A Boosting Approach to Reinforcement Learning

Nataly Brukhim [1] Elad Hazan [1] Karan Singh [2]

Abstract

We study efficient algorithms for reinforcement learning in Markov decision processes, whose complexity is independent of the number of states. This formulation succinctly captures large scale problems, but is also known to be computationally hard in its general form. We consider the methodology of boosting, borrowed from supervised learning, for converting weak learners into an accurate policy. The notion of weak learning we study is that of sampled-based approximate optimization of linear functions over policies. We give an efficient algorithm that is capable of improving the accuracy of such weak learning methods, till global optimality is reached. We prove sample complexity and running time bounds on our method, that are polynomial in the natural parameters of the problem: approximation guarantee, discount factor, distribution mismatch and number of actions. In particular, our bound does not explicitly depend on the number of states.

A technical difficulty in applying previous boosting results, is that the value function over policy space is not convex. We show how to use a non-convex variant of the Frank-Wolfe method, coupled with recent advances in gradient boosting that allow incorporating a weak learner with multiplicative approximation guarantee, to overcome the non-convexity and attain global convergence.

1. Introduction

The field of reinforcement learning, formally modelled as learning in Markov decision processes (MDP), models the mechanism of learning from rewards, as opposed to examples. Various techniques have been suggested and applied to cope with very large MDPs. The most common of these is function approximation of either the value or the transition function of the underlying MDP, many times using deep neural networks. Training deep neural networks in the supervised learning model is known to be computationally hard. Therefore reinforcement learning with neural function approximation is also computationally hard in general, and for this reason lacks provable guarantees. Previous approaches can be categorized in terms of the structural assumptions made on the MDP to circumvent the computational hardness. Some studies focus on structured dynamics, whereas others on structured value functions.

In this paper we study another methodology to derive provable algorithms for reinforcement learning: ensemble methods for aggregating weak or approximate algorithms into substantially more accurate solutions. Our method can be thought of as extending the methodology of boosting from supervised learning (Schapire & Freund, 2012) to reinforcement learning. In order to circumvent the computational hardness of solving general MDPs with function approximation, we assume access to a weak learner: an efficient sample-based procedure that is capable of generating an approximate solution to any linear optimization objective over the space of policies. We describe an algorithm that iteratively calls this procedure on carefully constructed new objectives, and aggregates the solution into a single policy. We prove that after sufficiently many iterations, our resulting policy is provably near-optimal.

1.1. Challenges and techniques

Extending the boosting methodology to reinforcement learning settings presents a number of challenges.

(A) The value function is not a convex or concave function of the policy. This is true in the tabular case, and even more so if we use a parameterized policy class.

(B) The transition matrix is unknown, or prohibitively large to manipulate for large state spaces.

(C) It is unrealistic to expect a weak learner that attains near-optimal value for a given linear objective over the policy class. At most one can hope for a multiplicative and/or additive approximation of the overall value.

Our approach overcomes these challenges by applying classical, as well as more recently developed techniques. To overcome the nonconvexity of the value function, we use a novel variant of the Frank-Wolfe optimization algorithm that

[1]Princeton University [2]Microsoft Research. Correspondence to: Karan Singh <iamkaransingh@gmail.com>, Nataly Brukhim <nbrukhim@princeton.edu>.

Reinforcement Learning Theory Workshop at ICML 2021.

	Supervised weak learner	Online weak learner	
Episodic model	$C_\infty^6(\Pi)/\alpha^4\varepsilon^5$	$C_\infty^4(\Pi)/\alpha^2\varepsilon^3$	$C_\infty = \max_{\pi\in\Pi} \left\| \frac{d^{\pi^*}}{d^\pi} \right\|_\infty$
Rollouts w. ν-resets	$\mathcal{D}_\infty^6/\alpha^4\varepsilon^6$	$\mathcal{D}_\infty^4/\alpha^2\varepsilon^4$	$D_\infty = \left\| \frac{d^{\pi^*}}{\nu} \right\|_\infty$

Table 1. Sample complexity of the proposed algorithms for different α-weak learning models: supervised & online (appendix), and modes of accessing the MDP (rollouts & rollouts with reset distribution ν), suppressing polynomial factors in $|A|, 1/(1-\gamma)$.

simultaneously delivers on two guarantees. First, it finds a first order stationary point with near-optimal rate. Secondly, if the objective happens to admit a certain gradient domination property, an important generalization of convexity, it also guarantees near optimal value. The application of the nonconvex Frank-Wolfe method is justified due to previous recent investigation of the policy gradient algorithm (Agarwal et al., 2019; 2020a), which identified conditions under which the value function is gradient dominated.

The second challenge of the unknown transition function is overcome by careful algorithmic design: our boosting algorithm requires only samples of the transitions and rewards. These are obtained by rollouts on the MDP.

The third challenge is perhaps the most difficult to overcome. We use techniques from (Hazan & Singh, 2021), which studies boosting with a multiplicative weak learner in the online convex optimization setup. We make critical use of this new technique by performing non-linear aggregation (using a 2-layer neural network) of the weak learners. This aspect is perhaps of general interest to boosting algorithm design, which is mostly based on linear aggregation.

1.2. Our contributions

Our main contribution is a novel efficient boosting algorithm for reinforcement learning. Given a weak learning method capable of approximately optimizing a linear function over a certain policy class, the output of the algorithm is a policy which does not belong to the original class considered. It is rather a non-linear aggregation of policies from the original class, according to a two-layer network. This is a result of the two-tier structure of our algorithm: an outer loop of non-convex Frank-Wolfe method, and an inner loop of online convex optimization boosting. The final policy comes with provable guarantees against the class of all policies. Our algorithm and guarantees come in four flavors, depending on the mode of MDP access, and the weak learner.

1.3. Related work

To cope with prohibitively large MDPs, policy-based methods that utilize a compactly parameterized policy class are popular resorts. Recently, (Agarwal et al., 2019; 2020a; Bhandari & Russo, 2019) established global convergence of policy gradient methods. Other policy iteration variants which make incremental updates include: Conservative Policy Iteration (CPI) (Kakade & Langford, 2002; Scherrer & Geist, 2014), and Policy Search by Dynamic Programming (PSDP)(Bagnell et al., 2003).

Our boosting approach builds on the vast literature of boosting for supervised learning (Schapire & Freund, 2012), and recently online learning (Leistner et al., 2009; Chen et al., 2012; 2014; Beygelzimer et al., 2015; Jung et al., 2017; Jung & Tewari, 2018). A closely related work is boosting in the context of control of dynamical systems (Agarwal et al., 2020b). However, this work critically requires knowledge of the underlying dynamics (transitions), which we do not, and cannot cope with a multiplicative approximate weak learner.

One of the crucial techniques is the extension of boosting to the online convex optimization setting, with bandit information (Brukhim & Hazan, 2021), and critically with a multiplicative weak learner (Hazan & Singh, 2021).

2. Preliminaries

Optimization. We say that a differentiable function $f : \mathcal{K} \mapsto \mathbb{R}$ over some domain $\mathcal{K}$ is L-smooth with respect to some norm $\|\cdot\|_*$ if for every $x, y \in \mathcal{K}$ we have $\left|f(y) - f(x) - \nabla f(x)^\top (y-x)\right| \leq \frac{L}{2}\|x-y\|_*^2$. For constrained optimization (such as over Δ_A), the projection $\Gamma : \mathbb{R}^{|A|} \to \Delta_A$ of a point x to onto a domain Δ_A is $\Gamma[x] = \arg\min_{y\in\Delta_A} \|x-y\|$.

