Appendix

A. A relation between AggreVaTeD with Natural Gradient and AggreVaTe with Weighted Majority

First, we show that Weighted Majority can be leveraged for imitation learning in discrete MDPs. Then we extend Weighted Majority to continuous MDPs, where we show that, with three steps of first-order approximations, WM leads to a natural gradient update procedure.

A.1. Weighted Majority in Discrete MDPs

For notation simplicity, for each state $s \in \mathcal{S}$, we represent the policy $\pi(\cdot|s)$ as a discrete probability vector $\pi^s \in \Delta(A)$. We also represent d_t^{π} as a S-dimension probability vector from S-d simplex, consisting of $d_t^{\pi}(s), \forall s \in \mathcal{S}$. For each s, we use $Q_t^*(s)$ to denote the A-dimension vector consisting of the state-action cost-to-go $Q_t^*(s,a)$ for all $a \in \mathcal{A}$. With this notation, the loss function $\ell_n(\pi)$ from Eq. 1 can now be written as: $\ell_n(\pi) = \frac{1}{H}\sum_{t=1}^{H}\sum_{s\in\mathcal{S}} d_t^{\pi_n}(s)(\pi^s \cdot Q_t^*(s))$, where $a \cdot b$ represents the inner product between vectors a and b. Weighted Majority updates π as follows: $\forall s \in \mathcal{S}$,

$$\begin{aligned}\pi_{n+1} = \mathop{\arg\min}_{\pi^s\in\Delta(A),\forall s\in\mathcal{S}} & \frac{1}{H}\sum_{t=1}^{H}\sum_{s\in\mathcal{S}} d_t^{\pi_n}(s)\big(\pi^s \cdot Q_t^*(s)\big) \\ & + \sum_{s\in\mathcal{S}} \frac{\bar{d}^{\pi_n}(s)}{\eta_{n,s}} KL(\pi_s \| \pi_n^s),\end{aligned} \tag{16}$$

where $KL(q\|p)$ is the KL-divergence between two probability distributions q and p. This leads to the following closed-form update:

$$\pi_{n+1}^s[i] = \frac{\pi_n^s[i]\exp\big(-\eta_{n,s}\tilde{Q}_s^e[i]\big)}{\sum_{j=1}^{|\mathcal{A}|}\pi_n^s[j]\exp\big(-\eta_{n,s}\tilde{Q}_s^e[j]\big)}, i \in [|\mathcal{A}|], \tag{17}$$

where $\tilde{Q}_s^e = \sum_{t=1}^{H} d_t^{\pi_n}(s)Q_t^*(s)/(H\bar{d}^{\pi_n}(s))$. We refer readers to (Shalev-Shwartz et al., 2012) or Appendix C for the derivations of the above closed-form updates.

A.2. From Discrete to Continuous

We now consider how to update the parametrized policy π_θ for continuous MDPs. Replacing summations by integrals, Eq. 16 can be written as:

$$\begin{aligned}\theta = \mathop{\arg\min}_{\theta} & \frac{1}{H}\sum_{t=1}^{H} \mathop{\mathbb{E}}_{s\sim d_t^{\pi_{\theta_n}}} \mathop{\mathbb{E}}_{a\sim\pi(\cdot|s;\theta)}[Q_t^*(s,a)] \\ & + \mathop{\mathbb{E}}_{s\sim\bar{d}^{\pi_{\theta_n}}} KL(\pi_\theta||\pi_{\theta_n})/\eta_n.\end{aligned} \tag{18}$$

In order to solve for θ from Eq. 18, we apply several first-order approximations. We first approximate $\ell_n(\theta)$ (the first part of the RHS of the above equation) by its first-order Taylor expansion: $\ell_n(\theta) \approx \ell_n(\theta_n) + \nabla_{\theta_n}\ell_n(\theta_n)\cdot(\theta - \theta_n)$. When θ and θ_n are close, this is a valid local approximation.

Second, we replace $KL(\pi_\theta||\pi_{\theta_n})$ by $KL(\pi_{\theta_n}||\pi_\theta)$, which is a local approximation since $KL(q||p)$ and $KL(p||q)$ are equal up to the second order (Kakade & Langford, 2002; Schulman et al., 2015).

Third, we approximate $KL(\pi_{\theta_n}||\pi_\theta)$ by a second-order Taylor expansion around θ_n, such that we can approximate the penalization using the Fisher information matrix:

$$\mathop{\mathbb{E}}_{s\sim\bar{d}^{\pi_{\theta_n}}} KL(\pi_{\theta_n}||\pi_\theta) \approx (1/2)(\theta-\theta_n)^T I(\theta_n)(\theta-\theta_n),$$

where the Fisher information matrix $I(\theta_n) = \mathbb{E}_{s,a\sim\bar{d}^{\pi_{\theta_n}}\pi_{\theta_n}(a|s)}\big(\nabla_{\theta_n}\log(\pi_{\theta_n}(a|s))\big)\big(\nabla_{\theta_n}\log(\pi_{\theta_n}(a|s)\big)^T$.

