is used in our experiments.

RAIL is similar in spirit to active forward training, though its analysis is quite different and more involved. Like forward-training RAIL iterates for $T$ iterations, but on each iteration, RAIL learns a new stationary policy $\hat{\pi}^t$ that can be applied across all time steps. Iteration $t+1$ of RAIL learns a new policy $\hat{\pi}^{t+1}$ that achieves a low error rate at predicting the expert’s actions with respect to the state distribution of the previous policy $d_{\pi^t}$. More formally, given an i.i.d. active learner $L_a$, the RAIL algorithm is defined by the following iteration:

- **(Initialize)** $\hat{\pi}^0$ is an arbitrary policy, possibly based on prior knowledge or existing data,
- **(Iterate)** $t = 1, \ldots, T$ \hspace{1em} $\hat{\pi}^t = L_a(\epsilon, \delta, d_{\pi^{t-1}})$.

Thus, similar to active forward training, RAIL makes a sequence of $T$ calls to an active learner. Unlike forward training, however, the unlabeled data distributions used at each iteration contains states from all time points within the horizon, rather than being restricted to a states arising at a particular time point. Because of this difference, the active learner is able to ask queries across a range of time point and we might expect policies learned in earlier iterations to achieve non-trivial performance throughout the entire horizon. In contrast, at iteration $t$ the policy produced by forward training is only well defined up to time $t$.

The complication faced by RAIL, however, compared to forward training, is that the distribution used to train $\hat{\pi}^{t+1}$ differs from the state distribution of the expert policy $d_{\pi^t}$. This is particularly true in early iterations of RAIL since $\hat{\pi}^0$ is initialized arbitrarily. We now show that as the iterations proceed, we can bound the similarity between the state distributions of the learned policy and the expert, which allows us to bound the regret of the learned policy. We first state the main result which we prove below.

**Theorem 1.** Given a PAC i.i.d. active learning algorithm $L_a$, if RAIL is run with parameters $\epsilon$ and $\frac{\delta}{T}$ passed to $L_a$ at each iteration, then with probability at least $1 - \delta$ it will return a stationary policy $\hat{\pi}^T$ such that $V(\hat{\pi}^T) \geq V(\pi^*) - cT^3$.

From this we see that the impact of moving from non-stationary to stationary policies in the worst case is a factor of $T$ in the regret bound. Similarly the bound is a factor of $T$ worse than the comparable result above for passive imitation learning, which suffered a worst-case regret of $cT^2$. From this we see that the total label complexity for RAIL required to guarantee a regret of $\epsilon$ with probability $1 - \delta$ is $T \cdot N_a(\frac{T}{\epsilon}, \frac{\delta}{T})$ compared to the above label complexity of passive learning $T \cdot N_\epsilon(\frac{T}{\epsilon}, \frac{\delta}{T})$. Thus, for a given policy class, if the label complexity of i.i.d. active learning is substantially less than the label complexity of passive learning, then our reduction can leverage those savings. For example, in the realizable learning case, ignore the dependence on $\delta$ (which is only logarithmic), we get an active label complexity of $T \cdot O(\log \frac{T}{\epsilon})$ versus the corresponding passive complexity of $T \cdot O(\frac{\log T}{\epsilon})$.

For the proof we introduce the quantity $P^T_\pi(M)$, which is the probability that a policy $\pi$ is consistent with a length $T$ trajectory generated by the expert policy $\pi^*$ in MDP $M$. It will also be useful to index the state distribution of $\pi$ by the MDP $M$, denoted by $d_\pi(M)$. The main idea is to show that at iteration $t$, $P^T_\pi(M)$ is not too small, meaning that the policy at iteration $t$ mostly agrees with the expert for the first $t$ actions. We first state two lemmas, which are useful for the final proof. First, we bound the regret of a policy in terms of $P^T_\pi(M)$.

**Lemma 1.** For any policy $\pi$, if $P^T_\pi(M) \geq 1 - \epsilon$, then $V(\pi) \geq V(\pi^*) - cT$.

**Proof.** Let $\Gamma^*$ and $\Gamma$ be all state-action sequences of length $T$ that are consistent with $\pi^*$ and $\pi$ respectively. If $R(T)$ is the total reward for a sequence $T$ then we get the following:

\[
V(\pi) = \sum_{T \in \Gamma} \Pr(T \mid M, \pi) R(T) \\
\geq \sum_{T \in \Gamma^*} \Pr(T \mid M, \pi^*) R(T) - \sum_{T \in \Gamma \cap \Gamma^*} \Pr(T \mid M, \pi^*) R(T) \\
= V(\pi^*) - \sum_{T \in \Gamma \cap \Gamma^*} \Pr(T \mid M, \pi^*) R(T) \\
\geq V(\pi^*) - \epsilon T.
\]

The last two inequalities follow since the reward for a sequence must be no more than $T$ and our assumption about $P^T_\pi(M)$.

Next, we show how the value of $P^T_\pi(M)$ changes across one iteration of learning.

**Lemma 2.** For any policies $\pi$ and $\hat{\pi}$ and $1 \leq t < T$, if $e_{\pi^t}(\hat{\pi}, d_\pi(M)) \leq \epsilon$, then $P^{t+1}_\pi(M) \geq P^t_\pi(M) - T\epsilon$.

**Proof.** We define $\hat{\Gamma}$ to be all sequences of state-action pairs of length $t+1$ that are consistent with $\hat{\pi}$. Also define $\Gamma$ to be all length $t+1$ state-action sequences that are consistent with $\pi$ on the first $t$ state-action pairs (so need not be consistent on the final pair). We also define $M$ to be an MDP that is identical to $M$, except that the transition distribution of any state-action pair $(s,a)$ is equal to the transition distribution