Markov decision process. An infinite-horizon discounted Markov Decision Process (MDP) $M = (S, A, P, r, \gamma, d_0)$ is specified by: a state space S, an action space A, a transition model P where $P(s'|s,a)$ denotes the probability of immediately transitioning to state s' upon taking action a at state s, a reward function $r : S \times A \to [0,1]$ where $r(s,a)$ is the immediate reward associated with taking action a at state s, a discount factor $\gamma \in [0,1)$; a starting state distribution d_0 over S. For any infinite-length state-action sequence (hereafter, called a trajectory), we assign the value $V(\tau =$

$(s_0, a_0, s_1, a_1, \dots)) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$. The agent interacts with the MDP through the choice of stochastic policy $\pi : S \to \Delta_A$ it executes, where Δ_A denotes the probability simplex over A. The execution of such a policy induces a distribution over trajectories $\tau = (s_0, a_0, \dots)$ as $P(\tau|\pi) = d_0(s_0) \prod_{t=0}^{\infty} (P(s_{t+1}|s_t, a_t)\pi(a_t|s_t))$.. Using this description we can associate a state $V^\pi(s)$ and state-action $Q^\pi(s, a)$ value function with any policy π. For an arbitrary distrbution d over S, define: $Q^\pi(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)|\ \pi, s_0 = s, a_0 = a]$, $V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[Q^\pi(s,a)|\pi, s\right], V_d^\pi = \mathbb{E}_{s_0\sim d}\left[V^\pi(s)|\pi\right]$. Here the expectation is with respect to the randomness of the trajectory induced by π in M. When convenient, we shall use V^π to denote $V_{d_0}^\pi$, and V^* to denote $\max_\pi V^\pi$. Similarly, to any policy π, one may ascribe a (discounted) state-visitation distribution $d^\pi = d_{d_0}^\pi$: $d_d^\pi(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \sum_{\tau: s_t = s} P(\tau|\pi, s_0 \sim d)$

Modes of Accessing the MDP. We consider two modes of accessing the MDP, that are standard in the reinforcement learning literature, and provide different results for each.

The first natural access model is the **episodic rollout setting.** This mode of interaction allows us to execute a policy, stop and restart at any point, and do this multiple times.

Another interaction model we consider is **rollout with ν-restarts.** This is similar to the episodic setting, but here the agent may draw from the MDP a trajectory seeded with an initial state distribution $\nu \neq d_0$. This interaction model was considered in prior work on policy optimization (Kakade & Langford, 2002; Agarwal et al., 2019). The motivation for this model is two-fold: first, ν can be used to incorporate priors (or domain knowledge) about the state coverage of the optimal policy; second, ν provides a mechanism to incorporate exploration into policy optimization procedures.

3. Setting

3.1. Policy aggregation

For a base class of policies Π_W, our algorithm incrementally builds a more expressive policy class by aggregating base policies via both linear combinations and non-linear transformations. In effect, the algorithm produces a finite-width depth-2 circuit over some subset of the base policy class. We start with the simpler linear aggregation.

Definition 1 (Function Aggregation)**.** Given some $N_0 \in \mathbb{Z}_+, w \in \mathbb{R}^{N_0}, (f_1, \dots f_{N_0}) \in (S \to \mathbb{R}^{|A|})^{\otimes N_0}$, we define $f = \sum_{n=1}^{N_0} w_n f_n$ to be such that simultaneously for all $s \in S$, it holds $f(s) = \sum_{n=1}^{N_0} w_n f(s)$.

The projection below may be viewed as a non-linear activation, such as ReLU, in deep learning terms. Note that the projection of any function from S to $\mathbb{R}^{|A|}$ produces a policy, i.e. a mapping states to distributions over actions.

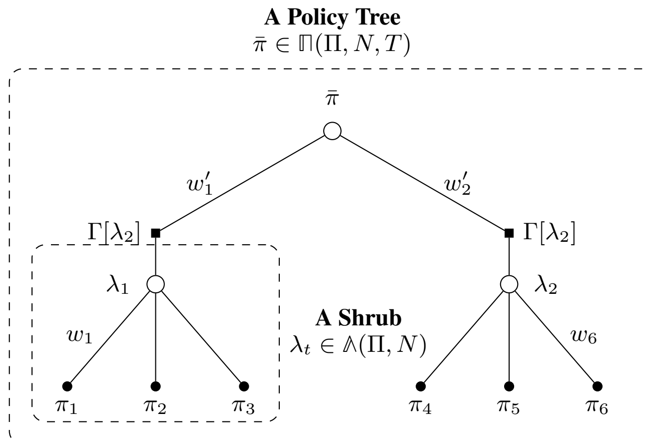

Figure 1. A typical Policy Tree (see Definition 4): $N = 3, T = 2$.

Definition 2 (Policy Projection)**.** Given a function $f : S \to \mathbb{R}^{|A|}$, define a projected policy $\pi = \Gamma[f]$ to be a policy such that for all $s \in S$, it holds that $\pi(\cdot|s) = \Gamma\left[f(s)\right]$.

The next definition defines the class of functions represented by circuits of depth 1 over a base policy class. Note that these function do not necessarily represent policies since they take an affine (vs. convex) combination of policies.

Definition 3 (Shrub)**.** For an arbitrary base policy class $\Pi \subseteq S \to \Delta_A$, define $\mathbb{\Lambda}(\Pi, N)$ to be a set such that $\lambda \in \Lambda(\Pi, N)$ if and only if there exists $N_0 \le N, w \in \mathbb{R}^{N_0}, (\pi_1, \dots \pi_{N_0}) \in \Pi^{\otimes N_0}$ such that $\lambda = \sum_{n=1}^{N_0} w_n \pi_n$.

The final definition describes the set of possible outputs of the boosting procedure.

Definition 4 (Policy Tree)**.** For an arbitrary base policy class $\Pi \subseteq S \to \Delta_A$, define $\mathbb{\Pi}(\Pi, N, T)$ to be a policy class such that $\pi \in \mathbb{\Pi}(\Pi, N, T)$ if and only if there exists $T_0 \le T, w \in \Delta_{T_0}, (\lambda_1, \dots \lambda_{T_0}) \in \mathbb{\Lambda}(\Pi, N)^{\otimes T_0}$ such that $\pi = \sum_{t=1}^{T_0} w_t \Gamma[\lambda_t]$.

Claim 5. *For any $\pi \in \mathbb{\Pi}(\Pi, N, T)$, $\pi(\cdot|s)$ for any $s \in S$ can be evaluated using TN base policy evaluations and $O(T \times (NA + A\log A))$ arithmetic and logical operations.*

3.2. Models of weak learning

The natural way to define weak learning is an algorithm whose performance is slightly better than that of random policy π_r, one that chooses an action uniformly at random at any given state. However, in general no learner can outperform a random learner over all label distributions (the "no free lunch" theorem). This motivates the literature on agnostic boosting (Kanade & Kalai, 2009; Brukhim et al., 2020; Hazan & Singh, 2021) that defines a weak learner as one that can approximate the best policy in a given class.

Definition 6 (Weak Supervised Learner)**.** Let $\alpha \in (0, 1)$. Consider a class $\mathcal{L}$ of linear loss functions $\ell : \mathbb{R}^A \to \mathbb{R}$, and $\mathbb{D}$ a family of distributions that are supported over $S \times \mathcal{L}$, policy classes Π_W, Π. A weak supervised learning algorithm, for every $\varepsilon, \delta > 0$, given $m(\varepsilon, \delta)$ samples D_m from any distribution $\mathcal{D} \in \mathbb{D}$ outputs a policy $\mathcal{W}(D_m) \in \Pi_W$ such that with probability $1 - \delta$,

$$\mathbb{E}_{(s,\ell)\sim\mathcal{D}}\big[\ell(\mathcal{W}(D_m))\big] \;\geq\; \alpha \max_{\pi^* \in \Pi} \mathbb{E}_{(s,\ell)\sim\mathcal{D}}\big[\ell(\pi^*(s))\big] + (1-\alpha)\mathbb{E}_{(s,\ell)\sim\mathcal{D}}\big[\ell(\pi_r(s))\big] - \varepsilon.$$

The weak learner outputs a policy in Π_W which is approximately competitive against the class Π. As an additional relaxation, it will be sufficient that the weak learning assumption holds over *natural* distributions – distributions whose marginal over states is realizable as the state-distribution of some policy $\pi \in \mathbb{\Pi}$. Therefore, the complexity of weak learning adapts to the complexity of the MDP itself.

Hereafter, we refer to $\mathbb{\Pi}(\Pi_W, N, T)$ as $\mathbb{\Pi}$ for $N, T = O(\text{poly}(|A|, (1-\gamma)^{-1}, \varepsilon^{-1}, \alpha^{-1}, \log \delta^{-1}))$ specified later.