Inserting these three approximations into Eq. 18, and solving for θ, we reach the following update rule $\theta_{n+1} = \theta_n - \eta_n I(\theta_n)^{-1}\nabla_\theta \ell_n(\theta)|_{\theta=\theta_n}$, which is similar to the natural gradient update rule developed in (Kakade, 2002) for the RL setting. Bagnell & Schneider (2003) provided an equivalent representation for Fisher information matrix:

$$I(\theta_n) = \frac{1}{H^2}\mathop{\mathbb{E}}_{\tau\sim\rho_{\pi_{\theta_n}}} \nabla_{\theta_n}\log(\rho_{\pi_{\theta_n}}(\tau))\nabla_{\theta_n}\log(\rho_{\pi_{\theta_n}}(\tau))^T, \tag{19}$$

where $\nabla_\theta \log(\rho_{\pi_\tau}(\tau))$ is the gradient of the log likelihood of the trajectory τ which can be computed as $\sum_{t=1}^{H}\nabla_\theta \log(\pi_\theta(a_t|s_t))$. In the remainder of the paper, we use this Fisher information matrix representation, which yields much faster computation of the descent direction δ_θ, as we will explain in the next section.

B. Derivation of Eq. 4

Starting from Eq. 1 with parametrized policy π_θ, we have:

$$\begin{aligned}
\ell_n(\theta) &= \frac{1}{H}\sum_{t=1}^{H}\mathop{\mathbb{E}}_{s_t\sim d_t^{\pi_{\theta_n}}}\Big[\mathop{\mathbb{E}}_{a_t\sim\pi(\cdot|s_t;\theta)}[Q_t^*(s_t,a_t)]\Big]\\
&= \frac{1}{H}\sum_{t=1}^{H}\mathop{\mathbb{E}}_{s_t\sim d_t^{\pi_{\theta_n}}}\Big[\int_a \pi(a|s_t;\theta)Q_t^*(s_t,a)da\Big]\\
&= \frac{1}{H}\sum_{t=1}^{H}\mathop{\mathbb{E}}_{s_t\sim d_t^{\pi_{\theta_n}}}\Big[\int_a \pi(a|s_t;\theta_n)\frac{\pi(a|s_t;\theta)}{\pi(a|s_t;\theta_n)}Q_t^*(s_t,a)da\Big]\\
&= \frac{1}{H}\sum_{t=1}^{H}\mathop{\mathbb{E}}_{s_t\sim d_t^{\pi_{\theta_n}}}\Big[\mathop{\mathbb{E}}_{a\sim\pi(\cdot|s_t;\theta_n)}\frac{\pi(a|s_t;\theta)}{\pi(a|s_t;\theta_n)}Q_t^*(s_t,a)\Big]\\
&= \frac{1}{H}\sum_{t=1}^{H}\mathop{\mathbb{E}}_{s_t\sim d_t^{\pi_{\theta_n}},a_t\sim\pi(a|s_t;\theta_n)}\Big[\frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta_n)}Q_t^*(s_t,a_t)\Big].
\end{aligned} \tag{20}$$

C. Derivation of Weighted Majority Update in Discrete MDP

We show the detailed derivation of Eq. 17 for AggreVaTeD with WM in discrete MDP. Recall that with KL-divergence as the penalization, one update the policy in each episode as:

$$\{\pi_{n+1}^s\}_{s\in\mathcal{S}} = \arg\min_{\{\pi^s\in\Delta(A),\forall s\}}\frac{1}{H}\sum_{t=1}^{H}\sum_{s\in\mathcal{S}} d_t^{\pi_n}(s)\big(\pi^s\cdot Q_t^*(s)\big) + \sum_{s\sim\mathcal{S}}\frac{\bar{d}^{\pi_n}(s)}{\eta_{n,s}}KL(\pi_s\|\pi_n^s)$$

Note that in the above equation, for a particular state s, optimizing π^s is in fact independent of $\pi^{s'}, \forall s'\neq s$. Hence the optimal sequence $\{\pi^s\}_{s\in\mathcal{S}}$ can be achieved by optimizing π^s independently for each $s\in\mathcal{S}$. For π^s, we have the following update rule:

$$\begin{aligned}
\pi_{n+1}^s &= \arg\min_{\pi^s\in\Delta(A)}\frac{1}{H}\sum_{t=1}^{H}d_t^{\pi_n}(s)(\pi^s\cdot Q_t^*(s)) + \frac{\bar{d}^{\pi_n}(s)}{\eta_{n,s}}KL(\pi_s\|\pi_n^s)\\
&= \arg\min_{\pi^s\in\Delta(A)}\pi^s\cdot(\sum_{t=1}^{H}d_t^{\pi_n}(s)Q_t^*(s)/H) + \frac{\bar{d}^{\pi_n}(s)}{\eta_{n,s}}KL(\pi_s\|\pi_n^s)\\
&= \arg\min_{\pi^s\in\Delta(A)}\pi^s\cdot(\sum_{t=1}^{H}d_t^{\pi_n}(s)Q_t^*(s)/(H\bar{d}^{\pi_n}(s))) + \frac{1}{\eta_{n,s}}KL(\pi_s\|\pi_n^s)\\
&= \arg\min_{\pi^s\in\Delta(A)}\pi^s\cdot\tilde{Q}^e(s) + \frac{1}{\eta_{n,s}}\sum_{j=1}^{A}\pi^s[j](\log(\pi^s[j]) - \log(\pi_n^s[j]))
\end{aligned} \tag{21}$$

