of action \( \pi(s) \) in state \( s \). That is, all actions taken in a state \( s \) behave like the action selected by \( \pi \) in \( s \).

We start by arguing that if \( \epsilon_{\pi^*}(\hat{\pi}, \hat{d}_s(M)) \leq \epsilon \) then \( P_{\pi}^{t+1}(M) \geq 1 - T\epsilon \), which relates our error assumption to the MDP \( \hat{M} \). To see this note that for MDP \( M \), all policies including \( \pi^* \), have state distribution given by \( d_{\pi^*} \). Thus by the union bound \( 1 - P_{\pi}^{t+1}(M) \leq \sum_{i=1}^{t+1} \epsilon_i \), where \( \epsilon_i \) is the error of \( \hat{\pi} \) at predicting \( \pi^* \) on distribution \( d_{\pi^*} \). This sum is bounded by \( T\epsilon \) since \( \epsilon_{\pi^*}(\hat{\pi}, \hat{d}_s(M)) = \frac{1}{T} \sum_{i=1}^{T} \epsilon_i \). Using this fact we can now derive the following:

\[
P_{\pi}^{t+1}(M) = \sum_{T \in \Gamma} \Pr(T \mid M, \pi^*) \\
\geq \sum_{T \in \Gamma_{\pi^*}} \Pr(T \mid M, \pi^*) \\
= \sum_{T \in \Gamma} \Pr(T \mid M, \pi^*) - \sum_{T \in \Gamma_{\pi^*}} \Pr(T \mid M, \pi^*) \\
= P_{\pi^*}^t(M) - \sum_{T \in \Gamma_{\pi^*}} \Pr(T \mid M, \pi^*) \\
= P_{\pi^*}^t(M) - \frac{1}{T} \sum_{T \in \Gamma} \Pr(T \mid M, \pi^*) \\
\geq P_{\pi^*}^t(M) - T\epsilon.
\]

The equality of the fourth line follows since \( \Gamma \) contains all sequences whose first \( t \) actions are consistent with \( \pi \) with all possible combinations of the remaining action and state transition. Thus, summing over all such sequences yields the probability that \( \pi^* \) agrees with the first \( t \) steps. The equality of the fifth line follows because \( \Pr(T \mid M, \pi^*) = \Pr(T \mid \hat{M}, \pi^*) \) for any \( T \) that is in \( \Gamma \) and for which \( \pi^* \) is consistent (has non-zero probability under \( \pi^* \)). The final line follows from the above observation that \( P_{\pi}^{t+1}(M) \geq 1 - T\epsilon \).

We can now complete the proof of the main theorem.

**Proof of Theorem 1.** Using failure parameter \( \delta \) ensures that with at least probability \( 1 - \delta \) that for all \( 1 \leq t \leq T \) we will have \( \epsilon_{\pi^*}(\hat{\pi}^{t+1}, d_{\pi^*}(M)) \leq \epsilon \). As a base case, we have \( P_{\pi^*}^{t+1} \geq 1 - T\epsilon \), since the the error rate of \( \hat{\pi}^1 \) relative to the initial state distribution at time step \( t = 1 \) is at most \( T\epsilon \). Combining these facts with Lemma 2 we get that \( P_{\pi^*}^{t+1} \geq 1 - \epsilon T^2 \). Combining this with Lemma 1 completes the proof.

\[\square\]

### 4.3 Agnostic Case

Above we considered the realizable setting, where the expert’s policy was assumed to be in a known hypothesis class \( H \). In the agnostic case, we do not make such an assumption. The learner still outputs a hypothesis from a class \( H \), but the unknown policy is not necessarily in \( H \). The agnostic i.i.d. PAC learning setting is defined similarly to the realizable setting, except that rather than achieving a specified error bound of \( \epsilon \) with high probability, a learner must guarantee an error bound of \( \inf_{\pi \in H} \epsilon_f(\pi, D_X) + \epsilon \) with high probability (where \( f \) is the target), where \( D_X \) is the unknown data distribution. That is, the learner is able to achieve close to the best possible accuracy given class \( H \). In the agnostic case, it has been shown that exponential improvement in label complexity with respect to \( \frac{1}{\epsilon} \) is achievable when \( \inf_{\pi \in H} \epsilon_f(\pi, D_X) \) is relatively small compared to \( \epsilon \) (Dasgupta, 2011). Further, there are many empirical results for practical active learning algorithms that demonstrate improved label complexity compared to passive learning.

It is straightforward to extend our above results for non-stationary and stationary policies to the agnostic case by using agnostic PAC learners for \( L_p \) and \( L_a \). Space precludes full details and here we outline the extension for RAIL. Note that the RAIL algorithm will call \( L_a \) using a sequence of unlabeled data distributions, where each distribution is of the form \( d_\pi \) for some \( \pi \in H \) and each of which may yield a different minimum error given \( H \). For this purpose, we define \( \epsilon^* = \sup_{\pi \in H} \inf_{\pi \in H} \epsilon_{\pi^*}(\hat{\pi}, d_\pi) \) to be the minimum generalization error achievable in the worst case considering all possible state distributions \( d_\pi \) that RAIL might possibly encounter. With minimal changes to the proof of Theorem 1, we can get an identical result, except that the regret is \( (\epsilon^* + \epsilon)T^3 \) rather than just \( \epsilon T^3 \). A similar change in regret holds for passive imitation learning. This shows that in the agnostic setting we can get significant improvements in label complexity via active imitation learning when there are significant savings in the i.i.d. case.

### 5 Practical Instantiation of RAIL

Despite the theoretical guarantees, a potential drawback of RAIL is that the unlabeled state distributions used at early iterations may be quite different from \( d_{\pi^*} \). In particular, at iteration \( t \), the distributions at times \( t' > t \) have no guarantees with respect to \( d_{\pi^*} \). Thus, early iterations may focus substantial query effort on parts of the state-space that are not relevant to \( \pi^* \). Another issue is that the idealized version does not share data across iterations, which is potentially wasteful. We now describe our practical instantiation of RAIL, called RAIL-DW (for density weighted), which addresses these issues in several ways.

First, we use an incremental version of RAIL that asks only one query per iteration and accumulates data across iterations. This allows rapid updating of state distributions and prevents RAIL from wasting its query budget on earlier inaccurate distributions but rather focus on later more accurate distributions. Second, recall that at iteration \( t + 1 \), RAIL learns using an unlabeled data distribution \( d_{\hat{\pi}^t} \), where \( \hat{\pi}^t \) is the policy learned at iteration \( t \). In order to help improve the