4. Algorithm & Main Results

Algorithm 1 RL Boosting via Weak Supervised Learning

Initialize a policy $\pi_0 \in \Pi_W$ arbitrarily.
for $t = 1$ **to** T **do**
  Set $\rho_{t,0}$ to be an arbitrary policy in Π_W.
  for $n = 1$ **to** N **do**
    Execute π_{t-1} M-times with initial distribution μ via Algorithm 2, to get $D_{t,n} = \{(s_i, \widehat{Q}_i)\}_{i=1}^m$.
    Produce a dataset $D'_{t,n} = \{(s_i, f_i)\}_{i=1}^m$:

$$f_i = -\nabla F_{G,\beta}[-\widehat{Q}_i](\rho_{t,n}(\cdot|s_i))$$

    Let $\mathcal{A}_{t,n}$ be the policy chosen by the weak learning oracle when given data set $D'_{t,n}$.
    Update $\rho_{t,n} = (1 - \eta_{2,n})\rho_{t,n-1} + \frac{\eta_{2,n}}{\alpha}\mathcal{A}_{t,n} - \eta_{2,n}\left(\frac{1}{\alpha} - 1\right)\pi_r$.
  end for
  Declare $\pi'_t = \Gamma\left[\rho_{t,N}\right]$.
  Choose $\eta_{1,t} = \min\{1, \frac{2C_\infty(\mathbb{\Pi})}{t}\}$ if $\mu = d_0$ else $\eta_{1,t} = \texttt{StepChooser}(\pi_{t-1}, \pi'_t, \mu, P)$.
  Update $\pi_t = (1 - \eta_{1,t})\pi_{t-1} + \eta_{1,t}\pi'_t$.
end for
Output $\bar{\pi}$=π_T if $\mu = d_0$ else π_{t-1} with the smallest η_t.

Smoothing & Extension. Define the inf-convolution smoothing operator as follows (Beck, 2017): $M_\beta[f](x) = \min_{y\in\Delta_A}\left\{f(y) + \frac{1}{2\beta}\|x - y\|^2\right\}$ The extension operator (Hazan & Singh, 2021) operate overs functions and modifies their value outside and near the boundary of the convex set Δ_A to aid the booster. $F_{G,\beta}[f](x) = M_\beta\left[f(x) + G\min_{y\in\Delta_A}\|y - x\|\right]$

We need the following definitions. The first generalizes the policy completeness notion from (Scherrer & Geist, 2014). It may be seen as the policy-equivalent analogue of inherent bellman error (Munos & Szepesvári, 2008). Intuitively, it measures the degree to which a policy in Π can best approximate the bellman operator in an average sense with respect to the state distribution induced by a policy from $\mathbb{\Pi}$.

Definition 7 (Policy Completeness)**.** For any initial state distribution μ, define $\mathcal{E}_\mu(\mathbb{\Pi}, \Pi) = \max_{\pi\in\mathbb{\Pi}}\min_{\pi^*\in\Pi}$ $\mathbb{E}_{s\sim d^\pi_\mu}\left[\max_{a\in A} Q^\pi(s, a) - Q^\pi(s, \cdot)^\top \pi^*(\cdot|s)\right]$.

The following notion of the distribution mismatch coefficient is often useful to characterize the exploration problem faced by policy optimization algorithms.

Definition 8 (Distribution Mismatch)**.** Let $\pi^* = \arg\max_\pi V^\pi$, and ν a fixed initial state distribution (see section 2). Define the following distribution mismatch coefficients: $C_\infty(\mathbb{\Pi}) = \max_{\pi\in\mathbb{\Pi}}\left\|\frac{d^{\pi^*}}{d^\pi}\right\|_\infty$, $D_\infty = \left\|\frac{d^{\pi^*}}{\nu}\right\|_\infty$.

Theorem 9. *Algorithm 1 samples $T(MN + P)$ episodes of length $\frac{1}{1-\gamma}\log\frac{T(MN+P)}{\delta}$ with probability $1 - \delta$. In the episodic model, Algorithm 1 guarantees as long as $T = \frac{16C_\infty^2(\mathbb{\Pi})}{(1-\gamma)^3\varepsilon}$, $N = \left(\frac{16|A|C_\infty(\mathbb{\Pi})}{(1-\gamma)^2\alpha\epsilon}\right)^2$, $M = m\left(\frac{(1-\gamma)^2\alpha\varepsilon}{8C_\infty(\mathbb{\Pi})|A|}, \frac{\delta}{NT}\right)$, $\mu = d_0$, we have with probability $1-\delta$*

$$V^* - V^\pi \;\leq\; C_\infty(\mathbb{\Pi})\frac{\mathcal{E}(\mathbb{\Pi}, \Pi)}{1-\gamma} + \varepsilon$$

In the ν-reset model, as long as $T = \frac{8D_\infty^2}{(1-\gamma)^6\varepsilon^2}$, $N = \left(\frac{16|A|D_\infty}{(1-\gamma)^3\alpha\epsilon}\right)^2$, $P = \frac{200|A|^2D_\infty^2}{(1-\gamma)^6\varepsilon^2}\log\frac{2TN}{\delta}$, $M = m\left(\frac{(1-\gamma)^3\alpha\varepsilon}{8|A|D_\infty}, \frac{\delta}{2NT}\right)$, $\mu = \nu$, we have with probability $1 - \delta$

$$V^* - V^\pi \;\leq\; D_\infty\frac{\mathcal{E}_\nu(\mathbb{\Pi}, \Pi)}{(1-\gamma)^2} + \varepsilon$$

If $m(\varepsilon, \delta) = \frac{\log|\mathcal{W}|}{\varepsilon^2}\log\frac{1}{\delta}$ for some measure of weak learning complexity $|\mathcal{W}|$, the algorithm samples $\tilde{O}\left(\frac{C_\infty^6(\mathbb{\Pi})|A|^4\log|\mathcal{W}|}{(1-\gamma)^{11}\alpha^4\varepsilon^5}\right)$ episodes in the episodic model, and $\tilde{O}\left(\frac{D_\infty^6|A|^4\log|\mathcal{W}|}{(1-\gamma)^{18}\alpha^4\varepsilon^6}\right)$ in the ν-reset model.

5. Conclusions

Building on recent advances in boosting for online convex optimization and bandits, we have described a boosting algorithm for reinforcement learning over large state spaces with provable guarantees. We see this as a first attempt at using a tried-and-tested methodology from supervised learning in RL, and many challenges remain.

References

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *arXiv preprint arXiv:1908.00261*, 2019.

Agarwal, A., Henaff, M., Kakade, S., and Sun, W. Pc-pg: Policy cover directed exploration for provable policy gradient learning. *arXiv preprint arXiv:2007.08459*, 2020a.

Agarwal, N., Brukhim, N., Hazan, E., and Lu, Z. Boosting for control of dynamical systems. In *International Conference on Machine Learning*, pp. 96–103. PMLR, 2020b.

Bagnell, J. A., Kakade, S., Ng, A. Y., and Schneider, J. G. Policy search by dynamic programming. In *Advances in Neural Information Processing Systems*, 2003.

Beck, A. *First-order methods in optimization*. SIAM, 2017.

Beygelzimer, A., Langford, J., Li, L., Reyzin, L., and Schapire, R. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 19–26. JMLR Workshop and Conference Proceedings, 2011.

Beygelzimer, A., Kale, S., and Luo, H. Optimal and adaptive algorithms for online boosting. In *International Conference on Machine Learning*, pp. 2323–2331, 2015.

Bhandari, J. and Russo, D. Global optimality guarantees for policy gradient methods. *arXiv preprint arXiv:1906.01786*, 2019.

Brukhim, N. and Hazan, E. Online boosting with bandit feedback. In *Algorithmic Learning Theory*, pp. 397–420. PMLR, 2021.

Brukhim, N., Chen, X., Hazan, E., and Moran, S. Online agnostic boosting via regret minimization. In *Advances in Neural Information Processing Systems*, 2020.

Chen, S.-T., Lin, H.-T., and Lu, C.-J. An online boosting algorithm with theoretical justifications. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1873–1880, 2012.

Chen, S.-T., Lin, H.-T., and Lu, C.-J. Boosting with online binary learners for the multiclass bandit problem. In *International Conference on Machine Learning*, pp. 342–350, 2014.

Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pp. 272–279, 2008.

Hazan, E. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

Hazan, E. and Singh, K. Boosting for online convex optimization. *arXiv preprint arXiv:2102.09305*, 2021.

Jung, Y. H. and Tewari, A. Online boosting algorithms for multi-label ranking. In *International Conference on Artificial Intelligence and Statistics*, pp. 279–287, 2018.

Jung, Y. H., Goetz, J., and Tewari, A. Online multiclass boosting. In *Advances in neural information processing systems*, pp. 919–928, 2017.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *In Proc. 19th International Conference on Machine Learning*. Citeseer, 2002.

Kanade, V. and Kalai, A. Potential-based agnostic boosting. In *Advances in neural information processing systems*, pp. 880–888, 2009.