Take the derivative with respect to $\pi^s[j]$, and set it to zero, we get:

$$\tilde{Q}^e(s)[j] + \frac{1}{\eta_{n,s}}(\log(\pi^s[j]/\pi_n^s[j]) + 1) = 0, \tag{22}$$

this gives us:

$$\pi^s[j] = \pi_n^s[j]\exp(-\eta_{n,s}\tilde{Q}^e(s)[j] - 1). \tag{23}$$

Since $\pi^s \in \Delta(A)$, after normalization, we get:

$$\pi^s[j] = \frac{\pi_n^s[j]\exp(-\eta_{n,s}\tilde{Q}^e(s)[j])}{\sum_{i=1}^{A}\pi_n^s[i]\exp(-\eta_{n,s}\tilde{Q}^e(s)[i])} \tag{24}$$

D. Lemmas

Before proving the theorems, we first present the *Performance Difference Lemma* (Kakade & Langford, 2002; Ross & Bagnell, 2014) which will be used later:

Lemma D.1. *For any two policies π_1 and π_2, we have:*

$$\mu(\pi_1) - \mu(\pi_2) = H\sum_{t=1}^{H}\mathbb{E}_{s_t\sim d_t^{\pi_1}}\left[\mathbb{E}_{a_t\sim\pi_1(\cdot|s_t)}[Q_t^{\pi_2}(s_t,a_t) - V_t^{\pi_2}(s_t)]\right]. \tag{25}$$

We refer readers to (Ross & Bagnell, 2014) for the detailed proof of the above lemma.

The second known result we will use is the analysis of Weighted Majority Algorithm. Let us define the linear loss function as $\ell_n(w) = w\cdot y_n$, for any $y_n \in \mathbb{R}^d$, and $w \in \Delta(d)$ from a probability simplex. Running Weighted Majority Algorithm on the sequence of losses $\{w\cdot y_n\}$ to compute a sequence of decisions $\{w_n\}$, we have:

Lemma D.2. *The sequence of decisions $\{w_n\}$ computed by running Weighted Majority with step size μ on the loss functions $\{w\cdot y_n\}$ has the following regret bound:*

$$\sum_{n=1}^{N} w_n\cdot y_n - \min_{w^*\in\Delta(d)}\sum_{n=1}^{N} w^*\cdot y_n \leq \frac{\ln(d)}{\mu} + \frac{\mu}{2}\sum_{n=1}^{N}\sum_{i=1}^{d} w_n[i]y_n[i]^2. \tag{26}$$

We refer readers to (Shalev-Shwartz et al., 2012) for detailed proof.

E. Proof of Theorem 5.1

Proof. We construct a reduction from stochastic Multi-Arm Bandits (MAB) to the MDP $\tilde{\mathcal{M}}$. A stochastic MAB is defined by S arms denoted as $I^1, ..., I^S$. Each arm I^t's cost c_i at any time step t is sampled from a fixed but unknown distribution. A bandit algorithm picks an arm I_t at iteration t and then receives an unbiased sample of the picked arm's cost c_{I_t}. For any bandit algorithm that picks arms $I_1, I_2, ..., I_N$ in N rounds, the expected regret is defined as:

$$\mathbb{E}[R_N] = \mathbb{E}[\sum_{n=1}^{N} c_{I_n}] - \min_{i\in[S]}\sum_{n=1}^{N}\bar{c}_i, \tag{27}$$

where the expectation is taken with respect to the randomness of the cost sampling process and possibly the randomness of the bandit algorithm. It has been shown that there exists a set of distributions from which the arms' costs sampled from, the expected regret $\mathbb{E}[R_N]$ is at least $\Omega(\sqrt{SN})$ (Bubeck et al., 2012).

Consider a MAB with 2^K arms. To construct a MDP from a MAP, we construct a $K+1$-depth binary-tree structure MDP with $2^{K+1}-1$ nodes. We set each node in the binary tree as a state in the MDP. The number of actions of the MDP is two, which corresponds to go left or right at a node in the binary tree. We associate each leaf nodes with arms in the original MAB: the cost of the i'th leaf node is sampled from the cost distribution for the i'th arm, while the non-leaf nodes have cost always equal to zero. The initial distribution ρ_0 concentrates on the root of the binary tree. Note that there are total 2^K trajectories from the root to leafs, and we denote them as $\tau_1, ...\tau_{2^K}$. We consider finite horizon ($H = K+1$) episodic RL algorithms that outputs $\pi_1, \pi_2, ..., \pi_N$ at N episodes, where π_n is any deterministic policy that maps a node to actions *left* or *right*. Any RL algorithm must have the following regret lower bound:

$$\mathbb{E}[\sum_{n=1}^{N} \mu(\pi_n)] - \min_{\pi^*} \sum_{n=1}^{N} \mu(\pi^*) \geq \Omega(\sqrt{SN}), \tag{28}$$

where the expectation is taken with respect to the possible randomness of the RL algorithms. Note that any deterministic policy π identifies a trajectory in the binary tree when rolling out from the root. The optimal policy π^* simply corresponds to the trajectory that leads to the leaf with the mininum expected cost. Note that each trajectory is associated with an arm from the original MAB, and the expected total cost of a trajectory corresponds to the expected cost of the associated arm. Hence if there exists an RL algorithm that achieves regret $O(\sqrt{SN})$, then we can solve the original MAB problem by simply running the RL algorithm on the constructed MDP. Since the lower bound for MAB is $\Omega(\sqrt{SN})$, this concludes that Eq. 28 holds. □

