General response and additional numerical experiments:

To address the reviewers' concerns regarding the empirical performance of implicit TD algorithms, we considered two settings: (1) a synthetic 100-state Markov Reward Process (MRP) environment with positive transition probabilities, and (2) classical continuous-domain control problems (Acrobot and Mountain Car). The performance of the standard and implicit TD algorithms in the 100-state MRP environment—with 20 random binary features—is shown in Figure 1 and Table 1. For the continuous-domain control problems, radial basis features were used, and the performance metric was the root mean squared Bellman error (RMSBE), as depicted in Figure 2, Table 2, and Table 3. For the synthetic 100-state MRP problem, the step size was set to $\alpha_t = \frac{300}{t+1}$, while for the continuous-domain control problems, we considered $\alpha_t = \frac{\alpha_0}{t+1}$ with $\alpha_0 = 1$ and $\alpha_0 = 0.1$.

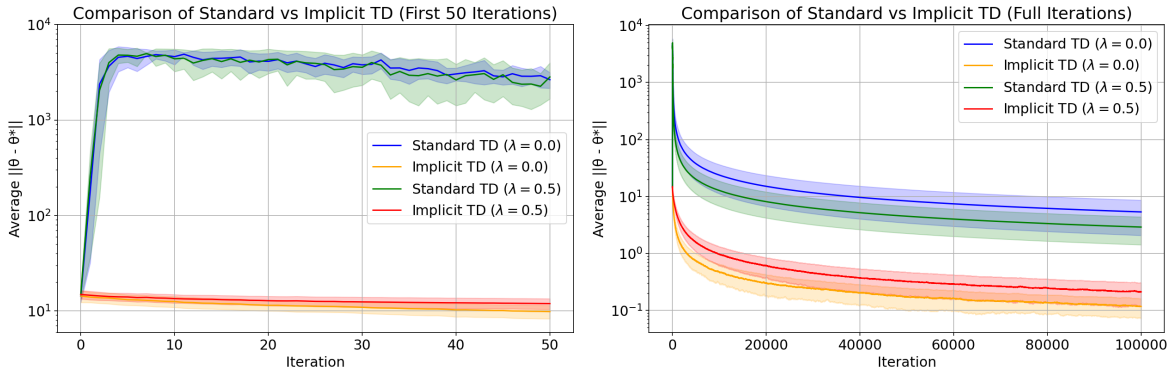

Figure 1: Parameter estimation error for synthetic MRP with 100 states (Left: 50 iterations, Right: 100000 iterations)

Method	λ	Mean	Std
Standard TD	0.0	5.355814	3.278592
Implicit TD	0.0	0.117330	0.044243
Standard TD	0.5	2.905596	1.483903
Implicit TD	0.5	0.212468	0.093600

Table 1: Final errors for 100 MRP experiments for each method and λ value.

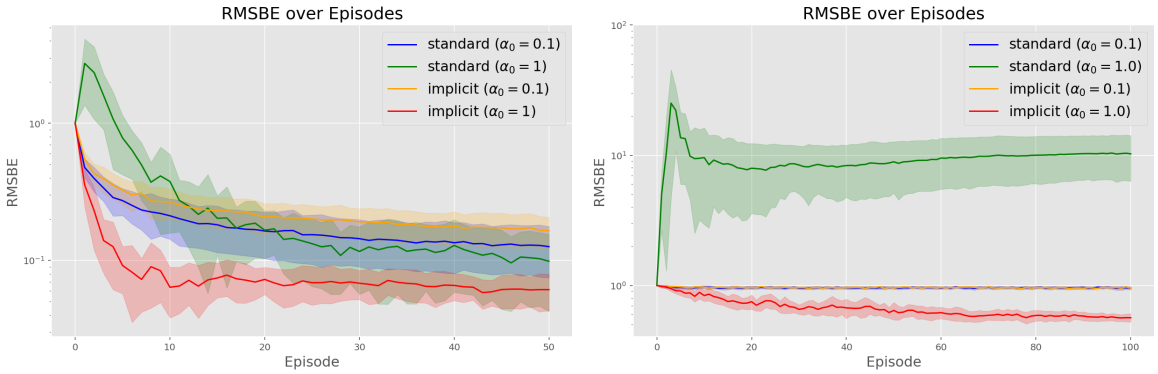

Figure 2: RMSBE for classical control problem (Left: Acrobot, Right: Mountain Car)

In addition, we have now included the variation (standard deviations) observed across the 50 independent experiments for the 11-state random walk experiments. We would like to emphasize that, in the bottom-left subfigure of Figure 3,

Method	α_0	Mean	Std
Standard TD	0.1	0.126078	0.051337
Standard TD	1.0	0.098693	0.056317
Implicit TD	0.1	0.164576	0.042195
Implicit TD	1.0	0.061291	0.018172

Table 2: Final RMSBE (Acrobot Environment) for standard/implicit TD(0) and step-size parameter α_0.

Method	α_0	Mean	Std
Standard TD	0.1	0.952269	0.026053
Standard TD	1.0	10.248247	3.938624
Implicit TD	0.1	0.951045	0.026131
Implicit TD	1.0	0.565690	0.041935

Table 3: Final RMSBE (Mountain Car Environment) for standard/implicit TD(0) and step-size parameter α_0.

the standard TD(0) iterates diverged, as evidenced by the y-axis scale reaching up to 10^{109}. A similar phenomenon is observed for projected TD(0). The numerical instability of standard TD algorithms resulted in large variability across the 50 independent experiments. In contrast, no such high variability was observed for implicit TD algorithms, demonstrating their superior numerical stability. The same phenomenon applies to the TD(1/2) results, as shown in Figure 4. While the standard TD(1/2) algorithm was numerically more stable than TD(0), the implicit version of the TD(1/2) algorithm exhibited enhanced numerical stability compared to its standard counterpart.

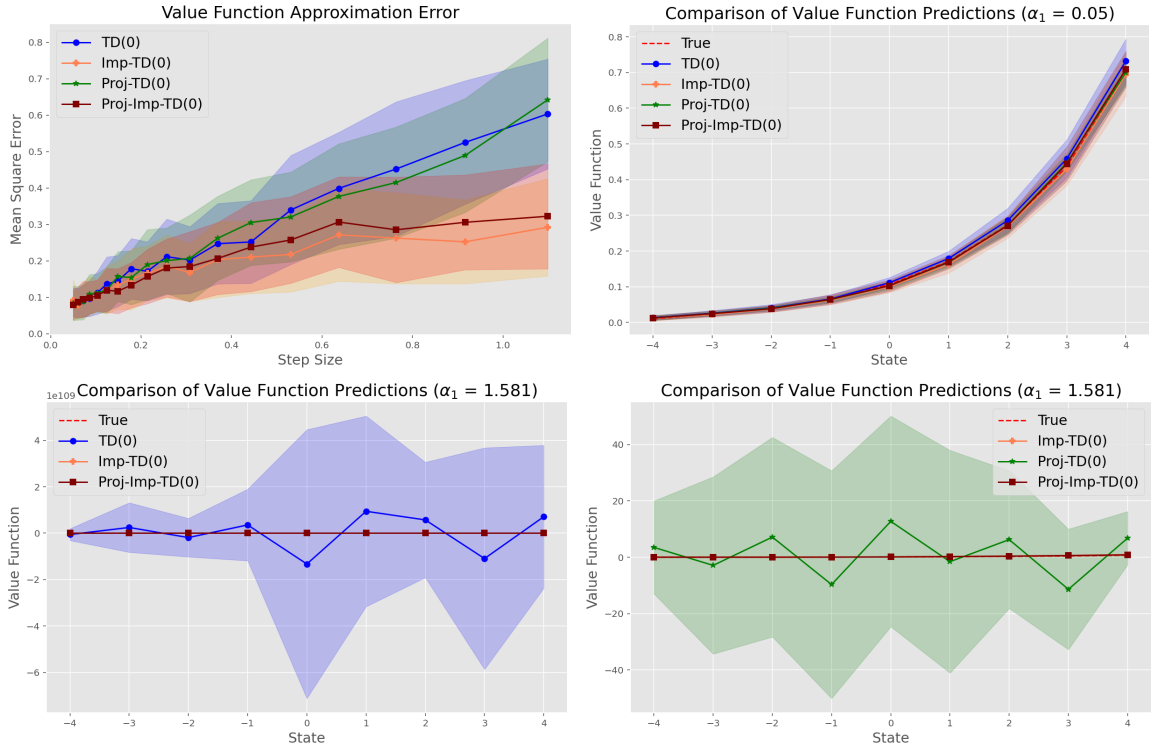

Figure 3: Value function approximation based on TD(0) over a range of constant step size values