Lazaric, A. and Munos, R. Hybrid stochastic-adversarial on-line learning. In *Conference on Learning Theory*, 2009.

Leistner, C., Saffari, A., Roth, P. M., and Bischof, H. On robustness of on-line boosting-a competitive study. In *IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pp. 1362–1369. IEEE, 2009.

Munos, R. and Szepesvári, C. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9 (5), 2008.

Rakhlin, A., Sridharan, K., and Tewari, A. Online learning: Stochastic and constrained adversaries. *arXiv preprint arXiv:1104.5070*, 2011.

Schapire, R. E. and Freund, Y. *Boosting: Foundations and Algorithms*. MIT Press, 2012.

Scherrer, B. and Geist, M. Local policy search in a convex space and conservative policy iteration as boosted policy search. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 35–50. Springer, 2014.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In Solla, S., Leen, T., and Müller, K. (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000. URL https://proceedings.neurips.cc/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

A. Sampling & Step-size subroutine

Algorithm 2 Trajectory Sampler: $s \sim d^\pi$, unbiased estimate of Q^π_s

1: Sample state $s_0 \sim \mu$, and action $a' \sim \mathcal{U}(A)$ uniformly.
2: Sample $s \sim d^\pi$ as follows: at every timestep h, with probability γ, act according to π; else, accept s_h as the sample and proceed to Step 3.
3: Take action a' at state s_h, then continue to execute π, and use a termination probability of $1-\gamma$. Upon termination, set $R(s_h, a')$ as the *undiscounted* sum of rewards from time h onwards.
4: Define the vector $\widehat{Q^\pi_{s_h}}$, such that for all $a \in A$, $\widehat{Q^\pi_{s_h}}(a) = |A| \cdot R(s_h, a') \cdot \mathbb{I}_{a=a'}$.
5: Return $(s_h, \widehat{Q^\pi_{s_h}})$.

Below we give an algorithm for choosing step sizes used in both of the RL boosting methods (for online, and supervised, weak learners).

Algorithm 3 $\texttt{StepChooser}(\pi_{t-1}, \pi'_t, \mu, P)$

Execute π_{t-1} for P episodes with initial state distribution μ via Algorithm 2, to get

$$D = \{(s_i, \widehat{Q}_i)_{i=1}^P\}.$$

For any policy π, let $\widehat{G^\pi} = \frac{1}{P}\sum_{p=1}^P \widehat{Q}_i^\top \pi(\cdot|s_i)$.

Return

$$\eta_{1,t} = \texttt{clip}_{[0,1]}\left(\frac{(1-\gamma)^2}{2}\left(\widehat{G^{\pi'_t}} - \widehat{G^{\pi_{t-1}}}\right)\right)$$

B. RL Boosting via Weak Online Learning

Algorithm 4 RL Boosting via Weak Online Learning

Initialize a policy $\pi_0 \in \Pi_W$ arbitrarily.
for $t = 1$ **to** T **do**
  Initialize online weak learners $\mathcal{W}_1, \dots \mathcal{W}^N$.
  for $m = 1$ **to** M **do**
    Execute π_{t-1} once with initial state distribution μ via Algorithm 2, to get $(s_{t,m}, \widehat{Q}_{t,m})$.
    Choose $\rho_{t,m,0} \in \Pi_W$ arbitrarily.
    for $n = 1$ **to** N **do**
      Set $\rho_{t,m,n} = (1-\eta_{2,n})\rho_{t,m,n-1} + \frac{\eta_{2,n}}{\alpha}\mathcal{W}^n - \eta_{2,n}\left(\frac{1}{\alpha} - 1\right)\pi_r$.
    end for
    Pass to each $\mathcal{W}^n$ the following loss linear $f_{t,m,n}$:

$$f_{t,m,n} = -\nabla F_{G,\beta}[-\widehat{Q}_{t,m}](\rho_{t,m,n}(\cdot|s_i))$$

  end for
  Declare $\pi'_t = \frac{1}{M}\sum_{m=1}^M \Gamma\left[\rho_{t,m,N}\right]$.
  Choose $\eta_{1,t} = \min\{1, \frac{2C_\infty(\Pi)}{t}\}$ if $\mu = d_0$ else set $\eta_{1,t} = \texttt{StepChooser}(\pi_{t-1}, \pi'_t, \mu, P)$.
  Update $\pi_t = (1-\eta_{1,t})\pi_{t-1} + \eta_{1,t}\pi'_t$.
end for
Output $\bar{\pi}$= π_T if $\mu = d_0$ else output π_{t-1} with the smallest η_t.

Weak Online Learning. The second model of weak learning we consider requires a stronger assumption, but will give us better sample and oracle complexity bounds henceforth.

Definition 10 (Weak Online Learner)**.** Let $\alpha \in (0,1)$. Consider a class $\mathcal{L}$ of linear loss functions $\ell : \mathbb{R}^A \to \mathbb{R}$. A weak online learning algorithm, for every $M > 0$, incrementally for each timestep computes a policy $\mathcal{W}_m \in \Pi_W$ and then observes the state-loss pair $(s, \ell_t) \in S \times \mathcal{L}$ such that

$$\sum_{m=1}^{M} \ell_m(\mathcal{W}_m(s_m)) \;\geq\; \alpha \max_{\pi^* \in \Pi} \sum_{m=1}^{M} \ell_m(\pi^*(s_m)) + (1-\alpha) \sum_{m=1}^{M} \ell_m(\pi_r(s_m)) - R_{\mathcal{W}}(M).$$

Assumption 1 (Weak Online Learning)**.** *The booster has access to a weak online learning oracle (Definition 10) over the policy class* Π*, for some* $\alpha \in (0,1)$*.*

Remark 11. A similar remark about *natural* distributions applies to the online weak learner. In particular, it is sufficient the guarantee in 10 holds for arbitrary sequence of loss functions with high probability over the sampling of the state from d^π for some $\pi \in \mathbb{\Pi}$. Although stronger than supervised weak learning, this oracle can be interpreted as a relaxation of the online weak learning oracle considered in (Brukhim et al., 2020; Brukhim & Hazan, 2021; Hazan & Singh, 2021). A similar model of hybrid adversarial-stochastic online learning was considered in (Rakhlin et al., 2011; Lazaric & Munos, 2009; Beygelzimer et al., 2011). In particular, it is known (Lazaric & Munos, 2009) that unlike online learning, the capacity of a hypothesis class for this model is governed by its VC dimension (vs. Littlestone dimension).

Theorem 12. *Algorithm 4 samples* $T(M+P)$ *episodes of length* $\frac{1}{1-\gamma}\log\frac{T(M+P)}{\delta}$ *with probability* $1-\delta$*. In the episodic model, Algorithm 4 guarantees as long as* $T = \frac{16C_\infty^2(\mathbb{\Pi})}{(1-\gamma)^3\varepsilon}$, $N = \left(\frac{16|A|C_\infty(\mathbb{\Pi})}{(1-\gamma)^2\alpha\epsilon}\right)^2$, $M = \max\left\{\frac{1000|A|^2C_\infty^2(\mathbb{\Pi})}{(1-\gamma)^4\varepsilon^2\alpha^2}\log^2 T\delta, \frac{8|A|C_\infty(\mathbb{\Pi})R_{\mathcal{W}}(M)}{(1-\gamma)^2\alpha\varepsilon}\right\}$*,*$\mu = d_0$*, we have with probability* $1-\delta$

$$V^* - V^\pi \;\leq\; C_\infty(\mathbb{\Pi})\frac{\mathcal{E}(\mathbb{\Pi},\Pi)}{1-\gamma} + \varepsilon$$

In the ν*-reset model, Algorithm 1 guarantees as long as* $T = \frac{100D_\infty^2}{(1-\gamma)^6\varepsilon^2}$, $N = \left(\frac{20|A|D_\infty}{(1-\gamma)^3\alpha\epsilon}\right)^2$, $P = \frac{250D_\infty^2|A|^2}{(1-\gamma)^6\varepsilon^2}\log^2\frac{T}{\delta}$, $M = \max\left\{\left(\frac{40|A|D_\infty}{(1-\gamma)^3\alpha\varepsilon}\log\frac{T}{\delta}\right)^2, \frac{10|A|D_\infty R_{\mathcal{W}}(M)}{(1-\gamma)^3\alpha\varepsilon}\right\}$*,*$\mu = \nu$*, we have with probability* $1-\delta$

$$V^* - V^\pi \;\leq\; D_\infty\frac{\mathcal{E}_\nu(\mathbb{\Pi},\Pi)}{(1-\gamma)^2} + \varepsilon$$

If $R_{\mathcal{W}}(M) = \sqrt{M\log|\mathcal{W}|}$ *for some measure of weak learning complexity* $|\mathcal{W}|$*, the algorithm samples* $\tilde{O}\left(\frac{C_\infty^4(\mathbb{\Pi})|A|^2\log|\mathcal{W}|}{(1-\gamma)^7\alpha^2\varepsilon^3}\right)$ *episodes in the episodic model, and* $\tilde{O}\left(\frac{D_\infty^4|A|^2\log|\mathcal{W}|}{(1-\gamma)^{12}\alpha^2\varepsilon^4}\right)$ *in the* ν*-reset model.*

C. Proof Overview

Throughout the analysis, we use the notation $\nabla_\pi V^\pi$ to denote the gradient of the value function with respect to the $|S| \times |A|$-sized representation of the policy π, namely the functional gradient of V^π.

