Common Questions

"Expert" policies. We now see how the term "expert" can be misleading. In the revision, we will replace it with "oracle" and change the paper title to "Policy Improvement via Imitation of Multiple Oracles". While we allow weaker oracle policies than is typical in imitation learning literature, they still need to have meaningful behaviors in order to provide an informative advantage function for policy update. For example, they cannot be completely uniform behavior policies as is often done in the batch RL setting.

Extra experiments. We plan to include more results on harder domains. In Fig. 1, we present a preliminary results of Humanoid-v2 in Mujoco, where we downloaded the pretrained policy from OpenAI baseline as the oracle. We see MAMBA significantly improves learning in this single expert case, as it uses λ compared with AggreVaTeD and uses the oracle value compared with PG-GAE. (The initial policy for MAMBA and AggreVaTeD are pretrained with BC).

Reviewer 1

Estimation of values (L115). The values of the oracle policies are estimated by MC in line 4 of Algorithm 1. They are used as the target to train a neural-net $\widehat{V}^k$, which is later used in the gradient computation in line 6 for policy update.

Value estimation complexity (L141). Correct, it depends on the horizon, which is captured by β and ν in Thm 1. The remark in line 141 is for the idealized case when the value functions (or rather a full MDP model) are known. Gradient estimation bias and variance can increase with horizon when learning is involved. We will further clarify this in the revision.

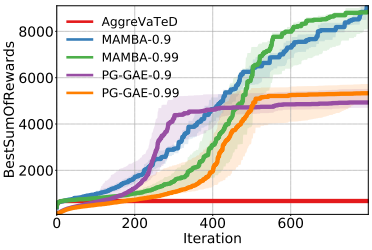

Figure 1: Results of 4 seeds. The values of λ are given in the legend.

Comparison with $\pi^\bullet$. We remark that both $\pi^\bullet$ and $\pi^{\max}$ can be learned without explicit value estimation. For example, once we have an estimate of $\pi^\bullet$ for picking the best oracle in a given state, we can use it in an RIRO setting to estimate $V^{\max}(s')$ and construct $r(s,a)+\mathbb{E}[V^{\max}(s')]$, which provides enough information to learn $\pi^{\max}$ (since changing the state dependent baseline in the advantage doesn't affect the policy derived from it). However this approach suffers from high variance in our experiences; that is why we designed MAMBA based on value function estimators. We recall also that $\pi^{\max}$ corresponds to one-step policy improvement while $\pi^\bullet$ corresponds to behavior cloning in the idealized case of a known value function with only one expert, hence we end up not considering $\pi^\bullet$.

Comparison with BC/DAgger. While MAMBA does require value estimation, this lets MAMBA 1) combine multiple oracle policies' strengths and 2) further improve upon their performance. Without the value information, this is impossible in general, because there is no signal to distinguish the quality or preference of different oracle policies.

Reviewer 2

Related work. Thanks for extra pointers! We'll move more rel. work discussion to the main text and include them there.

Properties of λ. Note that Eq (13) provides the definition of $A^{max,\pi}_\lambda$ used in Eq (12). For (13), when $\lambda=1$, because the sum in Eq (13) is *infinite* and the problem horizon T is *finite*, we have $A^{max,\pi}_\lambda(s_t,a_t)=\mathbb{E}_{\rho^\pi}[\sum_{\tau=t}^{T-1} r(s_\tau,a_\tau)]-f^{max}(s_t)$ (note $A^{max,\pi}_{(i)}(s_t,a_t)$ is the same for all $i>T-t-1$). Therefore, $\ell_n(\pi,1)=-\mathbb{E}_{\rho^\pi}[\sum_{\tau=t}^{T-1} r(s_\tau,a_\tau)]+f^{max}(d_0)$, which is the original RL problem. On the other hand for $\lambda=0$, $A^{max,\pi}_\lambda(s_t,a_t)=A^{max,\pi}_{(0)}(s_t,a_t)=r(s_t,a_t)+\mathbb{E}[f^{max}(s_{t+1})]-f^{max}(s_t)$ in Eq (13) (we use $0^0=1$). We will clarify and better motivate these equations.

L246 & L249-252. The estimator in (14) assumes $\pi=\pi_n$ (it's a typo). We meant that λ controls the bias and variance of (14) compared with the true gradient of (12) at $\pi=\pi_n$, as the effects of the function approximator decays as $\lambda\to 1$.

Experiments. The oracle policies here are obtained by prematurely terminating policy training initialized with different random seeds. We will include an ablation study of UpdateInputWhitening in the revision. We believe the result will be similar, since the only difference is in pre-training; after that, the algorithm collects the same amount of samples per iteration regardless of the number of oracle policies. We will include the range of oracle performance in the plots.

Misc. We can view Natural Gradient Descent as an instantiation of the first-order algorithm Mirror Descent, which uses Fisher information matrix to define Bregman divergence. AggreVaTe does not require action information, too.

Reviewer 3

Clarity. Thank you for the suggestions on improving the clarity! The version for general f is stated in the appendix as Proposition 2 and 3. We will draw a clearer connection in the revision.

Line 207-208. It's a typo. We meant $\mathbb{E}_{s\sim d_0}[\max_{k\in[K]} V^k(s)]+\Delta_N-o(1)$, where $o(1)$ is due to no-regret assumption.

Algorithm. We sample one trajectory using the learner's policy to estimate the on-policy gradient in Eq (14). We sample the other one to learn the value function of the oracle policies, which requires the RIRO setting.