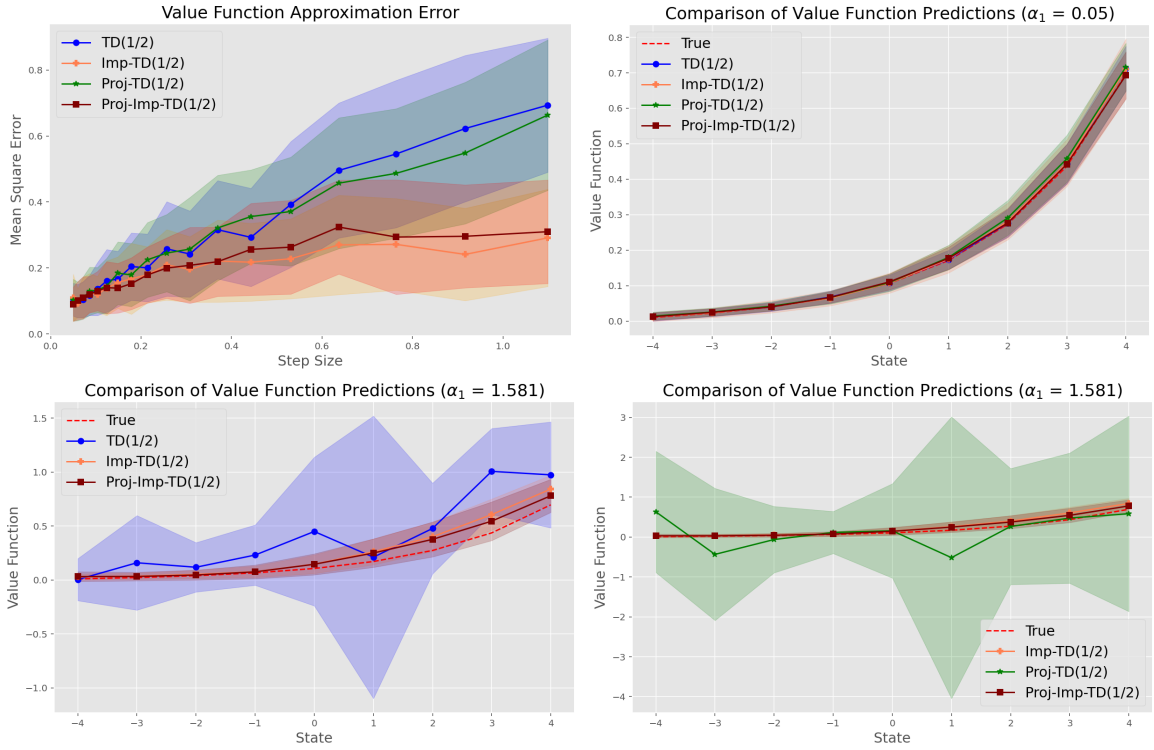

Figure 4: Value function approximation based on TD(1/2) over a range of constant step size values

Numerical experiments setting of the 100-state synthetic MRP environment

The numerical setting described below is motivated by the setting considered in [1], which considered standard TD algorithms in the average reward setting.

Step 1: MRP generation

The experiment begins by generating a Markov Reward Process (MRP) with 100 states. This process is defined by:

- **Transition probabilities:** For each state, a probability distribution over the next states is generated. Specifically, for a state s, $N-1$ random numbers are drawn uniformly from $[0,1]$, sorted, and then used to compute the differences between consecutive numbers. The first probability is set to the first random value, and the last is 1 minus the last random value.
- **Rewards:** Each state is assigned a reward $r(s)$ drawn from a uniform distribution on $[0,1]$.
- **Discount factor:** A discount factor $\gamma = 0.9$ is used, which weights future rewards.

Thus, the MRP is characterized by its state space, transition matrix P, and reward vector $\mathbf{r}$.

Step 2: True value function construction

Once the MRP is established, the true value function v is computed using the discounted Bellman equation:

$$v^* = (I - \gamma P)^{-1}\mathbf{r},$$

where I is the identity matrix, P is the transition matrix, and $\mathbf{r}$ is the reward vector.

Step 3: Feature matrix construction

For linear function approximation, a feature matrix $\Phi \in \mathbb{R}^{N\times d}$ is constructed with the following requirements:

- **Full Column Rank:** The matrix must have rank d, ensuring that the d features span a d-dimensional space.
- **Normalized Rows:** Each row $\Phi(i,:)$, corresponding to a state i, is normalized such that $\|\Phi(i,:)\| \leq 1$.
- **Inclusion of the True Value Function:** Φ is constructed by generating d random binary features and concatenating them in column-wise.

Step 3: Computing optimal parameters

Next, the optimal parameter vector θ^* is determined as the least-squares solution to the equation

$$\Phi\,\theta \approx v^*,$$

i.e.,

$$\theta^* = \arg\min_{\theta} \|\Phi\,\theta - v^*\|_2.$$

This vector θ^* serves as the benchmark against which the performance of the TD algorithms is measured.

References:

[1] Zhang, Sheng, Zhe Zhang, and Siva Theja Maguluri. "Finite Sample Analysis of Average-Reward TD Learning and Q-Learning." Advances in Neural Information Processing Systems 34 (2021): 1230-1242.

Reviewer Faqp

Comments of Experimental Designs or Analyses

- All figures (Figure 1-3) should show the variance across 50 runs in addition to their mean values.

  Response: Thank you for the helpful suggestion. We have now included the standard deviation along with the mean squared error to reflect the variation across independent experiments. The larger the shaded region, the greater the variation in algorithm outputs across the 50 independent experiments. Updated plots in Section 5 are provided in both Figure 3 and Figure 4.

- The test domain only has 11 states, if the proposed method can converge at a certain state, I don't see the reason why the discount factor set to be 0.9. What's the performance when the discount factor set to be 1.0?

  Response: The primary purpose of using a discount factor of 0.9 was to align with the condition of theoretical results in the manuscript. Per request, we now provide numerical experiment results using the same experimental setting as in Section 5 of the submitted manuscript, but with a discount factor of $\gamma = 1$. The results are highly similar to those for $\gamma = 0.9$, demonstrating the sensitivity of standard TD algorithms to the choice of the step size sequence. These numerical results are provided in both Figure 5 and Figure 6.

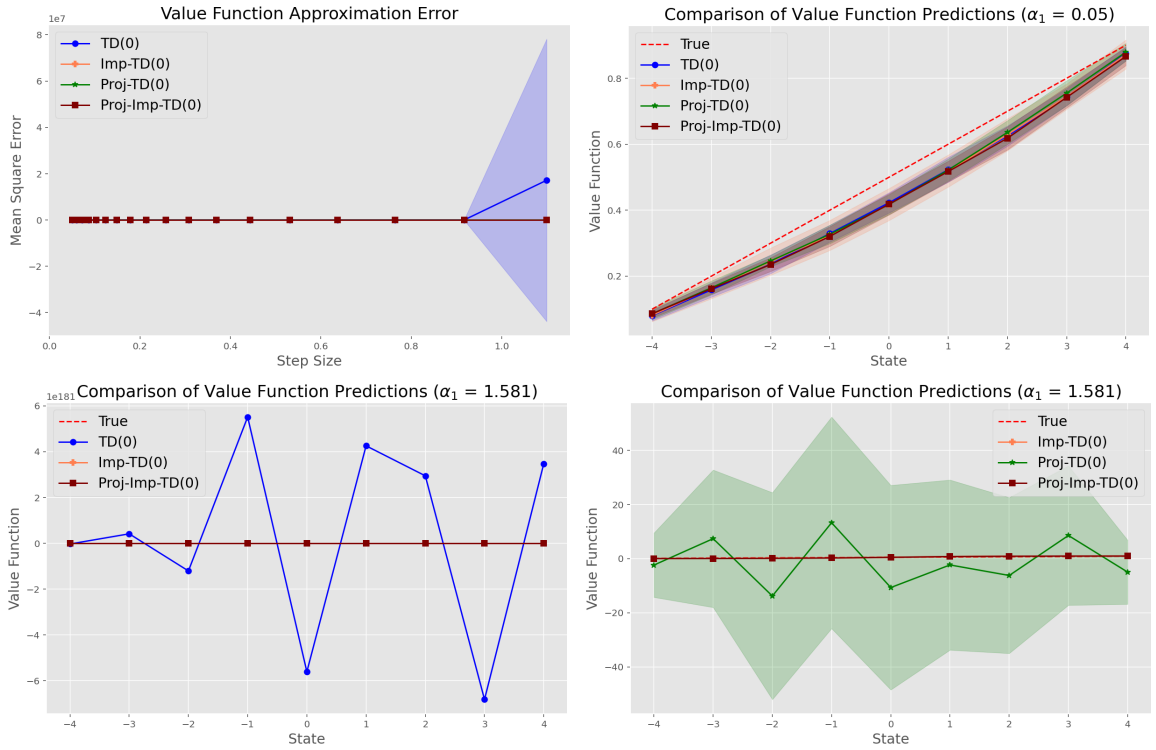

Figure 5: Value function approximation based on TD(0) over a range of constant step size values with a discount factor $\gamma = 1$