C.1. Internal-boosting weak learners

We utilize a variant of the Frank-Wolfe method as a form "internal-boosting" for the weak learners, by employing an adapted analysis of previous work that is stated below.

Note that $\widehat{Q}^\pi(s,\cdot)$ produced by Algorithm 2 satisfies $\|\widehat{Q}^\pi(s,\cdot)\| = \frac{|A|}{1-\gamma}$. We can now borrow the following result on boosting for statistical learning from (Hazan & Singh, 2021), specializing the decision set to be Δ_A. Let $\mathcal{D}_t$ be the distribution induced by the trajectory sampler in round t.

Theorem 13 ((Hazan & Singh, 2021))**.** *Let* $\beta = \sqrt{\frac{1}{\alpha N}}$*, and* $\eta_{2,n} = \min\{\frac{2}{n}, 1\}$*. Then, for any* t*,* π'_t *produced by Algorithm 1 satisfies with probability* $1-\delta$ *that*

$$\max_{\pi\in\Pi} \mathbb{E}_{(s,Q)\sim\mathcal{D}_t}\left[Q^\top\pi(s)\right] - \mathbb{E}_{(s,Q)\sim\mathcal{D}_t}\left[Q^\top\pi'_t(s)\right] \;\leq\; \frac{2|A|}{(1-\gamma)\alpha}\left(\frac{2}{\sqrt{N}} + \varepsilon\right)$$

C.2. From weak learning to linear optimization

In the following Lemma, we give an important observation which allows us to re-state the guarantee in the previous subsection in terms of linear optimization over functional gradients.

Lemma 14. *Applying Algorithm 2 for any given policy π, yields an unbiased estimate of the gradient, such that for any π',*

$$(\nabla_\pi V_\mu^\pi)^\top \pi' = \frac{1}{1-\gamma} \mathbb{E}_{(s,\widehat{Q^\pi}(s,\cdot))\sim\mathcal{D}} \left[\widehat{Q^\pi}(s,\cdot)^\top \pi'(\cdot|s) \right], \tag{1}$$

where $\pi'(\cdot|s) \in \Delta_A$, and $\mathcal{D}$ is the distribution induced on the outputs of Algorithm 2, for a given policy π and initial state distribution μ.

Proof. Recall $\nabla_\pi V^\pi$ denotes the gradient with respect to the $|S| \times |A|$-sized representation of the policy π – the functional gradient. Then, using the policy gradient theorem (Williams, 1992; Sutton et al., 2000), it is given by,

$$\frac{\partial V_\mu^\pi}{\partial \pi(a|s)} = \frac{1}{1-\gamma} d_\mu^\pi(s) Q^\pi(s,a). \tag{2}$$

The following sources of randomness are at play in the sampling algorithm (Algorithm 2): the distribution d^π (which encompasses the discount-factor-based random termination, the transition probability, and the stochasticity of π), and the uniform sampling over A. For a fixed s, π, denote by $\mathcal{Q}_s^\pi$ as the distribution over $\widehat{Q^\pi}(s,\cdot) \in \mathbb{R}^A$, induced by all the aforementioned randomness sources. To conclude the claim, observe that by construction

$$\mathbb{E}_{\mathcal{Q}^\pi(s,\cdot)}[\widehat{Q^\pi}(s,\cdot)|\pi, s] = Q^\pi(s,\cdot). \tag{3}$$

□

C.3. Main result - proof overview

Next we sketch the high-level ideas of the proof of our main result, stated in Theorem 9. We refer the reader to the appendix for the formal proof. First, we provide an abstract high-level procedural template that our RL booster operates in. It is based on a novel variant of the Frank-Wolfe (FW) optimization technique adapted to non-convex and gradient dominated function classes. The broad scheme here is compounded of several components.

We utilize a variant of the FW method as “internal-boosting” for the weak learners (by employing an adapted analysis of Theorem 13), using the observation given in the previous section. This is procedure is enveloped by our novel variant of the FW method for non-convex settings. Lastly, the result stems from demonstrating that the value function satisfy gradient domination properties, in the different settings considered, for which the guarantees of the high-level algorithm hold.

D. Non-convex Frank-Wolfe

In this section, we give an abstract high-level procedural template that the previously introduced RL boosters operate in. This is based on a variant of the Frank-Wolfe optimization technique, adapted to non-convex and gradient dominated function classes (see Definition 15).

Definition 15 (Gradient Domination)**.** A function $f : \mathcal{K} \to \mathbb{R}$ is said to be $(\kappa, \tau, \mathcal{K}_1, \mathcal{K}_2)$-locally gradient dominated (around $\mathcal{K}_1$ by $\mathcal{K}_2$) if for all $x \in \mathcal{K}_1$, it holds that

$$\max_{y\in\mathcal{K}} f(y) - f(x) \ \leq \ \kappa \times \max_{y\in\mathcal{K}_2} \left\{ \nabla f(x)^\top (y-x) \right\} + \tau.$$

The Frank-Wolfe (FW) method assumes oracle access to a black-box linear optimizer, denoted $\mathcal{O}$, and utilizes it by iteratively making oracle calls with modified objectives, in order to solve the harder task of convex optimization. Analogously, boosting algorithms often assume oracle access to a ”weak” learner, which are utilized by iteratively making oracle calls with modified objective, in order to obtain a ”strong” learner, with boosted performance. In the RL setting, the objective is in fact non-convex, but exhibits gradient domination. By adapting Frank-Wolfe technique to this setting, we will in subsequent section obtain guarantees for the algorithms given in Section 4.

Setting. Denote by $\mathcal{O}$ a black-box oracle to an $(\epsilon_0, \mathcal{K}_2)$-approximate linear optimizer over a convex set $\mathcal{K} \subseteq \mathbb{R}^d$ such that for any given $v \in \mathbb{R}^d$, we have

$$v^\top \mathcal{O}(v) \;\geq\; \max_{u \in \mathcal{K}_2} v^\top u - \epsilon_0.$$

Algorithm 5 Non-convex Frank-Wolfe

Input: $T > 0$, objective f, linear optimization oracle $\mathcal{O}$
Choose x_0 arbitrarily.
for $t = 1, \dots, T$ **do**
  Call $z_t = \mathcal{O}(\nabla_{t-1})$, where $\nabla_{t-1} = \nabla f(x_{t-1})$.
  Choose $\eta_t = \min\{1, \frac{2\kappa}{t}\}$ in the gradient-dominated case, else choose η_t so that

$$|LD^2 \eta_t - \nabla_{t-1}^\top (z_t - x_{t-1})| \;\leq\; \epsilon.$$

  Set $x_t = (1-\eta_t) x_{t-1} + \eta_t z_t$.
end for
Return $\bar{x} = x_T$ in the gradient-dominated case, else x_{t-1} with the smallest η_t.

