s,a,s') + \gamma Q(s',a') - Q(s,a)\ ] \tag{4}$$

Q-Learning:

$$Q^*(s,a) = Q(s,a) + \alpha[\ R(s,a,s') + \gamma \max_{a'} Q(s',a') - Q(s,a)\ ] \tag{5}$$

Expected SARSA:

$$Q^*(s,a) = Q(s,a) + \alpha[\ R(s,a,s') + \gamma \frac{1}{n}\sum_{i=1}^{n} Q(s_i',a_i') - Q(s_i,a_i)\ ] \tag{6}$$

In analysis of the above, one can see that the only difference that Expected SARSA possesses over Q-learning is the averaging of the rewards in **Equation 6** in place of their *argmax* in **Equation 5**. Due to this, Expected SARSA is more computationally expensive because the expected reward is being calculated at each update state. However, because of the reasons delineated above, it proves to be more accurate and robust in co-training since it eliminates the early volatility in reward estimation due to random selection of the next action. Especially in the beginning of co-training when the pseudo-rewards of unrewarded trajectories is highly erroneous, Expected SARSA can provide smoother learning. Since the objective is to accurately infer rewards for unobservable states, the increase in computational complexity seems justified. Expected SARSA affords us the freedom to fluctuate between on-policy and off-policy methods while providing a natural means of controlling reward loss.

3.4. DOSPCoT Framework

The DOSPCoT algorithm is simply the self-paced learning model presented in **Equation 1** added to the co-regularization term from **Equation 3**. Two other regularizer terms are also added—the model parameter regularization term and the self-paced learning regularization term from **Equation 2**. In its entirety, DOSPCoT can be defined as

$$\min_{\Theta,V,Y\in[0,1]} E(\theta, v; \lambda; \epsilon) = \sum_{j=1}^{2}\sum_{i=1}^{N_r} L(y_i, g^{(j)}(\phi_i^{(j)}; \theta^{(j)}, \epsilon)) + \frac{1}{2}\sum_{j=1}^{2} \|\theta^{(j)}\|_2 + \sum_{j=1}^{2}\sum_{k=N_r+1}^{N_r+N_u} (v_k^{(j)} L(y_k, g^{(j)}(\phi_i^{(j)}; \theta^{(j)}, \epsilon)) - \lambda^{(j)} v_k^{(i)}) + \gamma (v^{(1)})^T v^{(2)} \quad (7)$$

where $g(\phi_i; \theta, \epsilon)$ denotes the reward received by the policy of Expected SARSA as a function of the trajectory ϕ_i, model parameters θ, and greediness ϵ.

4. Optimization Strategy

The routine for optimizing the loss of DOSPCoT is denoted in **Algorithm 1.** While the algorithm generates two observations—one trajectory observation per reinforcement learning agent—we describe the steps of the algorithm in terms of one observation for the ease of interpretation since both agents are optimized in an identical manner. Our optimization algorithm is largely based on that proposed by (Meng & Zhao, 2015; Ma et al., 2017) in which they demonstrate that self-paced learning can be effectively deduced to a robust loss minimization problem that is solvable by a majorization-minimization algorithm. We make the necessary minor modifications to adapt it to policy learning.

Algorithm 1 DOSPCoT Optimization Algorithm

Input: samples $\phi_1^{(1)}, ..., \phi_{N_r+N_u}^{(1)}, \phi_1^{(2)}, ..., \phi_{N_r+N_u}^{(2)}$, labels $y_1, ...y_{N_r}$, parameters $\lambda^{(1)}, \lambda^{(2)}$, and γ
Output: $v^{(1)}, v^{(2)}$
  Initialize $\phi^{(1)}, \phi^{(2)}, \lambda^{(1)}, \lambda^{(2)}$, and γ
  Update $v^{(1)}, v^{(2)}$
  training_iter = 1
  while not converge || training_iter < max_iter **do**
   for $j \leftarrow 1 : 2$ **do**
    Update $v_k^{(3-j)}$: Prepare confident trajectories for $(3-j^{th})$ observation for training
    Update $v_k^{(j)}$: Add unrewarded trajectories into the training pool
    Update $\theta^{(j)}$: Train an RL agent on training pool of j^{th} observation
    Update y_k : Find optimal pseudo reward for selected unrewarded trajectories
    Augment $\lambda^{(1)}, \lambda^{(2)}$
   end for
end while
Return $v^{(1)}, v^{(2)}$