- Comparing TD and Implicit TD is insufficient since there are many RL algorithms proposed for the stability problem w.r.t. policy evaluation, such as Gradient TD, Emphatic TD, Proximal Gradient TD, Retrace(λ), V-trace, etc.

  Response: While we will cite the excellent papers mentioned by the reviewer, we would like to point out that all these methods are geared towards off-policy settings. No method is designed to solve the stability issues arising from stepsize choice in on-policy settings, like our method does. As a result, these methods do not provide a natural benchmark for our paper. As before, we would appreciate a clarification in case we misread your

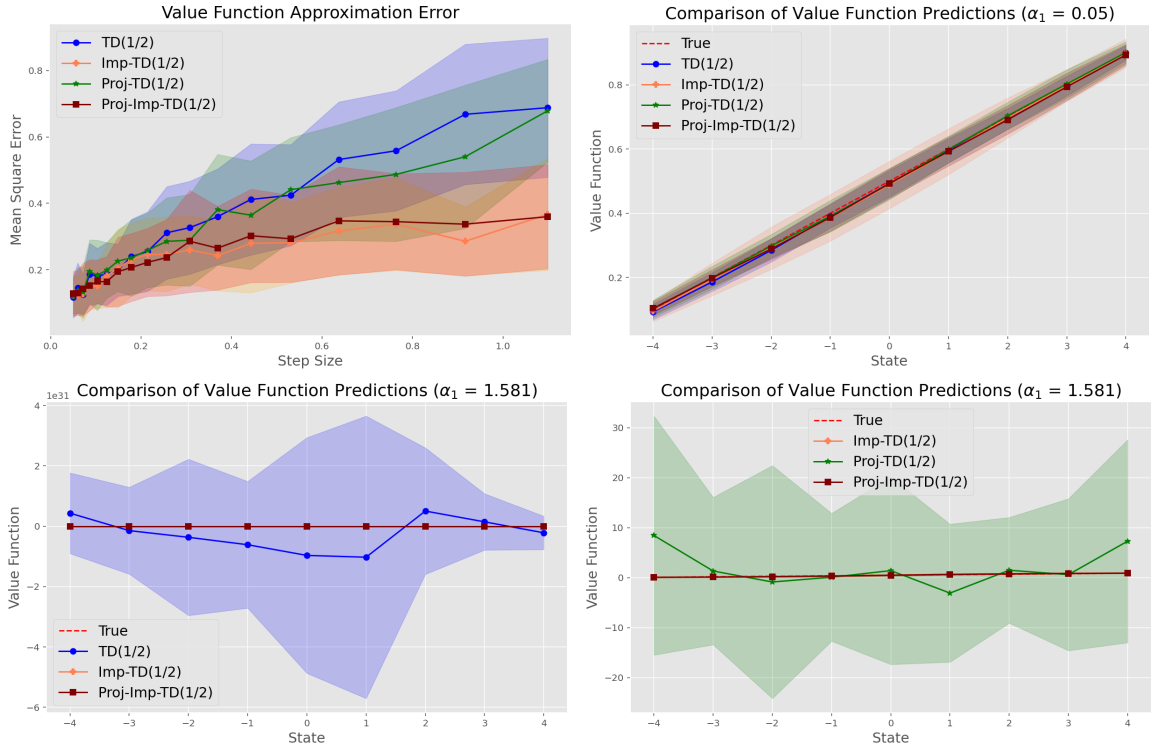

Figure 6: Value function approximation based on TD(1/2) over a range of constant step size values with a discount factor $\gamma = 1$

comment. As an illustration, we followed your suggestions and employed Gradient TD in the 11-state MDP setting. We observed significant algorithm divergence when used with large step size (notice that y-scale axis scale goes up to 10^{18} for step size larger than 1.0), which can be seen in left sub-figure of Figure 7. Even in the case, where it did not diverge, from the right sub-figure of Figure 7, the inferior performance of GTD2 algorithm was observed in comparison to standard TD and implicit TD algorithms for on-policy evaluation task.

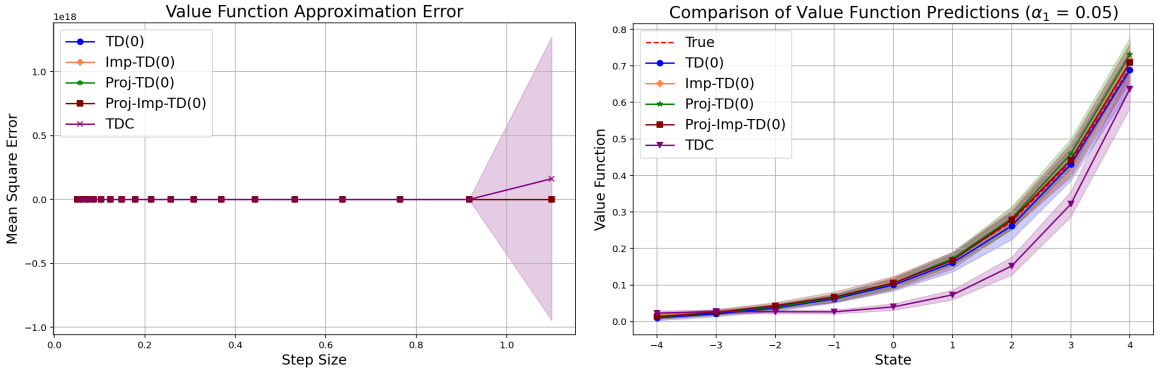

Figure 7: Comparison with GTD2

Our goal in the submitted manuscript is to resolve the stability issue in standard TD algorithms for on-policy evaluation with respect to the choice of step size, and we will highlight this in the revised manuscript. The topic of off-policy evaluation is beyond the scope of the current work, and we believe a valid comparison is to consider methodologies within the on-policy evaluation category.

Many existing off-policy evaluation methods, as noted by the reviewers, also suffer from sensitivity to the choice of step size. In particular, off-policy evaluation algorithms that fall into the category of two-time-scale stochastic

approximation algorithms are highly sensitive to the choice of step sizes, a fact that has been documented in numerous studies [1,2,4]. Therefore, we believe that an interesting future direction would be to develop implicit algorithms that relax the conditions on the step size sequence for off-policy evaluation algorithms. For now, however, we consider this is beyond the scope of the current manuscript.

- How does this work measure the computational efficiency?

  Response: The computational efficiency we refer to combines the following aspects. First, regarding run-time and memory complexities, the proposed implicit TD algorithms share the same run-time and memory requirements as the standard TD algorithms. Second, regarding the number of iterations required to reach the optimal parameter, our results for the 100-state MRP example (Figure 8 and Table 4) and the mountain car control problem (Figure 9, Table 5 and Table 6) show that using a decreasing step-size sequence with a large initial step size enables convergence to the optimal parameter much faster than the standard TD algorithm, requiring far fewer iterations.