Theorem 16. *Let $f : \mathcal{K} \to \mathbb{R}$ be L-smooth in some norm $\|\cdot\|_*$, H-bounded, and the diameter of $\mathcal{K}$ in $\|\cdot\|_*$ be D. Then, for a $(\epsilon_0, \mathcal{K}_2)$-linear optimization oracle, the output $\bar{x}$ of Algorithm D satisfies*

$$\max_{u \in \mathcal{K}_2} \nabla f(\bar{x})^\top (u - \bar{x}) \;\leq\; \sqrt{\frac{2HLD^2}{T}} + 3\epsilon + \epsilon_0.$$

Furthermore, if f is $(\kappa, \tau, \mathcal{K}_1, \mathcal{K}_2)$-locally gradient-dominated and $x_0, \dots x_T \in \mathcal{K}_1$, then it holds

$$\max_{x^* \in \mathcal{K}} f(x^*) - f(\bar{x}) \;\leq\; \frac{2\kappa^2 \max\{LD^2, H\}}{T} + \tau + \kappa\epsilon_0.$$

E. Analysis for Boosting with Supervised Learning (Proof of Theorem 9)

Proof of Theorem 9. The broad scheme here is to utilize an equivalence between Algorithm 1 and Algorithm D on the function V^π (or V^π_ν in the ν-reset model), to which Theorem 16 applies.

To this end, firstly, note V^π is $\frac{1}{1-\gamma}$-bounded. Define a norm $\|\cdot\|_{\infty,1} : \mathbb{R}^{|S|\times|A|} \to \mathbb{R}$ as $\|x\|_{1,\infty} = \max_{s\in S} \sum_{a\in A} |x_{s,a}|$. Further, observe that for any policy $\pi : S \to \Delta_A$, $\|\pi\|_{\infty,1} = 1$. The following lemma specifies the smoothness of V^π in this norm.

Lemma 17. *V^π is $\frac{2\gamma}{(1-\gamma)^3}$-smooth in the $\|\cdot\|_{\infty,1}$ norm.*

To be able to interpret Algorithm 1 as an instantiation of the algorithmic template Algorithm D presents, we advance two claims: one, the step-size choices of the two algorithms conincide; two, π'_t (Line 3-10) serves as an approximate linear optimizers for $\nabla V^{\pi_{t-1}}$. Together, these imply that the iterates produced by the two algorithms conincide. The first of these, which provides a value of ϵ to use in the statement of Theorem 16, is established below.

Claim 18. *Upon every invocation of `StepChooser`, the output $\eta_{1,t}$ satisfies with probability $1-\delta$*

$$\left| \frac{2\eta_{1,t}}{(1-\gamma)^3} - (\nabla V_\mu^{\pi_{t-1}})^\top (\pi'_t - \pi_{t-1}) \right| \;\leq\; \frac{16|A|}{(1-\gamma)^2\sqrt{P}} \log\frac{1}{\delta}$$

Next, we move onto the linear optimization equivalence. Indeed, Claim 19 demonstrates that π'_t serves a linear optimizer over gradients of the function V^π; the suboptimality specifies ϵ_0.

Claim 19. *Let* $\beta = \sqrt{\frac{1}{\alpha N}}$*, and* $\eta_{2,n} = \min\{\frac{2}{n}, 1\}$*. Then, for any t,* π'_t *produced by Algorithm 1 satisfies with probability* $1-\delta$

$$\max_{\pi\in\Pi} (\nabla V_\mu^{\pi_{t-1}})^\top (\pi - \pi'_t) \;\leq\; \frac{2|A|}{(1-\gamma)^2\alpha}\left(\frac{2}{\sqrt{N}} + \varepsilon_W\right)$$

Finally, observe that it is by construction that $\pi_t \in \sqcap$. Therefore, in terms of the previous section, $\mathcal{K}$ is the class of all policies, $\mathcal{K}_1 = \sqcap$, $\mathcal{K}_2 = \Pi$.

In the episodic model, we wish to invoke the second part of Theorem 16. The next lemma establishes gradient-domination properties of V^π to support this.

Lemma 20. V^π *is* $\left(C_\infty(\sqcap), \frac{1}{1-\gamma}C_\infty(\sqcap)\mathcal{E}(\Pi,\sqcap), \sqcap, \Pi\right)$*-gradient dominated, i.e. for any* $\pi\in\sqcap$*:*

$$V^* - V^\pi \;\leq\; C_\infty(\sqcap)\left(\frac{1}{1-\gamma}\mathcal{E}(\Pi,\sqcap) + \max_{\pi'\in\Pi}(\nabla V^\pi)^\top(\pi' - \pi)\right)$$

Deriving κ, τ from the above lemma along with ϵ_0 from Claim 19 and ϵ from Claim 18, as a consequence of the second part of Theorem 16, we have with probability $1 - NT\delta$

$$V^* - V^{\bar\pi} \;\leq\; C_\infty(\sqcap)\frac{\mathcal{E}(\sqcap,\Pi)}{1-\gamma} + \frac{4C_\infty^2(\sqcap)}{(1-\gamma)^3 T} + \frac{4|A|C_\infty(\sqcap)}{(1-\gamma)^2\alpha\sqrt{N}} + \frac{2|A|C_\infty(\sqcap)}{(1-\gamma)^2\alpha}\varepsilon_W.$$

Similarly, in the ν-reset model, the first part of Theorem 16 provides a local-optimality guarantee for V_ν^π. Lemma 21 provides a bound on the function-value gap (on V^π) provided such local-optimality conditions.

Lemma 21. *For any* $\pi\in\sqcap$*, we have*

$$V^* - V^\pi \;\leq\; \frac{1}{1-\gamma}D_\infty\left(\frac{1}{1-\gamma}\mathcal{E}_\nu(\Pi,\sqcap) + \max_{\pi'\in\Pi}(\nabla V_\nu^\pi)^\top(\pi' - \pi)\right)$$

Again, using the bound on $\max_{\pi'\in\Pi}(\nabla V_\nu^{\bar\pi})^\top(\pi' - \bar\pi)$ Theorem 16 provides, we have that with probability $1 - 2NT\delta$

$$V^* - V^{\bar\pi} \;\leq\; \frac{D_\infty\mathcal{E}_\nu(\sqcap,\Pi)}{(1-\gamma)^2} + \frac{2D_\infty}{(1-\gamma)^3\sqrt{T}} + \frac{2|A|D_\infty}{(1-\gamma)^3\alpha}\left(\frac{2}{\sqrt{N}} + \varepsilon_W\right) + \frac{48|A|D_\infty}{(1-\gamma)^3\sqrt{P}}\log\frac{1}{\delta}$$

□

F. Analysis for Boosting with Online Learning (Proof of Theorem 12)

Proof of Theorem 12. Similar to the proof of Theorem 9, we establish an equivalence between Algorithm 1 and Algorithm D on the function V^π (or V_ν^π in the ν-reset model), to which Theorem 16 applies provided smoothness (see Lemma 17).

Indeed, Claim 22 demonstrates π'_t serves a linear optimizer over gradients of the function V^π, and provides a bound on ϵ_0. Claim 18 ensures that that the step size choices (and hence iterates) of the two algorithms coincide. As before, observe that it is by construction that $\pi_t \in \sqcap$.

Claim 22. *Let* $\beta = \sqrt{\frac{1}{\alpha N}}$*, and* $\eta_{2,n} = \min\{\frac{2}{n}, 1\}$*. Then, for any t,* π'_t *produced by Algorithm 4 satisfies with probability* $1-\delta$

$$\max_{\pi\in\Pi}(\nabla V_\mu^{\pi_{t-1}})^\top(\pi - \pi'_t) \;\leq\; \frac{2|A|}{(1-\gamma)^2\alpha}\left(\frac{2}{\sqrt{N}} + \frac{R_{\mathcal{W}}(M)}{M} + \sqrt{\frac{16\log\delta^{-1}}{M}}\right)$$

In the episodic model, one may combine the second part of Theorem 16, which provides a bound on function-value gap for gradient dominated functions, which Lemma 20 guarantees, to conclude with probability $1 - T\delta$

$$V^* - V^{\bar\pi} \;\leq\; \frac{C_\infty(\sqcap)\mathcal{E}(\sqcap,\Pi)}{1-\gamma} + \frac{4C_\infty^2(\sqcap)}{(1-\gamma)^3T} + \frac{4|A|C_\infty(\sqcap)}{(1-\gamma)^2\alpha\sqrt{N}} + \frac{2|A|C_\infty(\sqcap)}{(1-\gamma)^2\alpha}\frac{R_{\mathcal{W}}(M)}{M} + \frac{8|A|C_\infty(\sqcap)\log\delta^{-1}}{(1-\gamma)^2\alpha\sqrt{M}}.$$