The algorithm first initializes the input instances $\phi^{(1)}, \phi^{(2)} \in \Phi$ as unrewarded trajectories; it assumes both policy learners, Expected SARSA agents $\alpha^{(1)}$ and $\alpha^{(2)}$, are also appropriately initialized. We need a small amount of unrewarded data to begin training $\alpha^{(1)}$ and $\alpha^{(2)}$ so we initialize the age parameters $\lambda^{(1)}$ and $\lambda^{(2)}$ to very small values to allow only a few unrewarded

trajectories into the training pool. In their original 2020 SPaMCo paper, Ma et al. suggest to initialize the age parameters to 0.3, which we have also found to provide a very controlled and robust training pace so we replicate it here. We set both of the weight vectors $v^{(1)}$ and $v^{(2)}$ as zero vectors, and then proceed with training both agents over the selected instances to acquire the initial losses over both unrewarded and rewarded trajectories.

From here, we begin co-training by selecting trajectories of which we are confident are rewarded in accordance to the optimal policy. Rewarded instances with loss values less than that of $\lambda^{(2)}$ from the observation of $\alpha^{(2)}$ are viewed as confidently rewarded trajectories that should be selected for the training pool for $\alpha^{(1)}$. In a similar manner, we must then select which *unrewarded* trajectories will be added into the training pool for $\alpha^{(1)}$ by sampling from values whose loss is greater than that of $\lambda^{(2)}$. If the age parameters are tuned appropriately and the self-paced nature of the training is correctly controlled, trajectories selected for training agent $\alpha^{(1)}$ will maintain a higher probability of being selected for the training pool of agent $\alpha^{(2)}$ and vice-versa.

The weight vectors $\theta^{(1)}$ and $\theta^{(2)}$ must be updated next to reflect new knowledge from the previous iteration. By updating these vectors, we are effectively training a classifier from truly rewarded and pseudo-rewarded samples. Note that this classifier is separate from the Expected SARSA policy learners $\alpha^{(1)}$ and $\alpha^{(2)}$, but it is easy to make the incorrect assessment that DOSPCoT is performing supervised learning over a trajectory classifier—this would be completely untrue. The classifier here is a separate learner that is being trained to perform simple binary classification over rewarded and pseudo-rewarded trajectories. In fact, in their original 2017 SPaCo paper, Ma et al. show that the SPaCo model degenerates to simple SVM optimization by setting hinge loss as the loss function. We do not provide the formal proof of this here, but by direct extension of their work, we can derive the following SVM optimization model from **Equation 7**:

$$\min_{\theta^{(i)} \in [0,1]} \frac{1}{2}\|\theta^{(i)}\|_2 + \sum_{j=1}^{N_r} L_j^{(i)} + \sum_{k=N_r+1}^{N_r+N_u} v_k^{(i)} L_k^{(i)} \tag{8}$$

where $L_t^{(i)} = L(y_t, g^{(i)}(\phi_t^{(i)}; \theta^{(i)}, \epsilon)), t = 1, \ldots, N_u + N_r$.

Next, we must update the pseudo-rewards of each trajectory. We can do so by computing each reward through the solving of the following minimization problem:

$$y_k = \underset{y_k}{\operatorname{argmin}} \sum_{i=1}^{2} v_k^{(i)} L(y_k, g^{(i)}(\phi_k^{(i)}; \theta^{(i)}, \epsilon)) \tag{9}$$

Finally, we marginally increment the age parameters, $\lambda^{(1)}$ and $\lambda^{(2)}$, in order to allow for more unrewarded trajectories to participate in policy training. The process above is performed for each model each at iteration until there are no more unrewarded trajectories to assign pseudo-rewards, the maximum amount of training iterations has been reached, or the model converges.

It is worth pausing for a moment to acknowledge further support behind why the above routine benefits explicitly from Expected SARSA. The DOSPCoT optimization algorithm is iteratively training two policy learners through the back-and-forth exchange of pseudo rewarded trajectories. The rate at which pseudo rewards are correctly applied to trajectories is initially highly erratic. Intuitively, an agent seeking the average reward will be able to better approximate the hidden true reward of an environment than an agent that is always seeking the maximum reward, such as a Q-learner. The maximum reward sought is erroneous when training begins, so a Q-learner is maximizing a lossy policy while an Expected SARSA learner is approximating it by averaging its

observations. In most state-of-the art reinforcement learning research where the environment can easily be observed, Q-learning seems to outperform the other two algorithms in producing the most rewarding policy. However, based on our assessments here, we have evidence to suggest that a Q-learner's performance is contingent on the degree to which it is able to confidently observe the true rewards for its actions.