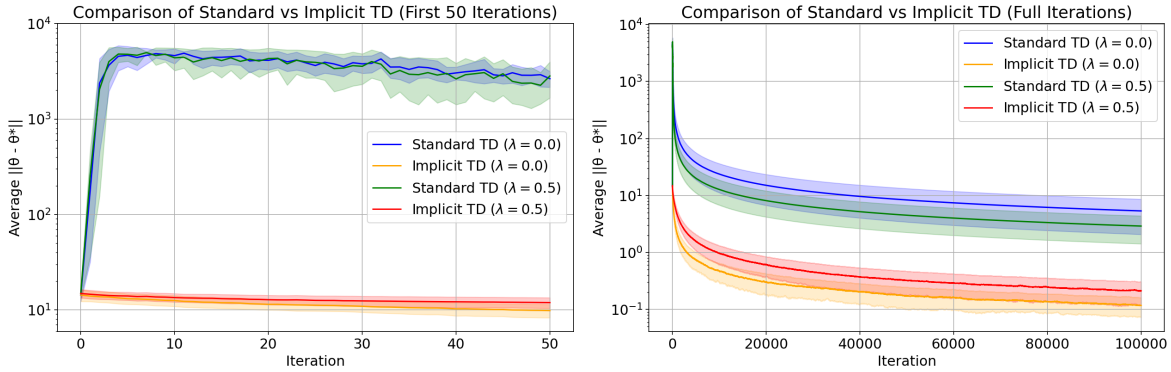

Figure 8: Parameter estimation error for synthetic MRP with 100 states (Left: 50 iterations, Right: 100000 iterations)

Method	λ	**Mean**	**Std**
Standard TD	0.0	5.355814	3.278592
Implicit TD	0.0	0.117330	0.044243
Standard TD	0.5	2.905596	1.483903
Implicit TD	0.5	0.212468	0.093600

Table 4: Final errors for 100 MRP experiments for each method and λ value.

Essential References Not Discussed

- Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., & Petrik, M. (2015). Finite-sample analysis of proximal gradient TD algorithms. In Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence (pp. 504-513).
- Zhang S, Whiteson S. Truncated emphatic temporal difference methods for prediction and control[J]. Journal of Machine Learning Research, 2022, 23(153): 1-59.
- Sutton R S, Mahmood A R, White M. An emphatic approach to the problem of off-policy temporal-difference learning[J]. Journal of Machine Learning Research, 2016, 17(73): 1-29.
- Touati A, Bacon P L, Precup D, et al. Convergent tree backup and retrace with function approximation[C]. International Conference on Machine Learning. PMLR, 2018: 4955-4964.

Response: Thanks for providing references on off-policy evaluations. We will cite these references, discuss off-policy evaluation problems, and emphasize that the goal of the manuscript is to develop on-policy evaluation algorithms that

are robust to the choice of step size.

Other Strengths And Weaknesses The stepsize could be one of the factors w.r.t. the convergence. It's not clear to me why this paper considers TD with the eligibility trace. Usually, the stability issue could be related to the off-policy setting, function approximation, and eligibility trace, particularly when these three are interwined together [Touati, et al.'2018]. I can understand the proposed algorithm could leverage the stochastic optimization method to stabilize the TD update term, but I don't see the real motivation behind the work. Besides, as I mentioned before, the empirical results are too weak. Why this test domain is selected? What's the performance on the off-policy setting and on-policy setting, respectively? What's the comparison with other methods, such as Gradient TD, Emphatic TD, Proximal Gradient TD, Retrace(λ), V-trace, etc.

Response: Thank you for the suggestions. Indeed, the stepsize is one factor that affects stability among others, and stability issues arise in many other tasks than those we consider in our paper. However, the focus of this paper is on stability issues related specifically to stepsize choice under the on-policy learning setting. Unlike our paper, all of the papers cited by reviewers focus on off-policy learning. Implicit ideas could be used in the off-policy setting as well, but this setting is not the focus of the current paper.

It seems that the reviewer is suggesting that we evaluate our algorithm in off-policy tasks. We admit we are confused by this comment because our algorithm is developed specifically for on-policy learning tasks. We would appreciate a clarification on this point. Specifically, which off-policy settings do you consider appropriate for our algorithm?

Regarding the experiments, we selected this domain because it is simple and illustrates the stability issues of TD. Please note that many other papers, follow a similar approach. For instance, the Gradient TD paper [5] demonstrated its effectiveness in 5-state random walk and 14-state Boyan chain example and [4] provided 2-state counter example to illustrate the difficulty of off-policy evaluation. In a similar spirit, the goal is not to show that the proposed method is better than any other conceivable solution, but only to demonstrate the issues in the widely used baseline algorithm. From Figures 1 and 2 in the submitted manuscript, the numerical instability (with respect to the choice of step size) of both standard TD(0) and TD(λ) algorithms is evident. The central motivation behind our work is to develop temporal difference learning algorithms that are more robust to the choice of step size sequence. Such robustness can lead to improved performance in value function approximation/policy evaluation problems, as demonstrated in experiments on an 11-state random walk, a 100-state synthetic MRP, and two continuous-domain classic control problems (including the Acrobot and Mountain Car environments). Our intention was to illustrate the numerical instability of standard TD algorithms even in simple environments like the 11-state random walk we considered. Emphasizing again, the focus of the current manuscript is on stability with respect to the choice of step size, not on the stability issues caused by off-policy samples.

Other Comments Or Suggestions:

- Add other test domains which are representative and challenging.

  Response: To address your concern regarding the numerical experiments section, we have conducted additional experiments on a 100-state synthetic MRP using 20 random binary features. The performance of the standard TD and implicit TD algorithms is shown in Figure 8 and Table 4. To further demonstrate the effectiveness of implicit TD algorithms, we have also considered classical continuous-domain control problems, including the Acrobot and Mountain Car environments, using radial basis features. For these control problems, the performance of the standard TD and implicit TD algorithms was measured in terms of the root mean squared Bellman error (RMSBE), as shown in Figure 9, Table 5 and 6. Regarding the step-size choice, we considered $\alpha_t = \frac{\alpha_0}{t+1}$ for both $\alpha_0 = 1$ and $\alpha = 0.1$. For a small initial step size ($\alpha_0 = 0.1$), neither the standard nor the implicit TD algorithms exhibited numerical instability. However, for larger initial step size ($\alpha_0 = 1$), as shown in Figure 9, the numerical instability of the standard TD algorithms drastically worsened their performance for the mountain car environment.

- Compare with other methods as mentioned.

  Response: Please see the response to the comments on the experimental designs or analyses.

- Need sensitivity test for hyper-parameters.

  Response: We would appreciate if the reviewer could clarify the type of hyper-parameters he/she is referring to.

- This cite is problematic: Sutton, R. S. Reinforcement learning: An introduction. A Bradford Book, 2018.

  Response: Thanks for the catch. We will fix the citation immediately.

Questions For Authors: What are the noisy or ill-conditioned settings mentioned in the paper? Could you show the cases in the experiments?

Response: In the 11-state random walk example, using a constant step size of $\alpha = 1.581$, we numerically demonstrated the explosive behavior of the standard TD algorithms, as shown in Figure 1. Furthermore, in continuous-domain control problems (the Acrobot and Mountain Car environments), the standard TD algorithms again suffer from numerical instability when a large initial step size is used (see Figure 9, Table 5 and 6). In particular, in the Mountain Car environment, we observe that the RMSBE of the standard TD algorithm does not decrease when a large initial step size is employed. This phenomenon illustrates an ill-conditioned setting where the numerical instability caused by a large initial step size gets worsened by the underlying structure of the MRP.

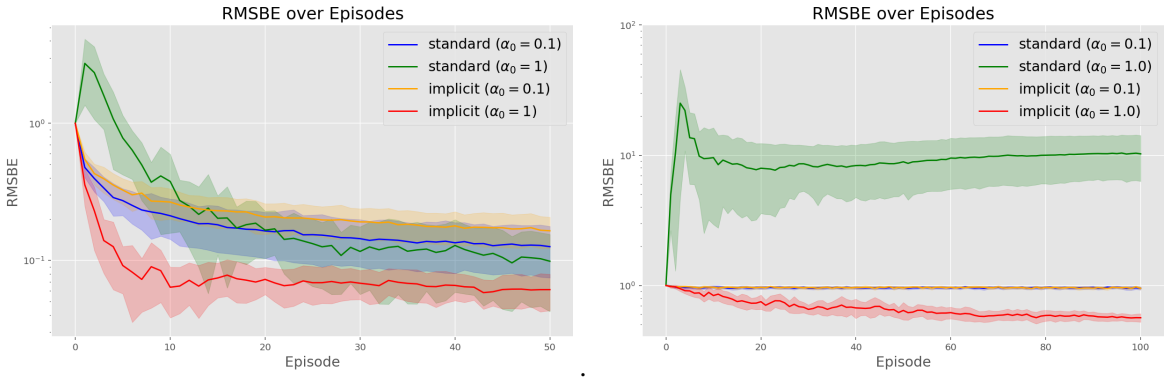

Figure 9: RMSBE plots for acrobot (left) and mountain car (right)

Method	α_0	**Mean**	**Std**
Standard TD	0.1	0.126078	0.051337
Standard TD	1.0	0.098693	0.056317
Implicit TD	0.1	0.164576	0.042195
Implicit TD	1.0	0.061291	0.018172

Table 5: Final RMSBE (Acrobot Environment) for standard/implicit TD(0) and step-size parameter α_0.