Similarly, in the ν-reset model, Lemma 21 provides a bound on the function-value gap provided local-optimality conditions, which the first part of Theorem 16 provides for. Again, with probability $1 - T\delta$

$$V^* - V^{\bar{\pi}} \;\leq\; \frac{D_\infty \mathcal{E}_\nu(\mathbb{\Pi}, \Pi)}{(1-\gamma)^2} + \frac{2D_\infty}{(1-\gamma)^3}\left(\frac{1}{\sqrt{T}} + \frac{|A|}{\alpha}\left(\frac{2}{\sqrt{N}} + \frac{R_{\mathcal{W}}(M)}{M} + \frac{4\log\delta^{-1}}{\sqrt{M}}\right) + \frac{24|A|}{\sqrt{P}}\log\frac{1}{\delta}\right)$$

□

G. Proofs of Supporting Claims

G.1. Non-convex Frank-Wolfe method (Theorem 16)

Proof of Theorem 16. **Non-convex case.** Note that for any timestep t, it holds due to smoothness that

$$\begin{aligned}
f(x_t) &= f(x_{t-1} + \eta_t(z_t - x_{t-1})) \\
&\geq\; f(x_{t-1}) + \eta_t \nabla_{t-1}^\top (z_t - x_{t-1}) - \eta_t^2 \frac{L}{2} D^2 \\
&= f(x_{t-1}) - \frac{1}{2LD^2}\left(LD^2\eta_t - \nabla_{t-1}^\top(z_t - x_{t-1})\right)^2 + \frac{(\nabla_{t-1}^\top(z_t - x_{t-1}))^2}{2LD^2}
\end{aligned}$$

Using the step-size definition to bound on the middle term, and telescoping this inequality over function-value difference values across successive iterates, we have

$$\min_t (\nabla_{t-1}^\top(z_t - x_{t-1}))^2 \;\leq\; \frac{1}{T}\sum_{t=1}^{T}(\nabla_{t-1}^\top(z_t - x_{t-1}))^2 \;\leq\; \frac{2LD^2H}{T} + \epsilon^2$$

Let $t' = \arg\min_t \eta_t$ and $t^* = \arg\min_t (\nabla_{t-1}^\top(z_t - x_{t-1}))^2$. Then

$$\begin{aligned}
\nabla_{t'-1}^\top(z_{t'} - x_{t'-1}) \;&\leq\; LD^2\eta_{t'} + \epsilon \;\leq\; LD^2\eta_{t^*} + \epsilon \\
&\leq\; \nabla_{t^*-1}^\top(z_{t^*} - x_{t^*-1}) + 2\epsilon \;\leq\; \sqrt{\frac{2LD^2H}{T} + \epsilon^2} + 2\epsilon
\end{aligned}$$

To connclude the claim for the non-convex part, observe $\sqrt{a+b} \;\leq\; \sqrt{a} + \sqrt{b}$ for $a, b > 0$, and that since $z_{t'} = \mathcal{O}(\nabla_{t'-1})$, it follows by oracle definition that

$$\max_{u \in \mathcal{K}_2} \nabla_{t'-1}^\top u \;\leq\; \nabla_{t'-1}^\top z_{t'} + \epsilon_0.$$

Gradient-Dominated Case. Define $x^* = \arg\max_{x \in \mathcal{K}} f(x)$ and $h_t = f(x^*) - f(x_t)$.

$$\begin{aligned}
h_t \;&\leq\; h_{t-1} - \eta_t \nabla_{t-1}^\top(z_t - x_{t-1}) + \eta_t^2 \frac{L}{2} D^2 && \text{smoothness} \\
&\leq\; h_{t-1} - \eta_t \max_{y \in \mathcal{K}_2} \eta_t \nabla_{t-1}^\top (y - x_{t-1}) + \eta_t^2 \frac{L}{2} D^2 + \eta_t \epsilon_0 && \text{oracle} \\
&\leq\; h_{t-1} - \frac{\eta_t}{\kappa}(f(x^*) - f(x_{t-1})) + \eta_t^2 \frac{L}{2} D^2 + \eta_t\left(\epsilon_0 + \frac{\tau}{\kappa}\right) && \text{gradient domination} \\
&= \left(1 - \frac{\eta_t}{\kappa}\right) h_{t-1} + \eta_t^2 \frac{L}{2} D^2 + \eta_t\left(\epsilon_0 + \frac{\tau}{\kappa}\right)
\end{aligned}$$

The theorem now follows from the following claim.

Claim 23. *Let $C \;\geq\; 1$. Let g_t be a H-bounded positive sequence such that*

$$g_t \;\leq\; \left(1 - \frac{\sigma_t}{C}\right) g_{t-1} + \sigma_t^2 D + \sigma_t E.$$

Then choosing $\sigma_t = \min\{1, \frac{2C}{t}\}$ implies $g_t \;\leq\; \frac{2C^2 \max\{2D, H\}}{t} + CE$.

□

G.2. Smoothness of value function (Lemma 17)

Proof of Lemma 17. Consider any two policies π, π'. Using the Performance Difference Lemma (Lemma 3.2 in (Agarwal et al., 2019), e.g.) and Equation 1, we have

$$\begin{aligned}
&|V^{\pi'} - V^{\pi} - \nabla V^{\pi}(\pi' - \pi)| \\
=&\frac{1}{1-\gamma}\left|\mathbb{E}_{s\sim d^{\pi'}}\left[Q^{\pi}(\cdot|s)^\top(\pi'(\cdot|s) - \pi(\cdot|s)\right] - \mathbb{E}_{s\sim d^{\pi}}\left[Q^{\pi}(\cdot|s)^\top(\pi'(\cdot|s) - \pi(\cdot|s)\right]\right| \\
\leq& \frac{1}{(1-\gamma)^2}\|d^{\pi'} - d^{\pi}\|_1\|\pi' - \pi\|_{\infty,1}
\end{aligned}$$

The last inequality uses the fact that $\max_{s,a} Q^\pi(s,a) \leq \frac{1}{1-\gamma}$. It suffices to show $\|d^{\pi'} - d^{\pi}\|_1 \leq \frac{\gamma}{1-\gamma}\|\pi' - \pi\|_{\infty,1}$. To establish this, consider the Markov operator $P^\pi(s'|s) = \sum_{a\in A} P(s'|s,a)\pi(a|s)$ induced by a policy π on MDP M. For any distribution d supported on S, we have

$$\begin{aligned}
\|(P^{\pi'} - P^{\pi})d\|_1 &= \sum_{s'}\left|\sum_{s,a} P(s'|s,a)d(s)(\pi'(a|s) - \pi(a|s)\right| \\
&\leq \sum_{s'} P(s'|s,a)\|d\|_1\|\pi' - \pi\|_{\infty,1} \leq \|\pi' - \pi\|_{\infty,1}
\end{aligned}$$

Using sub-additivity of the l_1 norm and applying the above observation t times, we have for any t

$$\|((P^{\pi'})^t - (P^{\pi})^t)d\|_1 \leq t\|\pi' - \pi\|_{\infty,1}.$$

Finally, observe that

$$\begin{aligned}
\|d^{\pi'} - d^{\pi}\|_1 &\leq (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\|((P^{\pi'})^t - (P^{\pi})^t)d_0\|_1 \\
&\leq \|\pi' - \pi\|_{\infty,1}(1-\gamma)\sum_{t=0}^{\infty} t\gamma^t = \frac{\gamma}{1-\gamma}\|\pi' - \pi\|_{\infty,1}
\end{aligned}$$

□

G.3. Step-size guarantee (Claim 18)

Proof of Claim 18. Let $\mathcal{D}$ be the distribution induced by Algorithm 2 upon being given π_{t-1}. Due to Lemma 14, it suffices to demonstrate that for any $\pi \in \{\pi'_t, \pi_{t-1}\}$ the following claim holds with probability $1 - \frac{\delta}{2}$. The claim in turn follows from Hoeffding's inequality, while noting $\widehat{Q^{\pi_{t-1}}}(s,\cdot)$ is $\frac{|A|}{1-\gamma}$-bounded in the l_∞ norm.

$$\left|\widehat{G^\pi} - \mathbb{E}_{(s,\widehat{Q^{\pi_{t-1}}}(s,\cdot))\sim\mathcal{D}}\left[\widehat{Q^{\pi_{t-1}}}(s,\cdot)^\top\pi(\cdot|s)\right]\right| \leq \frac{8|A|}{(1-\gamma)\sqrt{P}}\log\frac{1}{2\delta}$$

□

G.4. Gradient domination (Lemma 20 and Lemma 21)