F. Proof of Theorem 5.2

Proof. For notation simplicity we denote a_l as the go-left action while a_r is the go-right action. Without loss of generality, we assume that the leftmost trajectory has the lowest total cost (e.g., s_3 in Fig. 1 has the lowest average cost). We consider the deterministic policy class Π that contains all policy $\pi : \mathcal{S} \to \{a_l, a_r\}$. Since there are S states and 2 actions, the total number of policies in the policy class is 2^S. To prove the upper bound $R_N \leq O(\log(S))$, we claim that for any $e \leq K$, at the end of episode e, AggreVaTe with FTL identifies the e'th state on the best trajectory, i,e, the leftmost trajectory $s_0, s_1, s_3, ..., s_{(2^{K-1}-1)}$. We can prove the claim by induction.

At episode $e = 1$, based on the initial policy, AggreVaTe picks a trajectory τ_1 to explore. AggreVaTe with FTL collects the states s at τ_1 and their associated cost-to-go vectors $[Q^*(s, a_l), Q^*(s, a_r)]$. Let us denote D_1 as the dataset that contains the state,cost-to-go pairs: $D_1 = \{(s, [Q^*(s, a_l), Q^*(s, a_l)])\}$, for $s \in \tau_1$. Since s_0 is visited, the state-cost pair $(s_0, [Q^*(s_0, a_l), Q^*(s_0, a_r)])$ must be in D_1. To update policy from π_1 to π_2, AggreVaTe with FTL runs cost-sensitive classification D_1 as:

$$\pi_2 = \arg\min_{\pi} \sum_{k=1}^{|D_1|} Q^*(s_k, \pi(s_k)), \tag{29}$$

where s_k stands for the k'th data point collected at dataset D_1. Due to the construction of policy class Π, we see that π_2 must picks action a_l at state s_0 since $Q(s_0, a_l) < Q(s_0, a_r)$. Hence at the end of the episode $e = 1$, π_2 identifies s_1 (i.e., running π_2 from root s_0 leads to s_1), which is on the optimal trajectory.

Now assume that at the end of episode $n-1$, the newly updated policy π_n identifies the state $s_{(2^{n-1}-1)}$: namely at the beginning of episode n, if we roll-in π_n, the algorithm will keep traverse along the leftmost trajectory till at least state $s_{(2^{n-1}-1)}$. At episode n, let D_n as the dataset contains all data points from D_{n-1} and the new collected state, cost-to-go pairs from τ_n: $D_n = D_{n-1} \cup \{(s, [Q^*(s, a_l), Q^*(s, a_r)])\}$, for all $s \in \tau_n$. Now if we compute policy π_{n+1} using cost-sensitive classification (Eq. 29) over D_n, we must learn a policy π_{n+1} that identifies action a_l at state $s_{(2^j-1)}$, since $Q^e(s_{(2^j-1)}, a_l) < Q^*(s_{(2^j-1)}, a_r)$, and $s_{(2^j-1)}$ is included in D_n, for $j = 1, ..., n-1$. Hence at the end of episode n, we identify a policy π_{n+1} such that if we roll in policy π_{n+1} from s_0, we will traverse along the left most trajectory till we reach $s_{(2^n-1)}$.

Hence by the induction hypothesis, at the end of episode $K-1$, π_K will reach state $s_{(2^{K-1}-1)}$, the end of the best trajectory.

Since AggreVaTe with FTL with policy class Π identifies the best trajectory with at most $K-1$ episodes, the cumulative regret is then at most $O(K)$, which is $O(\log(S))$ (assuming the average cost at each leaf is a bounded constant), as S is the number of nodes in the binary-tree structure MDP $\tilde{\mathcal{M}}$. □

G. Proof of Theorem 5.3

Since in Theorem 5.3 we assume that we only have access to the noisy, but unbiased estimate of Q^*, the problem becomes more difficult since unlike in the proof of Theorem 5.2, we cannot simply eliminate states completely since the cost-to-go of the states queried from expert is noisy and completely eliminate nodes will potentially result elimination of low cost trajectories. Hence here we consider a different policy representation. We define 2^K deterministic base policies $\pi^1, ..., \pi^{2^K}$, such that rolling out policy π^i at state s_0 will traverse along the trajectory ending at the i'th leaf. We define the policy class Π as the convex hull of the base policies $\Pi = \{\pi : \sum_{i=1}^{2^K} w_i \pi^i, \sum_{i}^{2^K} w_i = 1, w_i \geq 0, \forall i\}$. Namely each $\pi \in \Pi$ is a stochastic policy: when rolling out, with probability w_i, π execute the i'th base policy π^i from s_0. Below we prove that AggreVaTeD with Weighted Majority achieves the regret bound $O(\sqrt{\ln(S)N})$.