Method	α_0	**Mean**	**Std**
Standard TD	0.1	0.952269	0.026053
Standard TD	1.0	10.248247	3.938624
Implicit TD	0.1	0.951045	0.026131
Implicit TD	1.0	0.565690	0.041935

Table 6: Final RMSBE (Mountain Car Environment) for standard/implicit TD(0) and step-size parameter α_0.

How to choose the range for constant step sizes? How to set the projection radius R?

Response: Based on our theoretical analysis, for the projected implicit TD algorithms the finite time error bounds hold for any positive constant step size as long as the discount factor $\gamma \geq 0.5$. We believe this is a significant improvement

over existing standard TD algorithms and their variants. Empirically, we have observed that implicit TD algorithms demonstrate superior numerical stability compared to standard TD algorithms in all of our experiments (11-state random walk, 100-state MRP, Acrobot, and Mountain Car environments). Furthermore, the performance of implicit TD algorithms is robust to the choice of projection radius R, which was seen in Figure 3 of the submitted manuscript as well as the 100-state MRP example, where we used $R = 500$. Therefore, we suggest choosing a sufficiently large radius R that contains the optimal parameter value. Note that R can be relatively loose, which is consistent with common practices in [3,4].

References:

[1] Gupta, Harsh, Rayadurgam Srikant, and Lei Ying. "Finite-time performance bounds and adaptive learning rate selection for two time-scale reinforcement learning." Advances in neural information processing systems 32 (2019).

[2] Haque, Shaan Ul, Sajad Khodadadian, and Siva Theja Maguluri. "Tight finite time bounds of two-time-scale linear stochastic approximation with markovian noise." arXiv preprint arXiv:2401.00364 (2023).

[3] Bhandari, Jalaj, Daniel Russo, and Raghav Singal. "A finite time analysis of temporal difference learning with linear function approximation." Conference on learning theory. PMLR, 2018.

[4] Xu, Tengyu, Shaofeng Zou, and Yingbin Liang. "Two time-scale off-policy TD learning: Non-asymptotic analysis over Markovian samples." Advances in neural information processing systems 32 (2019).

[5] Sutton, Richard S., et al. "Fast gradient-descent methods for temporal-difference learning with linear function approximation." Proceedings of the 26th annual international conference on machine learning. 2009.

Reviewer mtYA

Experimental Designs Or Analyses:

- The problem set up seems to simulate a Markov chain which is not irreducible (presence of absorbing states). I think a better simulation set would include setups that do indeed satisfy the assumption. And since the paper is geared towards larger state space sizes, performing these experiments for a larger cardinality would be more informative (the current state space cardinality is 11).

  Response: Thank you for the suggestion. We considered an MRP with 100 states and positive transition probabilities, following the construction in [1]. The total number of random binary features used in the experiment was 20, which is smaller than the state dimension of 100. For this problem, we used the decreasing step size sequence $\alpha_t = \frac{300}{t+1}$. The figures below show the average performance of the standard and implicit TD algorithms, along with their standard deviations, across 20 independent experiments. As seen in the left subfigure of Figure 10, which focuses on the behavior of the TD iterates during the first 50 iterations, the standard TD algorithms suffer from numerical instability when used with large initial step sizes, causing the TD iterates to deviate from w^*. This, in turn, leads to a longer number of iterations required to reach the pre-specified accuracy level, as shown in the right subfigure of Figure 10, which presents the performance over a total of 100,000 iterations. See also Table 10.

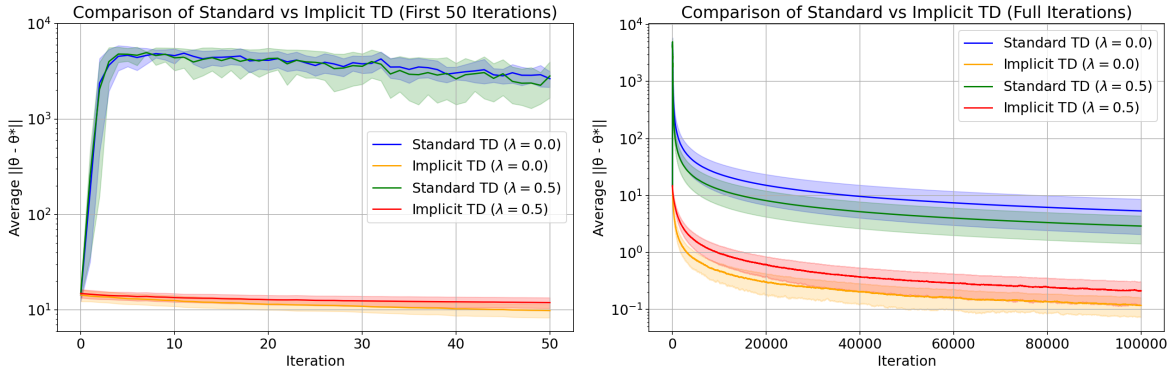

Figure 10: Parameter estimation error for synthetic MRP with 100 states (Left: 50 iterations, Right: 100000 iterations)

Method	λ	Mean	Std
Standard TD	0.0	5.355814	3.278592
Implicit TD	0.0	0.117330	0.044243
Standard TD	0.5	2.905596	1.483903
Implicit TD	0.5	0.212468	0.093600

Table 7: Final errors for 100 MRP experiments for each method and λ value.

- Another question I am curious about is the variation of the random step sizes with time. Just out of curiosity, I was wondering if the authors can plot vs as a function of the iteration number (even if is a fixed quantity).

  Response: The plot of decreasing step size $\alpha_t = \frac{300}{t+1}$ versus effective step sizes for implicit TD(0): $\frac{\alpha_t}{1+\alpha_t\|\phi_t\|^2}$ and implicit TD(λ): $\frac{\alpha_t}{1+\alpha_t\|e_t\|^2}$ is provided in Figure 11. The underlying problem we considered was the synthetic 100-state MRP problem. As one can see from Figure 11, all three step size schedules decrease to zero, which follows from our Lemma A.16. In the meantime, the effective step sizes for the implicit algorithms (both $\frac{\alpha_t}{1+\alpha_t\|\phi_t\|^2}$ and $\frac{\alpha_t}{1+\alpha_t\|e_t\|^2}$) are not necessarily monotonic, as they depend on the random quantity ϕ_t and e_t. Such an adaptive step size prevents numerical instability as it appropriately scales down drastic temporal difference updates.

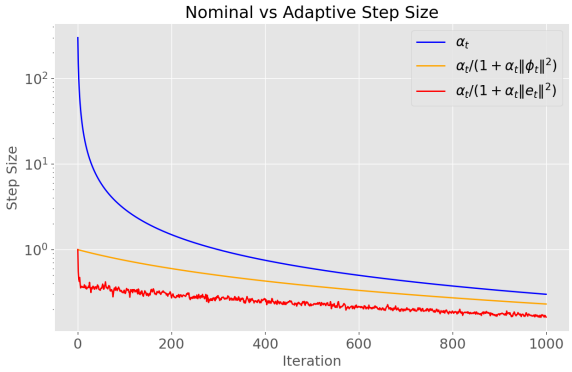

Figure 11: Chosen step size and effective step size

- I am wondering what the performance would look like if a fixed step size is used for a while, and then you shift to decreasing step sizes. Or just use the decreasing step size proposed in the paper?

  Response: Thanks for raising this question. As the strength of the implicit algorithms is in their robustness in the choice of step-size, one can use a constant step size or a decreasing sequence of step sizes with a large initial step size. In the aforementioned synthetic MRP experiment with 100 states, we have used the step size of $\alpha_t = \frac{300}{t+1}$. With the standard TD algorithm, due to a large initial step size, the algorithm at first deviates from the w^* and slowly gets closer to it. Unlike the standard TD, implicit TD algorithms remain numerically stable for large initial step-sizes, which allows us to get closer to w^* quickly.

