First of all, we would like to thank all reviewers for their insightful comments and suggestions!

Reviewer #1

Adding a description about policy iteration in Sec 3.2. Thanks for the suggestion. We will add a paragraph to provide background materials on policy iteration.

Do other norms work? Yes, we can also use L_1-norm or L_∞-norm, in which case the optimization problem becomes a linear program. But in the field of reinforcement learning (RL), L_2-norm is the most common choice due to its efficiency and effectiveness. Thus we adopt L_2-norm in the paper to ensure consistency between the objective of Anderson acceleration (AA) and the loss of Q-value function (critic).

Minors. Thanks! We will fix these issues in the revision.

Reviewer #2

Impact of the number of previous estimates m. Indeed, there is a tradeoff between performance and computational cost. We have analyzed the impact of using different m during the rebuttal period, and part of the results are shown in Fig.1. Overall, larger m leads to better performance, but the improvement becomes small when m exceeds a threshold. These additional experiments will be added to Sec. 5.3 (ablation studies) in the revision.

Value of m. We set m to 5 in our experiments. The detailed hyperparameter settings are given in Appendix C, where "number of previous estimates" corresponds to m. (Sorry for the broken link in Line 245.)

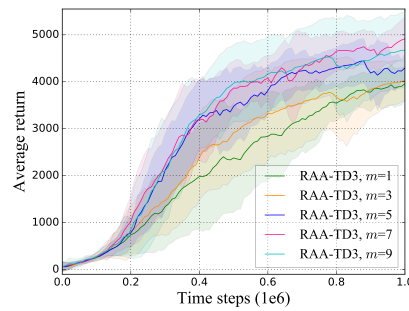

Figure 1: Performance of RAA-TD3 on Walker2d-v2 with different m.

Impact of the error/perturbations on the final solution found by RAA? Indeed, it is meaningful to construct error bounds for the value function. However, this seems to be difficult in the context of *deep* RL. First, the approximation error of value function largely originates from the deep neural networks, for which the generalization error bound is difficult to obtain. Second, there is no explicit connection between the error of coefficient vector α and the error of value function. Fortunately, for the coefficient vector α, we can still provide a clear connection between regularization and the approximation error, as shown by Proposition 1 in line 187-197. Constructing error bounds for value function under the setting of linear or other interpretable function approximators is an interesting topic for future work.

Minors. Thanks for your detailed comments! We will carefully fix all the minor issues into our revision.

Reviewer #3

Performance benefit. Please note that our motivation of introducing RAA is to improve the *sample efficiency* (convergence speed) of deep RL, instead of improving the final performance. Fig.1 in our paper shows that RAA-based RL algorithms generally require half the number of samples to achieve comparable performance as the counterparts without RAA, which is a significant boost in terms of *efficiency*. Interestingly, due to variance reduction of approximation error in the target values, our algorithm also improves the final performance in most of the cases. In other words, RAA not only substantially improves the sample efficiency of deep RL, but also boosts the final performance in many cases.

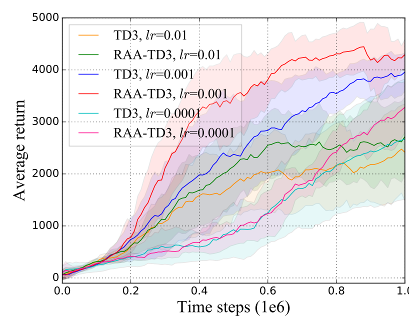

Figure 2: Performance comparison on Walker2d-v2 with different learning rates.

Conflating factors. In fact, we have tried our best to isolated all conflating factors in our experiments: we picked state-of-the-art (SOTA) deep RL algorithm and simply add the proposed RAA module to it *without changing any of the hyperparameters* (including step-size and the optimizer). This means that (1) our baselines are very competitive; (2) we used exactly the same hyperparameters as the baselines; and (3) the only difference between our algorithm and the baselines is using or not using RAA. Therefore, we believe our experiments are fair.

A sweep of step-sizes. During the rebuttal period, we performed additional experiments to compare the behavior of RAA over different learning rates (lr), and the results are shown in Fig.2. Overall, the improvement of our method is consistent across all learning rates, and the improvement is more significant when the learning rate is smaller. Additional experiments will be added to Sec. 5.3 (ablation studies) in the revision.

Momentum terms could mimic some benefits of RAA? Indeed, momentum also aggregates information across iterations and leads to faster convergence. But as noted above, our baselines are SOTA deep RL algorithms, which are already equipped with advanced momentum-based optimizer (e.g., ADAM). This means RAA is compatible with momentum and can further speed up the convergence.

Assess RAA in deep RL regime. This seems to be a misunderstanding. We actually focus on deep RL in this work.