5. Experiments

In the experiments below, we evaluate the Q-learning, SARSA, and Expected SARSA policy learners to demonstrate the ability of the DOSPCoT framework to aid in policy learning. Our secondary intent is to emphasize the advantage Expected SARSA can provide at the beginning of training with many incorrect pseudo-rewards. We demonstrate all three algorithms over a common RL benchmarking task: Cliff-Walking. To establish reproducibility of results, we leverage the environment for this task directly from Open AI's *gym* python package Brockman, et al. (2016). In performing our experiments, we leveraged the code contributed to the SPamCo repository by Ma et al (2020), making only the changes necessary to allow the repository to be used in a reinforcement learning setting; we strived to maintain parity with the existing repository as much as possible in order to ensure reliable results[1].

5.1. Setup

We evaluate the DOSPCoT framework over the Cliff-Walking problem, now a common toy dataset for reinforcement learning, but formally proposed by Sutton and Barto (2018). The goal of the Cliff-Walking problem is to learn a policy that can navigate the most efficient path from a starting position to a goal while avoiding the "cliff". The optimal path can be seen by the red arrow below; the most rewarding, yet longer path, is indicated by the blue arrow. Any path leading into the cliff results in an extreme reward penalty and defaulting the agent's position back to the start. **Figure 1** shows this problem illustratively.

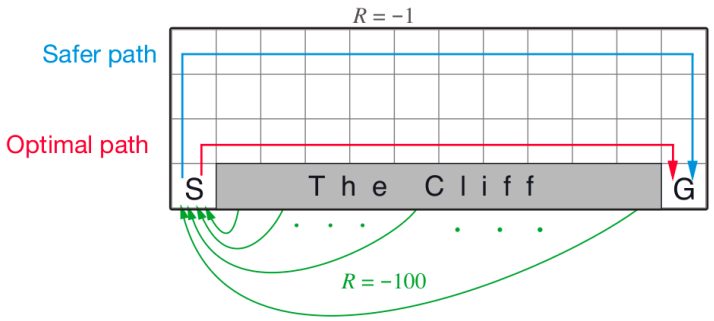

Figure 1 - A diagram of the Cliff-Walking problem from Sutton's and Barto's *Reinforcement Learning (2018)*.

5.2. Results among a Homogenous Model

All experiments were performed in the following manner. Each of Q-learning, SARSA, and Expected SARSA were assessed individually and interactively. First, we establish benchmark results for the accuracy of each algorithm to construct a policy for the optimal path *without* the use of DOSPCoT. Then, we collect results over ten runs for each co-trained pair of algorithms. We show results of each algorithm after only one iteration of the DOSPCoT optimization routine and after ten iterations.

[1] Code for these experiments will be made available at the following url: https://github.kordelfrancetech.com/dospcot.

Ma et al. (2020) suggest a default value of 0.3 for the age parameter γ, so we evaluate this value in the context of two additional values to illustrate the paerameter's effects. We select $\gamma = 0.5$ and $\gamma = 0.8$ as the other two values for the age parameter based on the suggestion of Ma et al. (2020) to begin training with an initial neighborhood around the default value. All learners were trained for 500 iterations with 100 steps per iteration and leveraged a discount factor, exploration rate and learning factor of 0.95, 0.1, and 0.8 respectively. We begin with a greediness 0.1 for Expected SARSA and increased increase it by $(1 - \epsilon)/1000$ after every policy iteration. Note that iterations of DOSPCoT optimization are different than iterations of the policy learner.

Table 1 gives benchmarking results of the Q-learning, SARSA, and Expected SARSA algorithms over the Cliff-Walking environment under the above conditions, but *without* leveraging the DOSPCoT optimization framework. **Table 2** shows the performance of each of the three algorithms over each value for γ, this time utilizing DOSPCoT over a single iteration. **Table 3** communicates similar data to **Table 2,** but shows metrics for five iterations of the DOSPCoT optimization routine instead of one. We do this to show the effects of using more iterations of the optimization algorithm in the training process and to expose any changes that may be conditional on γ. Note that, for this experiment, both models for the dual-observation element are of identical architecture (a homogeneous model) such that a Q-learner is paired with a Q-learner and so on.