- The bottom two sub-figures in each of the figures are unclear to me. Why is the true value function changing across them? Also, at what time step is each iteration of the algorithm stopped? I understand the number of runs of the algorithm is 50 but I do not know when each iteration ends.

  Response: Sorry for causing confusion. The true value function in the bottom two sub-figures is the same across all three figures. Due to the numerical instability of the standard TD algorithms, the y-axis scale of the plots differs between the sub-figures. Note that in Figure 1 of the submitted manuscript, for standard TD(0) with a constant step size $\alpha_1 = 1.581$, the approximated value function diverges significantly, as evidenced by the y-axis scale reaching up to 10^{110}. Since the Markov chain has an absorbing state, instead of specifying the total number of iterations, a total of 200 episodes was used.

- The plots require error bars across 50 experiments.

  Response: Thank you for the helpful suggestion. We have included the standard deviation along with the mean squared error to reflect the variation across independent experiments. The larger the shaded region, the greater the variation in algorithm outputs across the 50 independent experiments. Updated plots in Section 5 are provided in both Figure 3 and Figure 4.

Relation To Broader Scientific Literature: Other Strengths And Weaknesses:

- Strength: I think the paper addresses an important question to the community. It is true that using a decreasing step size essentially alleviates a lot of the issues raised in the paper. However, it does take longer to converge. Identifying a fixed step size sequence which approximately converges without diverging, where the step size can be determined in a problem independent fashion is indeed an important question that the paper seems to have answered.

  Response: Thank you for recognizing the significance of the questions addressed in this manuscript. With the additional numerical experiments provided in this rebuttal, we hope to further demonstrate both the importance of the research question and the effectiveness of the proposed implicit TD algorithms.

- Weakness: The paper needs to quite significantly improve on their numerical experiments section. (as indicated above) The projection step seems to be necessary for their analysis to work, which has been proven to be

unnecessary in the TD learning literature. However, the authors have addressed this in the paper and noted that this phenomenon is repeated in the experiments as well. It is perhaps a future work direction.

Response: To address your concern regarding the numerical experiments section, we have conducted additional experiments on a 100-state synthetic MRP with a total number of 20 random binary features. The performance of the standard TD and implicit TD algorithms is provided in Figure 10 and Table 7. To demonstrate the effectiveness of implicit TD algorithms numerically, we have also considered classical continuous domain control problems, including the Acrobot and Mountain Car environments, with radial basis features. The performance of the standard TD and implicit TD algorithms was measured in terms of the root mean squared Bellman error (RMSBE) for control problems, as shown in Figure 12, Table 8 and Table 9. Regarding the step-size choice, we considered $\alpha_t = \frac{\alpha_0}{t+1}$ for both $\alpha_0 = 1$ and $\alpha = 0.1$. For a small initial step size ($\alpha_0 = 0.1$), neither standard nor implicit TD algorithms exhibit numerical instability. However, for large initial step sizes ($\alpha_0 = 1$), as one can see from plots in Figure 12, the numerical instability of the standard TD algorithms drastically worsens their performance.

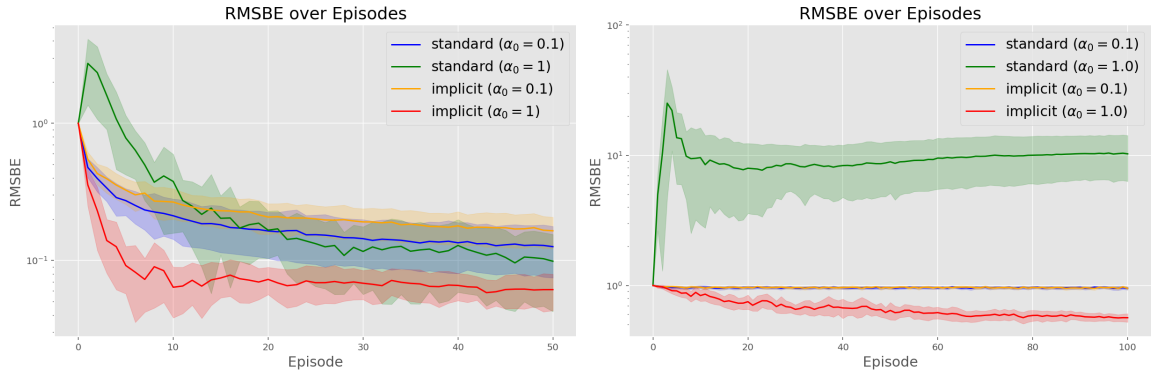

Figure 12: RMSBE plots for acrobot (left) and mountain car (right)

Method	α_0	**Mean**	**Std**
Standard TD	0.1	0.126078	0.051337
Standard TD	1.0	0.098693	0.056317
Implicit TD	0.1	0.164576	0.042195
Implicit TD	1.0	0.061291	0.018172

Table 8: Final RMSBE (Acrobot Environment) for standard/implicit TD(0) and step-size parameter α_0.

Method	α_0	**Mean**	**Std**
Standard TD	0.1	0.952269	0.026053
Standard TD	1.0	10.248247	3.938624
Implicit TD	0.1	0.951045	0.026131
Implicit TD	1.0	0.565690	0.041935

Table 9: Final RMSBE (Mountain Car Environment) for standard/implicit TD(0) and step-size parameter α_0.

Questions For Authors: I am wondering what the performance would be like when employing polyak-rupert averaging with the decreasing step sizes in the implicit TD learning algorithm? Is there a difference in the rate at which it converges (finite time bounds) or in the variance of the final iterates? Any intuition would be nice.

Response: Since our convergence analysis (with a decreasing step size) is not necessarily tight, it might be possible to achieve an improved rate (by a logarithmic factor) when combined with Polyak-Ruppert averaging. Meanwhile,

under the i.i.d. observation setting, it is known that both the Polyak-Ruppert averaged standard SGD estimator and the Polyak-Ruppert averaged implicit SGD estimator yield the asymptotically optimal variance [2]. In other words, implicit procedures retain the statistical optimality of Polyak-Ruppert averaging while demonstrating superior numerical stability. We suspect that such results can be generalized to the Markovian observation setting.

References:

[1] Zhang, Sheng, Zhe Zhang, and Siva Theja Maguluri. Finite Sample Analysis of Average-Reward TD Learning and Q-Learning. Advances in Neural Information Processing Systems. (2021): 1230-1242.

[2] Toulis, Panos, and Edoardo M. Airoldi. Asymptotic and finite-sample properties of estimators based on stochastic gradients. Annals of Statistics. (2017): 1694-1727.

Reviewer 1oni

Experimental Designs Or Analyses: The experiments are well-designed.

Response: Thank you for appreciating our experimental setting formulation!

Other Strengths And Weaknesses:

- Strengths: 1) Addresses a fundamental problem in TD learning (step-size sensitivity). 2) Theoretical guarantees (asymptotic convergence, finite-time bounds). 3) Computationally efficient (no significant increase in complexity). 4) Empirical results show clear improvements in stability.

  Response: Thank you for nicely summarizing our contributions and for appreciating the importance of the questions addressed in this manuscript.

- Weaknesses: 1) Limited empirical evaluation

  Response: To demonstrate the effectiveness of implicit temporal difference algorithms, we have included their performance on a synthetic 100-state MRP with a total of 20 random binary features, as well as on classical control problems, including the Acrobot and Mountain Car environments, where radial basis features were used. Regarding the step-size choice, for the synthetic 100-state MRP, a decreasing step size sequence $\alpha_t = \frac{300}{t+1}$ was used and for the control problems, $\alpha_t = \frac{\alpha_0}{t+1}$ was considered for both $\alpha_0 = 1$ and $\alpha = 0.1$. Results are provided in both Figure 13, Figure 14, Table 10, Table 11 and Table 12. From these results, we can observe the numerical instability of standard TD algorithms when used with a large initial step size (300 in the case of the MRP example and 1 in the case of the control problems). In contrast, the numerically stable implicit TD algorithms can be used with a large initial step size, and their performance becomes more efficient.