Proof. We consider finite horizon, episodic imitation learning setting where at each episode n, the algorithm can roll in the current policy π_n once and only once and traverses through trajectory τ_n . Let us define $\tilde{\ell}_n(w) = \frac{1}{K+1}\sum_{s\in\tau_n}\sum_{j=1}^{2^K} w_j \tilde{Q}^e(s, \pi^j(s))$, where τ_n is the trajectory traversed by rolling out policy π_n starting at s_0, and $\tilde{Q}^e$ is a noisy but unbiased estimate of Q^*. We simply consider the setting where $\tilde{Q}^e$ is bounded $|\tilde{Q}^e| \leq l_{\max}$ (note that we can easily extend our analysis to a more general case where $\tilde{Q}^e$ is from a sub-Gaussian distribution). Note that $\tilde{\ell}_n(w)$ is simply a linear loss with respect to w:

$$\tilde{\ell}_n(w) = w \cdot q_n, \tag{30}$$

where $q_n[j] = \sum_{s\in\tau_n} \tilde{Q}^e(s, \pi^j(s))/(K+1)$. AggreVaTeD with WM updates w using Exponential gradient descent. Using the result from lemma D.2, we get:

$$\sum_{n=1}^{N}(\tilde{\ell}_n(w_n) - \tilde{\ell}_n(w^*)) = \sum_{n=1}^{N}(w_n \cdot q_n - w^* \cdot q_n) \leq \frac{\ln(2^K)}{\mu} + \frac{\mu}{2}\sum_{n=1}^{N}\sum_{j=1}^{2^K} w_n[j]q_n[j]^2 \leq \frac{\ln(2^K)}{\mu} + \frac{\mu}{2}\sum_{n=1}^{N} l_{\max}^2$$
$$= \frac{\ln(2^K)}{\mu} + \frac{\mu N l_{\max}^2}{2} \leq l_{\max}\sqrt{\ln(S)N}. \tag{31}$$

Note that $S = 2^{K+1} - 1$. The above inequality holds for any $w^* \in \Delta(2^K)$, including the w^e that corresponds to the expert (i.e., $w^e[1] = 1, w^e[i] = 0, i \neq 1$ as we assumed without loss of generality the left most trajectory is the optimal trajectory).

Now let us define $\ell_n(w)$ as follows:

$$\ell_n(w) = \frac{1}{K+1}\sum_{t=1}^{K+1}\sum_{s\sim\mathcal{S}} d_t^{\pi_n}(s)\sum_{j=1}^{2^K} w_j Q^*(s, \pi^j(s)). \tag{32}$$

Note $\ell_n(w)$ can be understood as first rolling out π_n *infinitely many times* and then querying for the exact cost-to-go Q^* on all the visited states. Clearly $\tilde{\ell}_n(w)$ is an unbiased estimate of $\ell_n(w)$: $\mathbb{E}[\tilde{\ell}_n(w)] - \ell_n(w) = 0$, where the expectation is over the randomness of the roll-in and sampling procedure of $\tilde{Q}^e$ at iteration n, conditioned on all events among the previous $n-1$ iterations. Also note that $|\tilde{\ell}_n(w) - \ell_n(w)| \leq 2l_{\max}$, since $\ell_n(w) \leq l_{\max}$. Hence $\{\tilde{\ell}_n(w_n) - \ell_n(w_n)\}$ is a bounded martingale difference sequence. Hence by Azuma-Heoffding inequality, we get with probability at least $1 - \delta/2$:

$$\sum_{n=1}^{N} \ell_n(w_n) - \tilde{\ell}_n(w_n) \leq 2l_{\max}\sqrt{\log(2/\delta)N}, \tag{33}$$

and with probability at least $1 - \delta/2$:

$$\sum_{n=1}^{N} \tilde{\ell}_n(w^e) - \ell_n(w^e) \leq 2l_{\max}\sqrt{\log(2/\delta)N}. \tag{34}$$

Combine the above inequality using union bound, we get with probability at least $1 - \delta$:

$$\sum_{n=1}^{N}(\ell_n(w_n) - \ell_n(w^e)) \leq \sum_{n=1}^{N}(\tilde{\ell}_n(w_n) - \tilde{\ell}_n(w^e)) + 4l_{\max}\sqrt{\log(2/\delta)N}. \tag{35}$$

Now let us apply the Performance Difference Lemma (Lemma D.1), we get with probability at least $1-\delta$:

$$\sum_{n=1}^{N}\mu(\pi_n)-\sum_{n=1}^{N}\mu(\pi^*)=\sum_{n=1}^{N}(K+1)\big(\ell_n(w_n)-\ell_n(w^e)\big)\leq(K+1)(l_{\max}\sqrt{\ln(S)N}+4l_{\max}\sqrt{\log(2/\delta)N}), \tag{36}$$

rearrange terms we get:

$$\sum_{n=1}^{N}\mu(\pi_n)-\sum_{n=1}^{N}\mu(\pi^*)\leq\log(S)l_{\max}(\sqrt{\ln(S)N}+\sqrt{\log(2/\delta)N})\leq O(\ln(S)\sqrt{\ln(S)N}), \tag{37}$$

with probability at least $1-\delta$. □

H. Proof of Theorem 5.4

The proof of theorem 5.4 is similar to the one for theorem 5.3. Hence we simply consider the infinitely many roll-ins and exact query of Q^* case. The finite number roll-in and noisy query of Q^* case can be handled by using the martingale difference sequence argument as shown in the proof of theorem 5.3.