Proof of Lemma 20. Invoking Lemma 4.1 from (Agarwal et al., 2019) with $\mu = d_0$, we have

$$\begin{aligned}
V^* - V^\pi &\leq \left\|\frac{d^{\pi^*}}{d^\pi}\right\|_\infty \max_{\pi_0}(\nabla V^\pi)^\top(\pi_0 - \pi) \\
&\leq C_\infty(\sqcap)(\max_{\pi_0}(\nabla V^\pi)^\top\pi_0 - \max_{\pi'\in\Pi}(\nabla V^\pi)^\top\pi' + \max_{\pi'\in\Pi}(\nabla V^\pi)^\top(\pi' - \pi))
\end{aligned}$$

Finally, with the aid of Equation 1, observe that

$$\begin{aligned}
\max_{\pi_0}(\nabla V^\pi)^\top\pi_0 - \max_{\pi'\in\Pi}(\nabla V^\pi)^\top\pi' &= \min_{\pi'\in\Pi}\frac{1}{1-\gamma}\mathbb{E}_{s\sim d^\pi}\left[\max_a Q^\pi(s,a) - Q^\pi(\cdot|s)^\top\pi'\right] \\
&\leq \frac{1}{1-\gamma}\mathcal{E}(\Pi, \sqcap)
\end{aligned}$$

□

Proof of Lemma 21. Invoking Lemma 4.1 from (Agarwal et al., 2019) with $\mu = \nu$, we have

$$\begin{aligned} V^* - V^\pi &\leq \frac{1}{1-\gamma} \left\| \frac{d^{\pi^*}}{\nu} \right\|_\infty \max_{\pi_0} (\nabla V_\nu^\pi)^\top (\pi_0 - \pi) \\ &\leq \frac{1}{1-\gamma} D_\infty (\max_{\pi_0} (\nabla V_\nu^\pi)^\top \pi_0 - \max_{\pi' \in \Pi} (\nabla V_\nu^\pi)^\top \pi' + \max_{\pi' \in \Pi} (\nabla V_\nu^\pi)^\top (\pi' - \pi)) \end{aligned}$$

Again, with the aid of Equation 1, observe that

$$\begin{aligned} \max_{\pi_0} (\nabla V_\nu^\pi)^\top \pi_0 - \max_{\pi' \in \Pi} (\nabla V_\nu^\pi)^\top \pi' &= \min_{\pi' \in \Pi} \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\nu^\pi} \left[\max_a Q^\pi(s,a) - Q^\pi(\cdot|s)^\top \pi' \right] \\ &\leq \frac{1}{1-\gamma} \mathcal{E}_\nu(\Pi, \mathbb{\Pi}) \end{aligned}$$

□

G.5. Supervised linear optimization guarantees (Claim 19)

Proof of Claim 19. The subroutine presented in lines 3-10 (which culminate in π'_t) is an instantiation of Algorithm 3 from (Hazan & Singh, 2021), specializing the decision set to be Δ_A. To note the equivalence, note that in (Hazan & Singh, 2021) the algorithm is stated assuming that the center-of-mass of the decision set is at the origin (after a coordinate transform); correspondingly, the update rule in Algorithm 1 can be written as

$$(\rho_{t,n} - \pi_r) = (1 - \eta_{2,n})(\rho_{t,n-1} - \pi_r) + \frac{\eta_{2,n}}{\alpha} (\mathcal{A}_{t,n} - \pi_r).$$

For any state s, $\pi_r(\cdot|s) = \frac{1}{A} \mathbf{1}_{|A|}$ corresponds to the center-of-masss of Δ_A. Finally, note that maximizing $f^\top x$ over $x \in \mathcal{K}$ is equivalent to minimizing $(-f)^\top x$ over the same domain. Therefore, we can borrow the following result on boosting for statistical learning from (Hazan & Singh, 2021) (Theorem 13). Note that $\widehat{Q^\pi}(s, \cdot)$ produced by Algorithm 2 satisfies $\|\widehat{Q^\pi}(s, \cdot)\| = \frac{|A|}{1-\gamma}$. Let $\mathcal{D}_t$ be the distribution induced by the trajectory sampler in round t.

Theorem 24 ((Hazan & Singh, 2021))**.** *Let $\beta = \sqrt{\frac{1}{\alpha N}}$, and $\eta_{2,n} = \min\{\frac{2}{n}, 1\}$. Then, for any t, π'_t produced by Algorithm 1 satisfies with probability $1 - \delta$ that*

$$\max_{\pi \in \Pi} \mathbb{E}_{(s,Q) \sim \mathcal{D}_t} \left[Q^\top \pi(s) \right] - \mathbb{E}_{(s,Q) \sim \mathcal{D}_t} \left[Q^\top \pi'_t(s) \right] \leq \frac{2|A|}{(1-\gamma)\alpha} \left(\frac{2}{\sqrt{N}} + \varepsilon \right)$$

Lemma 14 allows us to restate the guarantees in the previous subsection in terms of linear optimization over functional gradients. The conclusion thus follows immediately by combining Lemma 14 and Theorem 24. □

G.6. Online linear optimization guarantees (Claim 22)

Proof of Claim 22. In a similar vein to the proof of Claim 19, here we state the a result on boosting for online convex optimization (OCO) from (Hazan & Singh, 2021) (Theorem 6), the counterpart of Theorem 13 for the online weak learning case.

Theorem 25 ((Hazan & Singh, 2021))**.** *Let $\beta = \sqrt{\frac{1}{\alpha N}}$, and $\eta_{2,n} = \min\{\frac{2}{n}, 1\}$. Then, for any t, $\Gamma[\rho_{t,m,N}]$ produced by Algorithm 4 satisfies*

$$\max_{\pi \in \Pi} \sum_{m=1}^{M} \left[\hat{Q}_{t,m}^\top \pi(s_{t,m}) \right] - \sum_{m=1}^{M} \left[\hat{Q}_{t,m}^\top \Gamma[\rho_{t,m,N}](s_{t,m}) \right] \leq \frac{2|A|}{(1-\gamma)\alpha} \left(\frac{2M}{\sqrt{N}} + R_{\mathcal{W}}(M) \right)$$

Next we invoke online-to-batch conversions. Note that in Algorithm 4, $(s_{t,m}, \hat{Q}_{t,m})$ for any fixed t is sampled i.i.d. from the same distribution. Therefore, we can apply online-to-batch results, i.e. Theorem 9.5 in (Hazan, 2019), on Theorem 25 to get

$$\max_{\pi\in\Pi} \mathbb{E}_{(s,Q)\sim\mathcal{D}_t}\left[Q^\top \pi(s)\right] - \mathbb{E}_{(s,Q)\sim\mathcal{D}_t}\left[Q^\top \pi'_t(s)\right] \;\leq\; \frac{2|A|}{(1-\gamma)\alpha}\left(\frac{2}{\sqrt{N}} + \frac{R_{\mathcal{W}}(M)}{M} + \sqrt{\frac{16\log\delta^{-1}}{M}}\right)$$

We finally invoke Lemma 14. □

G.7. Remaining proofs (Claim 23 & Claim 5)

Proof of Claim 23. Let $T^* = \arg\max_t\{t : t \leq 2C\}$. For any $t \leq T^*$, we have $\sigma_t = 1$ and $g_t \leq H \leq \frac{2C^2 H}{t}$. For $t \geq T^*$, we proceed by induction. The base case ($t = T^*$) is true by the previous display. Now, assume $g_{t-1} \leq \frac{2C^2\max\{2D,H\}}{t-1} + CE$ for some $t > T^*$.

$$\begin{aligned} g_t &\leq \left(1-\frac{2}{t}\right)\left(\frac{2C^2\max\{2D,H\}}{t-1} + CE\right) + \frac{4C^2 D}{t^2} + \frac{2CE}{t} \\ &\leq CE + 2C^2\max\{2D,H\}\left(\frac{1}{t-1}\left(1-\frac{2}{t}\right) + \frac{1}{t^2}\right) \\ &= CE + 2C^2\max\{2D,H\}\frac{t^2-2t+t-1}{t^2(t-1)} \\ &\leq CE + 2C^2\max\{2D,H\}\frac{t(t-1)}{t^2(t-1)} \end{aligned}$$

□

Proof of Claim 5. Since $\pi \in \sqcap(\Pi, N, T)$, it is composed of TN base policies. Producing each aggregated function takes NA additions and multiplications; there are T of these. Each projection takes time equivalent to sorting $|A|$ numbers, due to a water-filling algorithm (Duchi et al., 2008); these are also T in number. The final linear transformation takes an additional TA operations. □