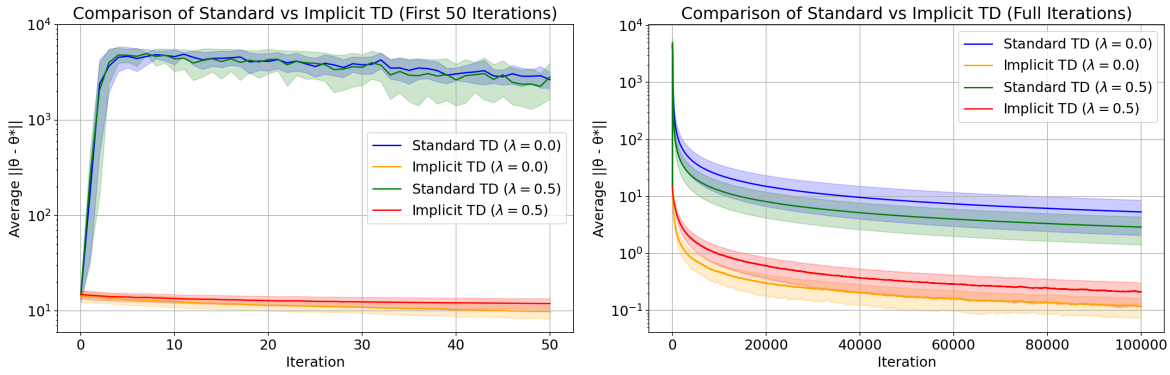

Figure 13: Parameter estimation error for synthetic MRP with 100 states (Left: 50 iterations, Right: 100000 iterations)

Method	λ	Mean	Std
Standard TD	0.0	5.355814	3.278592
Implicit TD	0.0	0.117330	0.044243
Standard TD	0.5	2.905596	1.483903
Implicit TD	0.5	0.212468	0.093600

Table 10: Final errors for 100 MRP experiments for each method and λ value.

Other Comments Or Suggestions:

- In Remark 4.18, it would be helpful to provide a bound on α and compare it with the standard bound on α.

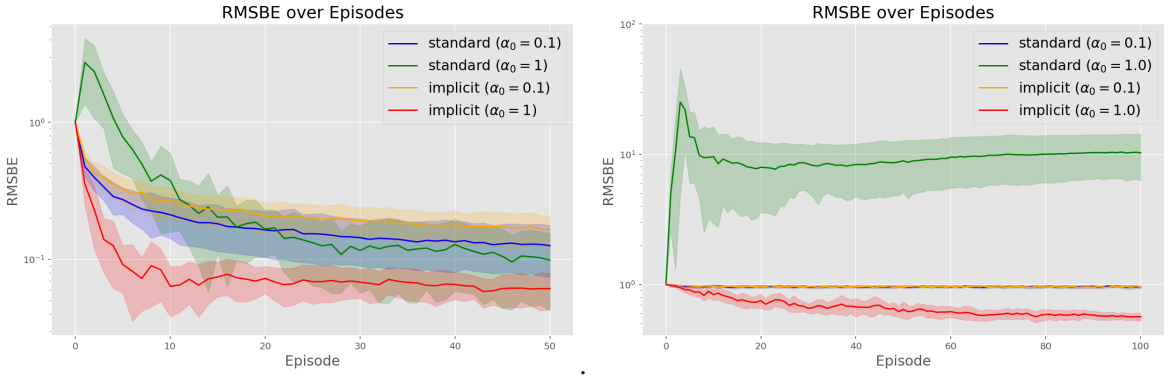

Figure 14: RMSBE plots for acrobot (left) and mountain car (right)

Method	α_0	**Mean**	**Std**
Standard TD	0.1	0.126078	0.051337
Standard TD	1.0	0.098693	0.056317
Implicit TD	0.1	0.164576	0.042195
Implicit TD	1.0	0.061291	0.018172

Table 11: Final RMSBE (Acrobot Environment) for standard/implicit TD(0) and step-size parameter α_0.

Method	α_0	**Mean**	**Std**
Standard TD	0.1	0.952269	0.026053
Standard TD	1.0	10.248247	3.938624
Implicit TD	0.1	0.951045	0.026131
Implicit TD	1.0	0.565690	0.041935

Table 12: Final RMSBE (Mountain Car Environment) for standard/implicit TD(0) and step-size parameter α_0.

Response: Thank you for the suggestion. We will provide a one-to-one comparison between the new bound and the existing one.

- The expressions in each statement should be clearer. For instance, in Theorem 4.20, does the statement hold for all $\alpha_1 > 0$ or only for some $\alpha_1 > 0$?

  Response: Thank you for the suggestion. We will clarify the expression in each theoretical statement. And regarding your question, yes, the statement holds for any $\alpha_1 > 0$.

- Additionally, what does the assumption $R \geq |w_*|$ imply? Is it an assumption or is it satisfied always? Clarifying this assumption would improve understanding.

  Response: Yes, to use the projection step, the radius R must be large enough to ensure that the optimal parameter w_* lies within the ball of radius R. The intuition is that once the projection step is included, the TD iterates are confined to the ball of radius R. If the true w_* is outside of this ball, there is no hope of converging to the optimal parameter. In practice, a standard choice is to use a sufficiently large radius R to ensure that w_* is contained within the ball [1, 2]. In Figure 3 of the submitted manuscript, we conducted experiments to examine the effect of the choice of radius and demonstrated the robustness of the implicit TD algorithms' performance to this parameter. In our additional 100-state MRP example, we used a vacuously large radius $R = 5000$, just to prevent divergent behavior of the standard TD(0) algorithm. Without such a projection step, the standard TD(0) algorithm would exhibit explosive behavior, which can be seen in the bottom subfigures of Figure 3. Unlike standard TD(0) algorithms, our extensive numerical experiments indicate that implicit TD algorithms tend to remain numerically stable even without the projection step.

- The experiments were conducted in very small-scale environments, and the results may differ in larger environments. Conducting more experiments would better support the superiority of the proposed method.

  Response: In response to the request for larger-scale experiments, we considered both 100-state MRP problems with 20 random binary features and classical continuous-domain control problems, including the Acrobot and Mountain Car environments. In all of our experiments, as shown in Figure 13, Figure 14, Table 10, Table 11 and Table 12, the numerical stability of implicit TD algorithms coupled with the use of large initial step size, which facilitated faster search space exploration and yielded a computationally efficient parameter estimation strategy.

Questions For Authors:

- In Section 3, the authors state, "both of whose norms are less than or equal to one, providing insight into why implicit TD algorithms are stable." However, it is unclear how having norms less than or equal to one provides such insight. Could the authors clarify this connection?

  Response: The notion of matrix norm $\|A\|$ can be thought as the maximum length by which the matrix A stretches a unit-length vector (i.e., for any vector x with $\|x\| = 1$). Therefore, when a matrix with norm less than or equal to 1 is applied to a vector, the length of the vector will likely decrease. In our context, this means that implicit TD algorithms sequentially apply the matrix $(I - \alpha_t \phi_t \phi_t^T)^{-1}$ to

  $$w_t^{\text{im}} + \alpha_t (r_t + \gamma \phi_{t+1}^\top w_t^{\text{im}}) \phi_t$$

  or the matrix $(I - \alpha_t e_t e_t^T)^{-1}$ to

  $$w_t^{\text{im}} + \alpha_t (r_t + \gamma \phi_{t+1}^\top w_t^{\text{im}} + \lambda \gamma e_{t-1}^T w_t^{\text{im}}) e_t,$$

  ensuring that the TD updates are not made too drastically.

- For the implicit version of TD algorithms, is the only difference the use of an adaptive step size?

  Response: While the motivation behind implicit algorithms arises from utilizing hypothetical future information to establish a more informative TD recursion, operationally, you are correct that the only difference is the use of a data-adaptive step size. Compared to standard TD algorithms, implicit TD algorithms employ step sizes that scale inversely with the magnitude of the features or eligibility traces.

- Could you provide an intuitive explanation of the derived adaptive step size? In particular, how this adaptive step size can improve the robustness on the step size?