Proof. Recall that in general setting, the policy π consists of probability vectors $\pi^{s,t}\in\Delta(A)$, for all $s\in\mathcal{S}$ and $t\in[H]$: $\pi=\{\pi^{s,t}\}_{\forall s\in\mathcal{S},t\in[H]}$. Also recall that the loss functions WM is optimizing are $\{\ell_n(\pi)\}$ where:

$$\ell_n(\pi)=\frac{1}{H}\sum_{t=1}^{H}\sum_{s\in\mathcal{S}}d_t^{\pi_n}(s)(\pi^{s,t}\cdot Q_t^*(s))=\sum_{t=1}^{H}\sum_{s\in\mathcal{S}}\pi^{s,t}\cdot q_n^{s,t} \tag{38}$$

where as we defined before $Q_t^*(s)$ stands for the cost-to-go vector $Q_t^*(s)[j]=Q_t^*(s,a_j)$, for the j'th action in $\mathcal{A}$, and $q_n^{s,t}=\frac{d_t^{\pi_n}(s)}{H}Q_t^*(s)$.

Now if we run Weighted Majority on ℓ_n to optimize $\pi^{s,t}$ for each pair of state and time step independently, we can get the following regret upper bound by using Lemma D.2:

$$\sum_{n=1}^{N}\ell_n(\pi)-\min_{\pi}\sum_{n=1}^{N}\ell_n(\pi_n)\leq\sum_{t=1}^{H}\sum_{s\in\mathcal{S}}\big(\frac{\ln(A)}{\mu}+\frac{\mu}{2}\sum_{n=1}^{N}\sum_{j=1}^{A}\pi^{s,t}[j]q_n^{s,t}[j]^2\big). \tag{39}$$

Note that we can upper bound $(q_n^{s,t}[j])^2$ as:

$$(q_n^{s,t}[j])^2\leq\frac{d_t^{\pi_n}(s)^2}{H^2}(Q^*_{\max})^2\leq\frac{d_t^{\pi_n}(s)}{H^2}(Q^2_{\max})^2 \tag{40}$$

Substitute it back, we get:

$$\begin{aligned}\sum_{n=1}^{N}(\ell_n(\pi_n)-\ell_n(\pi^*))&\leq\sum_{t=1}^{H}\sum_{s\in\mathcal{S}}\big(\frac{\ln(A)}{\mu}+\frac{\mu}{2}\sum_{n=1}^{N}\sum_{j=1}^{A}\pi^{s,t}[j]d_t^{\pi_n}(s)\frac{(Q^*_{\max})^2}{H^2}\big)\\&=\sum_{t=1}^{H}\big(\frac{S\ln(A)}{\mu}+\frac{\mu(Q^*_{\max})^2}{2H^2}\sum_{n=1}^{N}\sum_{s\in\mathcal{S}}d_t^{\pi_n}(s)\sum_{j=1}^{A}\pi^{s,t}[j]\big)=\sum_{t=1}^{H}\big(\frac{S\ln(A)}{\mu}+\frac{\mu(Q^*_{\max})^2}{2H^2}N\big)\\&\leq\frac{Q^*_{\max}}{H}\sqrt{2S\ln(A)N},\end{aligned} \tag{41}$$

if we set $\mu=\sqrt{(Q^*_{\max})^2NS\ln(A)/(2H^2)}$.

Now let us apply the performance difference lemma (Lemma D.1), we get:

$$R_N=\sum_{n=1}^{N}\mu(\pi_n)-\sum_{n=1}^{N}\mu(\pi^*)=H\sum_{n=1}^{N}(\ell_n(w_n)-\ell_n(w^e))\leq HQ^e_{\max}\sqrt{S\ln(A)N}. \tag{42}$$

□

I. Proof of Theorem 5.5

Let us use $\tilde{Q}^e(s)$ to represent the noisy but unbiased estimate of $Q^*(s)$.

Proof. For notation simplicity, we denote $\mathcal{S} = \{s_1, s_2, ..., s_S\}$. We consider a finite MDP with time horizon $H = 1$. The initial distribution $\rho_0 = \{1/S, ..., 1/S\}$ puts $1/S$ weight on each state. We consider the algorithm setting where at every episode n, a state $s^n \in \mathcal{S}$ is sampled from ρ_0 and the algorithm uses its current policy $\pi_n^{s_n} \in \Delta(A)$ to pick an action $a \in \mathcal{A}$ for s^n and then receives a noisy but unbiased estimate $\tilde{Q}^e(s^n)$ of $Q^*(s^n) \in \mathbb{R}^{|\mathcal{A}|}$. The algorithm then updates its policy from $\pi_n^{s^n}$ to $\pi_{n+1}^{s^n}$ for s^n while keep the other polices for other s unchanged (since the algorithm did not receive any feedback regarding $Q^*(s)$ for $s \neq s^n$ and the sample distribution ρ_0 is fixed and uniform). For expected regret $\mathbb{E}[R_N]$ we have the following fact:

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{s^n\sim\rho_0,\forall n}\Big[\mathop{\mathbb{E}}_{\tilde{Q}^e(s_n)\sim P_{s_n},\forall n}\big[\sum_{n=1}^{N}(\pi_n^{s^n}\cdot\tilde{Q}^e(s^n)-\pi^*_{s^n}\cdot\tilde{Q}^e(s^n))\big]\Big]\\
&=\mathop{\mathbb{E}}_{s^n\sim\rho_0,\forall n}\Big[\sum_{n=1}^{N}\mathop{\mathbb{E}}_{\tilde{Q}^e_i(s_i)\sim P_{s_i},i\leq n-1}\big[(\pi_n^{s^n}\cdot Q^*(s^n)-\pi^e_{s^n}\cdot Q^*(s^n))\big]\Big]\\
&=\sum_{n=1}^{N}\mathop{\mathbb{E}}_{s^i\sim\rho_0,i\leq n-1}\Big[\mathop{\mathbb{E}}_{\tilde{Q}^e_i(s_i)\sim P_{s_i},i\leq n-1}\big[\mathop{\mathbb{E}}_{s\sim\rho_0}(\pi_n^{s}\cdot Q^*(s)-\pi^*_{s}\cdot Q^*(s))\big]\Big]\\
&=\mathbb{E}\big[\sum_{n=1}^{N}\mathop{\mathbb{E}}_{s\sim\rho_0}\pi_n^s\cdot Q^*(s)-\mathop{\mathbb{E}}_{s\sim\rho_0}\pi^*_s\cdot Q^*(s)\big]\\
&=\mathbb{E}\sum_{n=1}^{N}[\mu(\pi_n)-\mu(\pi^*)],
\end{aligned}
\tag{43}
$$

where the expectation in the final equation is taken with respect to random variables $\pi_i, i \in [N]$ since each π_i is depend on $\tilde{Q}^e_j$, for $j < i$ and s^j, for $j < i$.

We first consider $\mathbb{E}_{\tilde{Q}^e(s^n)\sim P_{s^n},\forall n}\big[\sum_{n=1}^{N}(\pi_n^{s^n}\cdot\tilde{Q}^e(s^n)-\pi^*_{s^n}\cdot\tilde{Q}^e(s^n))\big]$ conditioned on a given sequence of $s^1, ..., s^N$. Let us define that among N episodes, the set of the index of the episodes that state s_i is sampled as $\mathcal{N}_i$ and its cardinality as N_i, and we then have $\sum_{i=1}^{S} N_i = N$ and $\mathcal{N}_i \cap \mathcal{N}_j = \emptyset$,for $i \neq j$.

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{\tilde{Q}^e(s^n)\sim P_{s^n},\forall n}\big[\sum_{n=1}^{N}(\pi_n^{s^n}\cdot\tilde{Q}^e(s^n)-\pi^*_{s^n}\cdot\tilde{Q}^e(s^n))\big]\\
&=\sum_{i=1}^{S}\sum_{j\in\mathcal{N}_i}\mathop{\mathbb{E}}_{\tilde{Q}^e_j(s_i)\sim P_{s_i}}(\pi_j^{s_i}\cdot\tilde{Q}^e_j(s_i)-\pi^e_{s_i}\tilde{Q}^e_j(s_i))
\end{aligned}
\tag{44}
$$

Note that for each state s_i, at the rounds from $\mathcal{N}_i$, we can think of the algorithm running any possible online linear regression algorithm to compute the sequence of policies $\pi_j^{s_i}, \forall j \in \mathcal{N}_i$ for state s_i. Note that from classic online linear regression analysis, we can show that for state s_i there exists a distribution P_{s_i} such that for any online algorithm:

$$
\mathop{\mathbb{E}}_{\tilde{Q}^e_j(s_i)\sim P_{s_i},\forall j\in\mathcal{N}_i}\big[\sum_{j\in\mathcal{N}_i}(\pi_j^{s_i}\cdot\tilde{Q}^e_j(s_i)-\pi^e_{s_i}\cdot\tilde{Q}^e_j(s_i))\big]\geq c\sqrt{\ln(A)N_i},
\tag{45}
$$

for some non-zero positive constant c. Substitute the above inequality into Eq. 44, we have:

$$
\mathop{\mathbb{E}}_{\tilde{Q}^e(s_n)\sim P_{s_n},\forall n}\big[\sum_{n=1}^{N}(\pi_n^{s^n}\cdot\tilde{Q}^e(s^n)-\pi^*_{s^n}\cdot\tilde{Q}^e(s^n))\big]\geq\sum_{i=1}^{S}c\sqrt{\ln(A)N_i}=c\sqrt{\ln(A)}\sum_{i=1}^{S}\sqrt{N_i}.
\tag{46}
$$

Now let us put the expectation $\mathbb{E}_{s^i\sim\rho_0,\forall i}$ back, we have:

$$
\mathop{\mathbb{E}}_{s^n\sim\rho_0,\forall n}\Big[\mathop{\mathbb{E}}_{\tilde{Q}^e(s_n)\sim P_{s_n}}\big[\sum_{n=1}^{N}(\pi_n^{s^n}\cdot\tilde{Q}^e(s^n)-\pi^*_{s^n}\cdot\tilde{Q}^e(s^n))|s^1,...,s^n\big]\Big]\geq c\sqrt{\ln(A)}\sum_{i=1}^{N}\mathbb{E}[\sqrt{N_i}].
\tag{47}
$$