Table 1: Baseline Results without DOSPCoT

Algorithm	Accuracy
Q-Learning	0.6250
SARSA	0.7708
Expected SARSA	0.8958

Table 2: One iteration of DOSPCoT

Algorithm	$\gamma = 0.3$	$\gamma = 0.5$	$\gamma = 0.8$	Mean
Q-Q	**0.6410**	0.6319	0.6354	0.6361
SARSA-SARSA	**0.7861**	0.7701	0.7778	0.7780
E.SARSA-E.SARSA	**0.9188**	0.9146	0.9091	0.9142

Table 3: Five iterations of DOSPCoT

Algorithm	$\gamma = 0.3$	$\gamma = 0.5$	$\gamma = 0.8$	Mean
Q-Q	**0.6458**	0.6333	0.6450	0.6414
SARSA-SARSA	**0.9167**	0.7625	0.7917	0.8236
E.SARSA-E.SARSA	**0.9792**	0.9354	0.9271	0.9472

The effects of the age parameter are seen to be relatively negligible between values for only one iteration of DOSPCoT. Intuitively, the true merit of the optimization routine and γ are indicated upon comparing differences between the one-iteration and five-iteration metrics. Between both **Table 2** and **Table 3**, $\gamma = 0.3$ gives the maximum set of accuracies for every model. This provides us with insight in three domains: (1) it is consistent with the advice provided by Ma et al. (2020) in leveraging the default value for the age parameter enabling us to confirm their work, (2) it allows us to have a more confident heuristic on a single tuning value to use moving forward, and (3) it gives us insight into how each agent is interacting with the self-paced learning element of the framework.

All three agents demonstrate increased performance from both co-training and additional iterations of the optimization routine. Unsurprisingly, Expected SARSA received the highest level of performance among the three agents while SARSA demonstrated the highest overall increase in performance from the baseline results. Expected SARSA also benefitted the most from additional optimization time in comparison to its counterparts, confirming our hypothesis that it contains properties that are particularly suited for co-training among very noisy pseudo-rewards. The Q-learning agent remained statistically unchanged throughout each experiment. We do not have enough evidence to suggest that Q-learning can not benefit from co-training to the degree that the other two can, but rather that perhaps we did not find its optimal tuning value. Interestingly, Expected SARSA still seems to capitalize off of a high greediness value that approaches that of the Q-learner's after the first few training iterations while the similar greediness does not seem to benefit the Q-learner.

In summary of the above section, we conclude that there is evidence to support that self-paced co-training and Expected SARSA can enable better policy construction in unobservable environments by enabling the collaboration of two learners with DOSPCoT.

5.3. Results (Interactions) among a Heterogenous Model

In the section above, we demonstrated the ability of two Expected SARSA agents to exhibit superior performance with co-training in comparison to two Q-learning agents or two SARSA agents. However, what happens when Expected SARSA is paired with a different agent type? Perhaps the interaction of the two differing models could collaborate in such a way as to increase policy generalization due to differing "views." We evaluate these effects in **Table 4.** Since we acquired enough evidence from the previous experiment to support that $\gamma = 0.3$ works well among the DOSPCoT training routine, we assess the performance of these interactions with this value only. All other parameters and training specifications are identical to the previous experiment.

Table 4: Expected SARSA with Interactions

Algorithm	$\gamma = 0.3$
E.SARSA-E.SARSA	0.9779
E.SARSA-SARSA	0.9388
E.SARSA-Q	0.9188

Intriguingly, our hypothesis was proved correct[2]. When coupled with a differing model, Expected SARSA seems to inform the other model's policy better than if that model was co-trained with a model of identical type. This speaks to the value of leveraging co-training as a learning technique while providing further evidence to support that Expected SARSA exhibits behavior that can prove advantageous in largely unobservable environments.

We conclude this section with further support of our hypothesis that self-paced co-training and Expected SARSA can enable more efficient policy construction among both homogeneous and heterogeneous policy learners.

[2] The "E.SARSA-E.SARSA" results can be interpreted as a second set of 10 identical runs performed by the previous experiment since both models are Expected SARSA agents. We simply evaluated another set of runs for a fresh baseline and to provide further confirmation of the performance observed in the previous experiment with homogenous models.

6. Future Work

The toy problems evaluated above, while demonstrating feasibility, do not fully justify the potential of DOSPCoT. Performance would be better assessed over datasets with much higher dimensionality. Our interest in co-training stems from our interest in swarm learning and collaboratively reasoning about an unknown environment. To this point, a more representative problem is needed in order to truly assess our hypothesis and claims in this context. By increasing the dimensionality of the problem, we also enable the use of neural networks in deep reinforcement learning.