  Response: Recall that standard TD(0) and TD(λ) update rules are given by

  $$w_{t+1} = w_t + \alpha_t \delta_t \phi_t,$$
  $$w_{t+1} = w_t + \alpha_t \delta_t e_t,$$

  where $\delta_t = r_t + \gamma \phi_{t+1}^T w_t - \phi_t^T w_t$ is the TD error and $e_t = \sum_{k=0}^t (\gamma\lambda)^{t-k} \phi_k$ is the eligibility trace. Note that standard TD(0) and TD(λ) iterates move in the direction of the feature ϕ_t or eligibility trace e_t at each time t. For some features or eligibility traces, coupled with a large step size α_t, the iterates may move drastically far away from the current location w_t. In contrast to the standard TD algorithms, implicit TD(0) and TD(λ) algorithms inversely scale the update direction by a factor of $1 + \alpha_t \|\phi_t\|^2$ and $1 + \alpha_t \|e_t\|^2$, respectively. Their update rules are given below.

  $$w_{t+1}^{\text{im}} = w_t^{\text{im}} + \frac{\alpha_t}{1 + \alpha_t \|\phi_t\|^2} \delta_t \phi_t,$$
  $$w_{t+1}^{\text{im}} = w_t^{\text{im}} + \frac{\alpha_t}{1 + \alpha_t \|e_t\|^2} \delta_t e_t,$$

  Such modifications prevent drastic movements by scaling the update direction by a factor inversely proportional to $\alpha_t \|\phi_t\|$ (for features) or $\alpha_t \|e_t\|$ (for eligibility traces).

References:

[1] Bhandari, Jalaj, Daniel Russo, and Raghav Singal. ”A finite time analysis of temporal difference learning with linear function approximation.” Conference on learning theory. PMLR, 2018.

[2] Xu, Tengyu, Shaofeng Zou, and Yingbin Liang. ”Two time-scale off-policy TD learning: Non-asymptotic analysis over Markovian samples.” Advances in neural information processing systems 32 (2019).

Reviewer 8Joz

Other Strengths And Weaknesses:

- Strengths: The authors propose Implicit TD learning method, which significantly reduces sensitivity to the learning rate. This claim is well-supported by numerical results presented in the paper. The paper also includes a finite sample analysis of the proposed method, aligning with state-of-the-art results in TD learning.

  Response: Thank you for acknowledging our contributions!

Questions For Authors: The idea behind this paper is quite straightforward. My main concern is about the numerical result.

- I find the bottom two figures in Figure 1 somewhat unclear. The y-axis includes a factor of $10^1 10$. It appears that when $\alpha = 1.581$, TD(0) fails to converge, right?

  Response: Yes, thank you for restating our intention. The main message we intended to convey in Figure 1 is that standard TD algorithms tend to suffer substantially from numerical instability when an inappropriately large step size is chosen.

- The results for TD(λ) are split into separate figures (the bottom two figures in Figures 2 and 3). If they correspond to the same experimental setup, I suggest merging them into a single figure.

  Response: The difference in Figure 2 and 3 is in the choice of projection radius. In Figure 2, we used the projection radius $R = 10$, while in Figure 3, the results are based on the projection radius $R = 100$. The message we intended to convey in these figures is that the performance of projected implicit TD algorithms tends to be robust with respect to the choice of the projection radius R.

- The environment used in the numerical experiments is relatively simple. Since the key advantage of implicit learning lies in reducing sensitivity to step size —an aspect that is not easily inferred from finite sample analysis— more extensive numerical validation is necessary. The random walk setting may be insufficient to show the novelty of the proposed method. I recommend testing the algorithm in more complex environments to better demonstrate its robustness.

  Response: Thanks for the suggestion. In response to the request for larger-scale experiments, we considered both 100-state MRP problems with 20 random binary features as well as classical continuous-domain control problems, including the Acrobot and Mountain Car environments. In all of our experiments, as shown in Figure 15, Figure 16, Table 13, Table 14 and Table 15, the numerical stability of implicit TD algorithms allowed us to deploy large initial step sizes, which facilitated faster exploration in the search space and yielded a computationally efficient parameter estimation strategy.

Method	λ	Mean	Std
Standard TD	0.0	5.355814	3.278592
Implicit TD	0.0	0.117330	0.044243
Standard TD	0.5	2.905596	1.483903
Implicit TD	0.5	0.212468	0.093600

Table 13: Final errors for 100 MRP experiments for each method and λ value.

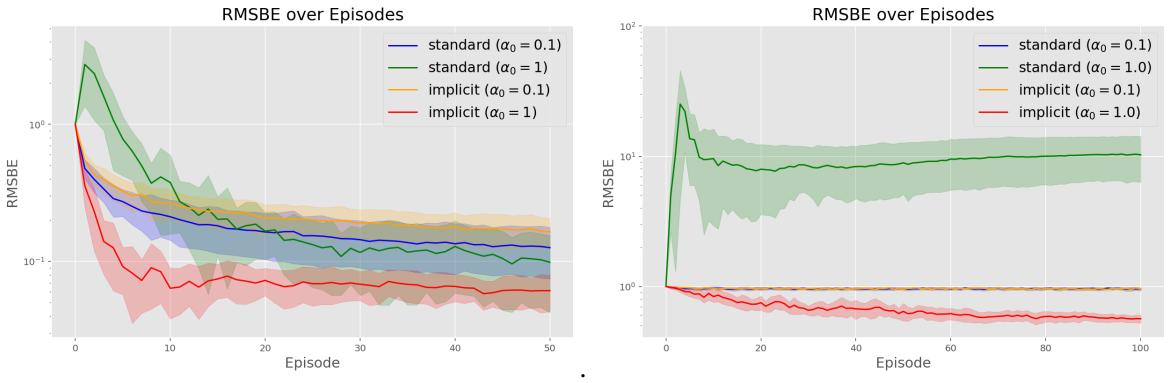

Figure 16: RMSBE plots for acrobot (left) and mountain car (right)

Method	α_0	Mean	Std
Standard TD	0.1	0.126078	0.051337
Standard TD	1.0	0.098693	0.056317
Implicit TD	0.1	0.164576	0.042195
Implicit TD	1.0	0.061291	0.018172

Table 14: Final RMSBE (Acrobot Environment) for standard/implicit TD(0) and step-size parameter α_0.

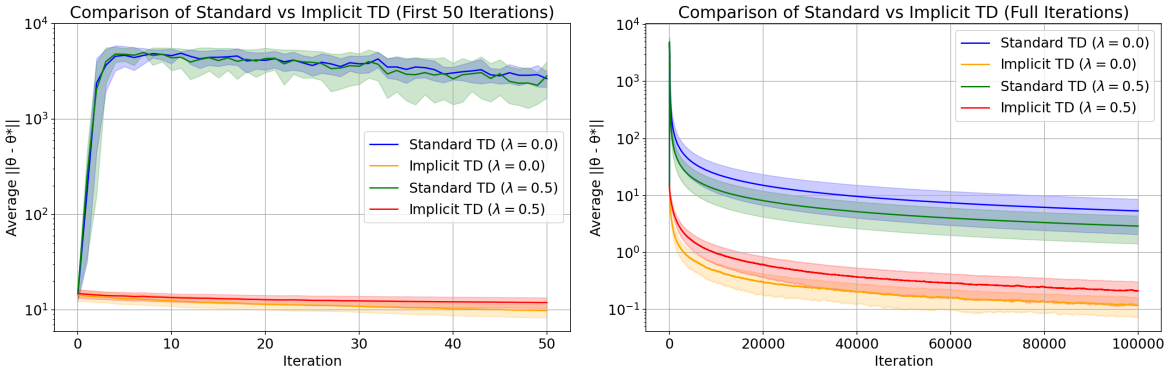

Figure 15: Parameter estimation error for synthetic MRP with 100 states (Left: 50 iterations, Right: 100000 iterations)

Method	α_0	Mean	Std
Standard TD	0.1	0.952269	0.026053
Standard TD	1.0	10.248247	3.938624
Implicit TD	0.1	0.951045	0.026131
Implicit TD	1.0	0.565690	0.041935

Table 15: Final RMSBE (Mountain Car Environment) for standard/implicit TD(0) and step-size parameter α_0.

- I think variance is also important in evaluating the stability of learning algorithms, as it reflects sensitivity to the initial conditions. The authors should incorporate variance estimates in their numerical results, possibly using shaded regions in the plots to illustrate uncertainty and improve interoperability.

  Response: Thank you for the helpful suggestion. We have included the standard deviation alongside the mean squared error to reflect the variation across independent experiments. The larger the shaded region, the greater the variation in algorithm outputs across the 50 independent experiments. Updated plots in Section 5 are provided in both Figure 3 and Figure 4.