Reviewer 1: **Q:** Compare and highlight new challenges of analyzing TDC relative to other GTD algorithms in [30,34].

A: (a) As mentioned in [14], TDC does not have an explicit saddle point representation as GTD and GTD2, and hence its analysis cannot follow the convex-concave optimization framework developed in [30,34]. (b) [30,34] assume that two variables' updates have the same constant stepsize. For TDC, we analyze more general cases: two stepsizes have different diminishing rates, and two stepsizes are different valued constants. Consequently, in our case, interaction between two variables requires more sophisticated techniques to analyze, e.g., recursively sharpening error bounds.

Q: The paper generalizes stagewise stepsize in conventional (one timescale) optimization to two-timescale optimization. Discuss new challenges of analyzing algorithms with blockwise diminishing stepsize in this new settings.

A: Comparing to conventional optimization with stagewise stepsize, here we need to handle the bias induced by non-i.i.d. samples and characterize non-asymptotic behavior of two timescale variable update. Hence, the update scheme for stepsize and block length in each block is designed based on two time-scale analysis to yield linear convergence rate blockwisely and desirable sample complexity.

Q: How the theoretical guarantees can be affected in the non-asymptotic analysis of actor-critic and gradient Q-learning.

A: Since both actor-critic and gradient Q-learning algorithms are two time-scale algorithms, our non-asymptotic analysis for two time-scale algorithms can be very useful. Moreover, analysis of these two algorithms will further require to deal with their special structures such as policy update, presence of multiple fixed points, local convergence, etc.

Reviewer 2: **Q:** How is θ^* obtained in the experiments.

A: In our experiment (Garnet problem), since we pick behavior policy π_b and transition probability $p(s'|s,a)$, the stationary distribution μ_{π_b} can be computed. Since we also know target policy π and feature matrix Φ, we can compute the matrix A and the vector b by definition to obtain $\theta^* = -A^{-1}b$. Alternatively, θ^* can also be estimated by running the algorithm with diminishing stepsize for sufficiently long time and taking the average of outputs of several runs.

Q: How would worst-case errors predicted by the bound compare to errors observed empirically in experiments.

A: Our theory captures how the error bound changes with stepsize diminishing parameters (ν, σ), which agrees with how the empirical error changes with stepsize diminishing parameters in our experiments (see Fig. 1 in the paper). Furthermore, specializing Theorem 1 to i.i.d. scenarios, our convergence rate order-wisely matches the best known result in [Dalal et al. COLT 2018]. It is a good idea to plot theoretical errors and compare with empirical bounds. The main challenge here is that precisely estimating some parameters in the error bound (e.g., eq (25)) can be difficult (although they are known to be constants in convergence analysis). For example, mixing time parameters τ_α and τ_β in (25) depend on geometric ergodicity of Markov chain, but constants m and ρ (see Assumption 3) are usually difficult to estimate in practice. We are currently further exploring such an issue.

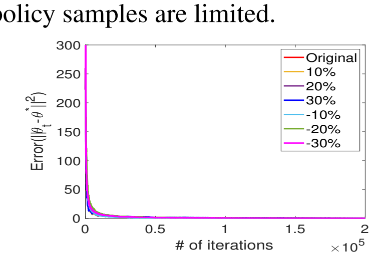
Q: Besides implications for the choice of step-size, do these bounds provide insight on what properties of the problem, the behavior policy, and the representation affect the rate of convergence?

A: Theorem 1 (more precisely eq (25) in supple.) captures how other properties (besides stepsize) affect convergence rate. For example, convergence rate depends on λ_θ, which is lower-bounded by the largest eigenvalue of matrix $2A^\top C^{-1}A$, and such a matrix is determined by behavior policy π_b, target policy π, transition probability $p(s'|s,a)$ and feature matrix Φ. Convergence rate also depends on the mixing time τ_α due to geometric ergodicity of the Markov chain, which is determined by π_b and $p(s'|s,a)$. Other constant terms in (25) such as $L_{f_1,\theta}$, K_{f_1} and K_{g_1} capture the dependence on π_b, π, Φ and the discount factor γ.

Q: Explain what "more flexible" mean when saying gradient TD are "more flexible than on-policy learning in practice."

A: We meant gradient TD algorithms are flexible because they converge even with off-policy data and hence can exploit abundant samples (obtained under behavior policies) for learning when the on-policy samples are limited.

Reviewer 3: **Q:** How to set blocksize properly without prior knowledge and how robust the algorithm is with respect to blocksize hyperparameter.



A: In practice, blocksize T_s and stepsize α_s are set by parameter tuning, but we do not directly tune them for all blocks because there are too many tuning parameters this way. Instead, Theorem 3 indicates that T_s and α_s for all blocks are fully determined by only four parameters ϵ_0, $|\lambda_x|$, C_7, and η, and among them $|\lambda_x|$ and η can be estimated by matrices A and C from samples. Hence we mainly tune only ϵ_0 and C_7. Our experiments demonstrate that this approach yields desirable performance. For robustness, we run experiments (see the figure on the right) and find that perturbing blocksize even by $\pm 30\%$ for all blocks changes the convergence rate only very slightly, demonstrating that the performance of algorithm is very robust to blocksize.