Note that each N_i is sampled from a Binomial distribution $\mathcal{B}(N, 1/S)$. To lower bound $\mathbb{E}_{n\sim\mathcal{B}(N,1/S)}\sqrt{n}$, we use Hoeffding's Inequality here. Note that $N_i = \sum_{n=1}^{N} a_n$, where $a_n = 1$ if s_i is picked at iteration n and zero otherwise. Hence a_i is from a Bernoulli distribution with parameter $1/S$. Using Hoeffding bound, for N_i/N, we get:

$$P(|N_i/N - 1/S| <= \epsilon) \geq 1 - \exp(-2N\epsilon^2). \tag{48}$$

Let $\epsilon = 1/(2S)$, and substitute it back to the above inequality, we get:

$$P(0.5(N/S) \leq N_i \leq 1.5(N/S)) = P(\sqrt{0.5(N/S)} \leq \sqrt{N_i} \leq \sqrt{1.5(N/S)}) \geq 1 - \exp(-2N/S^2). \tag{49}$$

Hence, we can lower bound $\mathbb{E}[\sqrt{N_i}]$ as follows:

$$\mathbb{E}[\sqrt{N_i}] \geq \sqrt{0.5N/S}(1 - \exp(-2N/S^2)). \tag{50}$$

Take N to infinity, we get:

$$\lim_{N\to\infty} \mathbb{E}[\sqrt{N_i}] \geq \sqrt{0.5N/S}. \tag{51}$$

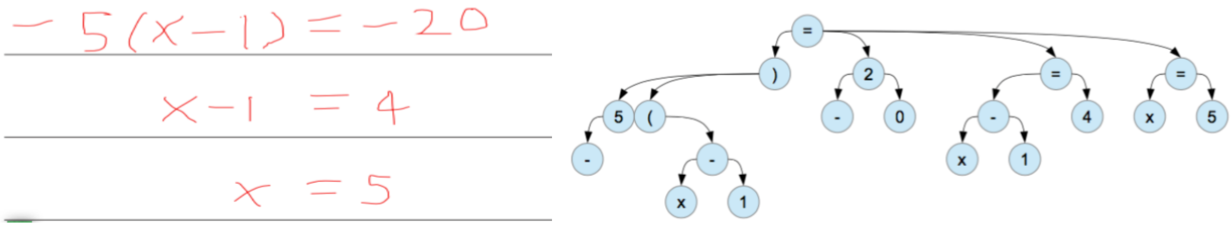
Substitute this result back to Eq. 47 and use the fact from Eq. 43, we get:

$$\begin{aligned}\lim_{N\to\infty} \mathbb{E}[R_N] = \lim_{N\to\infty} \mathbb{E}_{s^n\sim\rho_0,\forall n}\Big[\mathbb{E}_{\tilde{Q}^e(s_n)\sim P_{s_n},\forall n}\Big[\sum_{n=1}^{N}(\pi_n^{s^n}\cdot\tilde{Q}^e(s^n) - \pi^*_{s^n}\cdot\tilde{Q}^e(s^n))\Big]\Big] \geq c\sqrt{\ln(A)}\sum_{i=1}^{S}\mathbb{E}[\sqrt{N_i}]\\ \geq c\sqrt{\ln(A)}S\sqrt{0.5N/S} = \Omega(\sqrt{S\ln(A)N}).\end{aligned}$$

Hence we prove the theorem. □

J. Details of Dependency Parsing for Handwritten Algebra

In Fig. 4, we show an example of set of handwritten algebra equations and its dependency tree from a arc-hybird sequence $slssslssrrllslsslssrrslssrlssrrslssrr$. The preprocess step cropped individual symbols one by one from left to right and from the top equation to the bottom one, centered them, scaled symbols to 40 by 40 images, and finally formed them as a sequence of images.



(a) Handwritten algebra equations

(b) Dependency tree

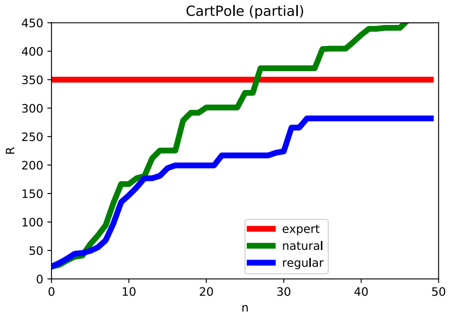
Figure 4. An example of a set of handwritten algebra equations (a) and its corresponding dependency tree (b).

Since in the most common dependency parsing setting, there is no immediate reward at every parsing step, the reward-to-go $Q^*(s,a)$ is computed by using UAS as follows: start from s and apply action a, then use expert π^* to roll out til the end of the parsing process; $Q^*(s,a)$ is the UAS score of the final configuration. Hence AggreVaTeD can be considered as directly maximizing the UAS score, while previous approaches such as DAgger or SMILe (Ross et al., 2011) tries to mimic expert's actions and hence are not directly optimizing the final objective.

K. Additional Experiments on Partial Observable Setting

We test AggreVaTeD with Gated Recurrent Unit (GRU) based policies on a partially observable CartPole environment. Again the expert has access to the full state while the observation excludes the velocity information of the cart.

Fig. 5 shows that even under partial observable setting, AggreVaTeD with RNN-based policies can also outperform suboptimal experts.



(a)

Figure 5. AggreVaTeD with GRU on the partial observable CartPole setting.