We would like to thank all the reviewers for their comments and feedback. We reply to the reviews in order.

Move algorithm pseudocode to main text. We agree it deserves to be part of the main text; we will move it there.

R#1. *Theoretical analysis.* Providing theoretical guarantees with function approximation is hard, and few papers do so in RL. When data goes to infinity and the true value function falls in our family of models, the MC intervals will converge to the true values and Adaptive TD will be forced to converge too. However, these are asymptotic results.

Parameter tuning justification. Adaptive TD seems to be robust to the choice of hyper-parameters. We illustrate this by simply choosing typical values for the hyper parameters, without doing an exhaustive search for the best values. For example, we take alpha to be 95% confidence intervals, as this is standard in most statistical work. Even with this off-the-shelf choice for all environments, Adaptive TD shows strong experimental performance and robustness.

Performance of different choices of confidence interval L_{MC} and U_{MC}. We can definitely assume distributions $\mathbf{F}$ other than Gaussian, or bootstrap estimates, leading to different confidence intervals. We also tried some naive ideas like setting $L_{MC} = \min_i v_i$ and $U_{MC} = \max_i v_i$. However, this did not perform as well as the Gaussian intervals.

R#2. *Add other benchmark algorithms from the field of approximate dynamic programming.* In addition to MC and TD, in our experiments we also show the most relevant competitor: TD(λ) baselines for a number of different λ's.

R#3. *The computational complexity increases with time.* We are not sure what is meant here. Our algorithm fits m+1 networks rather than 1, on the same data (m for MC, 1 for TD). Therefore, it requires m times more work. For small m like m=3 or m=5, this is still reasonable. Fortunately, the training of the m MC networks is completely parallelizable. One can also train the MC networks for much fewer steps (as they tend to converge way faster as they don't bootstrap).

What is the practical problem that is solved here? On-Policy evaluation from log data is a very important practical problem by itself (e.g. recommender systems). Using Adaptive TD within the full RL loop (policy improvement) is left for future research, as it involves different trade-offs between computation, accuracy and sample efficiency.

If the MC target is not reliable enough to be used why would these networks' inferred estimates be more reliable? Assume first that the function approximation family is rich enough. The MC estimates should be unbiased or, at least, should have low bias. As a consequence, if the MC target is not "reliable", this means the true underlying variance of the distribution of values $\mathbf{F}$, σ^2 above (5), must be large. The point is that our goal is not to get accurate MC estimates, but just to detect when TD estimates are not reasonable. When variance is large, while our MC estimates may not be reliable, we expect to end up with wide confidence intervals, thus increasing the chances of accepting the TD estimate as a plausible one. On the other hand, if the MC estimates are biased –so they aren't reliable while the variance can still be small–, there is no reason to believe MC or TD (usually even more biased) will do a better job than Adaptive TD.

How much is the algorithm affected by the Gaussianity assumption? It's important to note that each network is trying to fit the value function: $V(s) = E_\pi[R(s)]$. While the return distribution $R(s)$ may be complex (e.g. multimodal), $\mathbf{F}$ models the disagreement in our predictions for its expectation $V(s)$ as a function of iid training runs. We expect a smoother and better behaved distribution then, and Gaussianity may be a reasonable assumption. If we still think Gaussian is a strong assumption, then we have two options. First, if we have a better guess for $\mathbf{F}$ (e.g. heavy tailed), we can simply use it to compute the confidence intervals. Second, if the distribution is unknown, we can use a non-parametric bootstrap approach to approximate the predictive confidence interval directly from our samples $v_{1:n}$.

How are m, α and the confidence threshold chosen? If for example we pick m to be large, this would create very small confidence intervals which would reduce the algorithm to always using MC estimates (since the TD estimate is always outside)? This is not true. $\mathbf{F}$ (line 192) is determined by the training data and algorithm, not by m. In particular, the confidence intervals become narrower with more data, as the value of the true parameter σ^2 ideally decreases (under a reasonable training algorithm). Under the Gaussian assumption, $\bar{v}$ and $\hat{\sigma}^2_m$ will tend to their true values for large m, and the TD estimate will need to fall in the confidence interval with respect to the true distribution $\mathbf{F}$. Thus, for larger m the algorithm does not reduce to always using MC. However, when the amount of data tends to infinity, and assuming the true value function belongs to our family of models, Adaptive TD reduces to MC (as desired). Actually, we did some experiments with $m = 10, 15$, and results improved (it was also more expensive as we didn't parallelize MC training).

In the experiments on Lab2D - Figure 2, it seems MC ensemble(3) is not done learning. There is a misunderstanding of the plot here. The x-axis corresponds to the total training data size (in terms of episodes or rollouts), it does not correspond to training time or steps. MC and TD take similar training time; Adaptive TD requires m times longer.

In Figure 3, Adaptive TD(3) behaves very poorly on 2 of the 6 maps. In maps 0 and 4, in the high-data regime, MC methods outperform Adaptive TD. This could be expected, as with abundant data MC tends to be best. Maps 0 and 4 are those where the target region is the furthest from any wall (actually, for map 0 there are no walls at all). In these cases, due to lack of discontinuities and leakage, the issue we are trying to address in this paper is less severe. Still, in both cases, Adaptive TD does better than TD (except in map 4 with 100 training rollouts, where it does around 5% worse).