To this end, there are some other natural extensions of this work that the authors intend to pursue in order to reconcile the above.

6.1. Varying Trajectories

Our framework evaluates trajectories for single state-action pairs. In other words, DOSPCoT aims to train a policy that can obtain the highest award by observing single state-action sequences from multiple agents. There is obvious curiosity in analyzing the algorithm's affinity to produce similar (or better) results with trajectories of longer length. For example, a trajectory $\phi_i \in \Phi$ could indicate the 5-tuple of $\{s_{i-1}, a_{i-1}, r_{i-1}, s_i, a_i\}$ where s_i, a_i, and r_i pertain to the state, action, and reward of the i^{th} trajectory, respectively. Pseudo-rewards would then be assigned to the 5-tuple instead of the 2-tuple state-action pair analyzed here. Furthermore, one might investigate a strategy that accepts trajectories of variable lengths to provide additional robustness in policy construction.

6.2. Extension to Inverse Reinforcement Learning and Imitation Learning

We evaluate the the DOSPCoT framework under the context of traditional reinforcement learning where we construct a policy that can learn a reward function. However, DOSPCoT can be tuned to easily accommodate both inverse reinforcement learning and imitation learning.

With IRL, one does not attempt to directly learn a policy that maps states to actions. Instead, IRL attempts to learn a reward function through observation of expert demonstrations. DOSPCoT can be leveraged to construct this reward function in a similar manner as shown by our work here, but with the exception being that trajectories would be generated from an expert policy. IL also assumes trajectories come from an expert policy, but the goal is to learn this policy directly by finding actions that minimize the loss between the learner and the expert (e.g. minimizing the regret). Learning the policy is then effectively just a supervised learning problem which can be mapped to original co-training work proposed by Blum and Mitchell (1998) and which is well demonstrated in a MDP-based context by Song et al. (2020).

Chen et al. (2021) provide very intriguing work regarding decision transformers for reinforcement learning that lends itself well to co-training methodologies. To accommodate a decision transformer, one might input longer trajectories into DOSPCoT and simply mask certain states, actions, and rewards of the trajectory in order to train the attention heads of the transformer. While Zhang et al. (2021) provide support for co-training transformers, there is minimal work being performed at the intersection of co-training, transformers and reinforcement learning based on our research; this is particularly motivating as a next step for DOSPCoT.

6.3. Multi-Observation Self-Paced Policy Co-Training

This work sets up the framework for leveraging policy construction with co-training among only two trajectory observations. However, a very natural extension of the framework would be the adaptation to $m \in \{3, 4, \ldots n\}$ trajectory observations such that any number of agents can be trained simultaneously. In particle swarm optimization, one can imagine how co-training a policy among more than two agents would be highly advantageous. In a future work, we hope to

demonstrate the feasibility of this idea along with the potential to decrease the training time of learning the policy as a function of the number of agents. In other words, we hypothesize that the more agents that are added, the faster a policy can be constructed in a *multi-observation self-paced policy co-training* framework.

7. Concluding Remarks

From our research above, self-paced co-training shows definite potential in being able to accelerate policy learning in an environment whose observation space contains a large degree of ambiguity. Our analysis shows that Expected SARSA couples well with the self-paced aspect of co-training and allows for a smoother learning schedule in comparison to the other policy learners. We evaluated the performance of DOSPCoT over a well-known benchmarking dataset for reinforcement learning and compare the results to standard two other standard reinforcement algorithms. In summary, we established strong support of our initially proposed hypothesis that self-paced co-training can be extended to allow for more efficient policy construction in reinforcement learning. Through future work, we hope to extend our methods to more than two trajectory observations as well as to inverse reinforcement learning and imitation learning. Our work is motivated by the interest in training autonomous agents to act collaboratively in unobservable environments and we hope that DOSPCoT inspires other co-training methods within this context.

Code for our experiments and algorithmic analyses will be made available at the following url: https://github.kordelfrancetech.com/dospcot.

8. References

Vidya Muthukumar, “Learning from an Unknown Environment”, PhD thesis, University of California at Berkeley, 2020.

Jialin Song, Ravi Lanka, Yisong Yue, Masahiro Ono. Co-training for Policy Learning. *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, PMLR 115:1191-1201, 2020.

Vidya Muthukumar, Soham Phade, Anant Sahai. On the Impossibility of Convergence of Mixed Strategies with No Regret Learning. *arXiv preprint arXiv:2012.02125v3,* 2022.

Fan Ma, Deyu Meng, Xuany Dong, and Yi Yang. Self-paced Multi-view Co-training. *Journal of Machine Learning Research, pages 1-38, 2020.*

Fan Ma, Deyu Meng, Qi Xie, Zina Li, and Xuany Dong. Self-paced Co-training. *Journal of Machine Learning Research, pages 2275-2284, 2017.*

Jiawei Wu, Lei Li, and William Yang Wang. 2018. Reinforced Co-Training. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, *pages 1252–1262,* New Orleans, Louisiana. Association for Computational Linguistics.

Richard S. Sutton, Andrew G. Barto. Reinforcement Learning. Second Edition. 2018. The MIT Press.

Russel, Stuart J.; Norvig, Peter. Artificial Intelligence: A Modern Approach. Third Edition. 2015, Pearson India Education Services Pvt. Ltd., pages 961-962.

Ciosek, Kamil. Imitation Learning by Reinforcement Learning. *The International Conference on Learning Representations,* 2022.

Wang, Ziyu; et al. Dueling Network Architectures for Deep Reinforcement Learning. The International Conference on Machine Learning, 2016.

Hester, Todd; et al. Deep Q-learning from Demonstrations. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence.* February 2018 Article No.: 394, pages 3223–3230.

Peter Abbeel, Andrew Y. Ng. Apprenticeship Learning via Inverse Reinforcement Learning. *Proceedings of the 21st International Conference on Machine Learning (ICML)*. 2004.

Ashlesha Akella, Chin-Teng Lin. Time and Action Co-Training in Reinforcement Learning Agents. *Frontiers in Control Engineering, Volume 2, Article 722092*. 6 August 2021.

Maria-Florina Balcan and Avrim Blum. A discriminative model for semi-supervised learning. J. ACM, 57:19:1–19:46, 2010.

Maria-Florina Balcan, Avrim Blum, and Ke Yang. Co-training and expansion: Towards bridging theory and practice. *Advances in neural information processing systems*, pages 89–96, 2004.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

Ulf Brefeld and Tobias Scheffer. Co-em support vector learning. *Proceedings of the twenty-first international conference on Machine learning, page 16.* ACM, 2004.

Xuanyi Dong, Deyu Meng, Fan Ma, and Yi Yang. A dual-network progressive approach to weakly supervised object detection. In Proceedings of the 2017 ACM on Multimedia Conference, MM ’17, pages 279–287, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4906-2.

Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, and Deyu Meng. Few-example object detection with model communication. IEEE transactions on pattern analysis and machine intelligence, 41(7):1641–1654, 2018.

Vikas Sindhwani and David S Rosenberg. An rkhs for multi-view learning and manifold co-regularization. *Proceedings of the 25th international conference on Machine learning, pages 976–983.* ACM, 2008.

Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. Beyond the point cloud: from transductive to semi-supervised learning. *International Conference on Machine Learning, pages 824–831.* 2005a.

Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. A co-regularization approach to semi-supervised learning with multiple views. *Proceedings of ICML workshop on learning*

with multiple views, pages 74–79. Citeseer, 2005b.

Zongmo Huang, Yazhou, Ren, Xiaorong Pu, et al. Dual self-paced multi-view clustering. *Elsevier - Neural Networks, Volume 140, pages 184-192*. August 2021.

Chang Xu, Dacheng Tao, Chao Xu. Multi-View Self-Paced Learning for Clustering. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, pages 3974-3980*. 2015.

Ping Wang, Jizong Peng, Marco Pedersoli, et al. Self-paced and self-consistent co-training for semi-supervised image segmentation. *Elsevier - Medical Image Analysis, Volume 73*. 2021.

Yingda Xia, Dong Yang, Zhiding Yu, et al. Uncertainty-aware multi-view co-training for semi- supervised medical image segmentation and domain adaptation. *Elsevier - Medical Image Analysis, Volume 65*. 2020.

Lili Chen, Kevin Lu, Aravind Rajeswaran, et al. "Decision Transformer: Reinforcement Learning via Sequence Modeling". *arXiv preprint arXiv:2106.01345*. 2021.

Bowen Zhang, Jiahui Yu, Christopher Fifty, et al. "Co-training Transformer with Videos and Images Improves Action Recognition". *arXiv preprint arXiv:2112.07175*. 2021.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, et al. "OpenAI Gym". *arXiv preprint arXiv:1606.01540v1*. 2016.