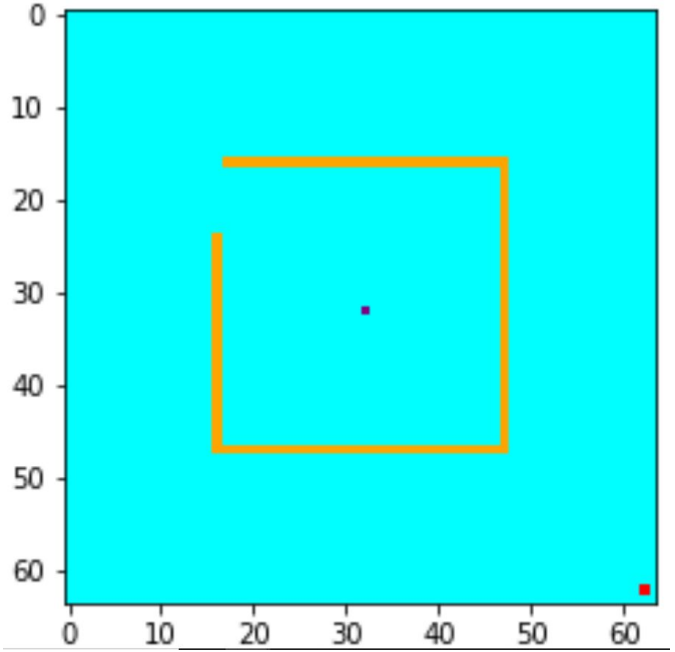
GridWorld Experiments

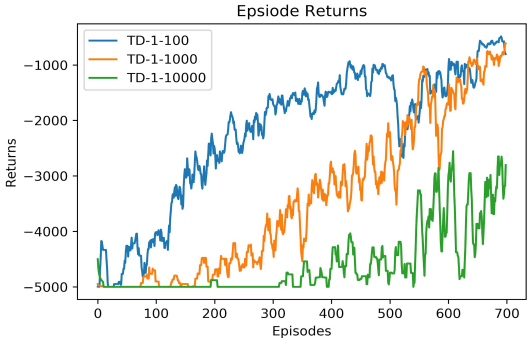
Environment

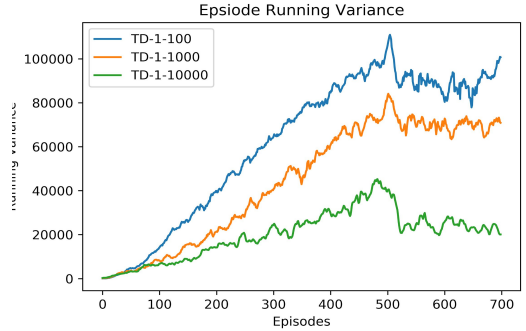
- Dim = 64x64
- Actionspace 4
- Rewards:
  - 0 on Goal
  - -1 otherwise
- Legend
  - Orange - Walls
  - Purple - Goal
  - Red - Start



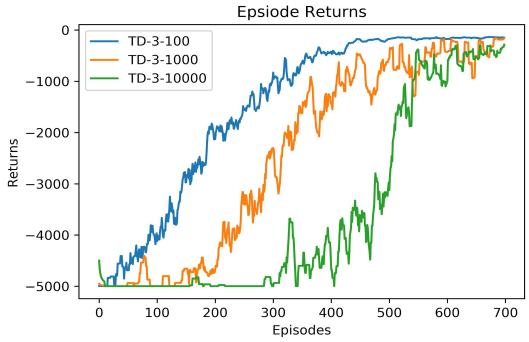
Comparison - Uncorrected n-step reward (Const. n)

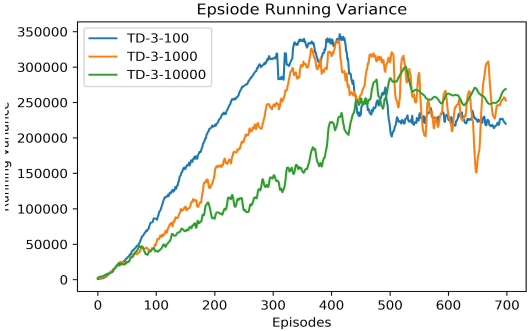
N=1





N=3



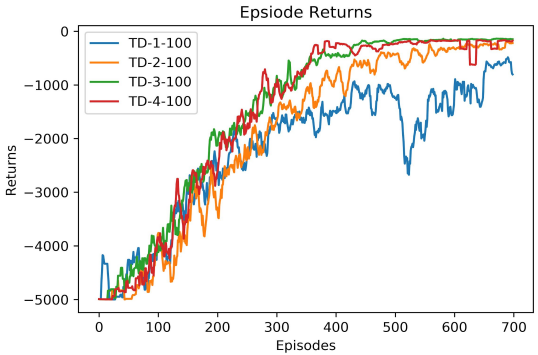


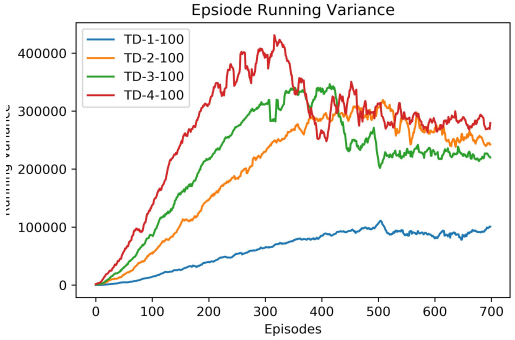
Observations - Uncorrected n-step reward (Const. n)

- Larger Replay buffers help mitigate the variance of the target update. This is true irrespective of the length of the n step as the phenomenon is seen even at n=1.
- The running variance decreases as the q values converge to an optimum.
- In the grid world setup all methods guaranteed to converge. However, a larger replay buffer slows this process down. This can viewed from the lens of on-policyness as smaller replay buffers guarantee more up to date transitions.

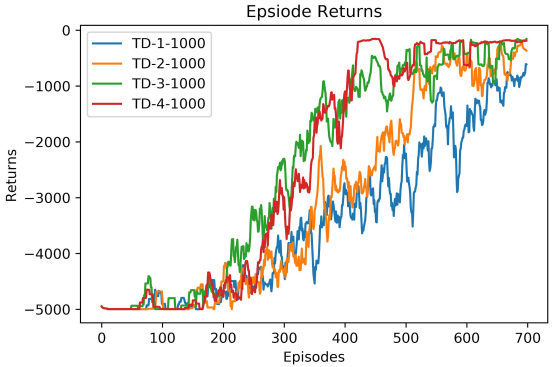
Comparison - Uncorrected n-step reward (Const. buffer)

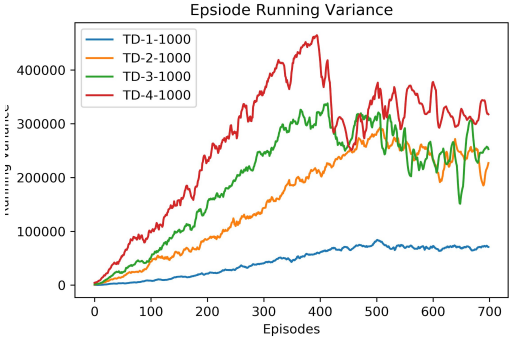
Buf = 100



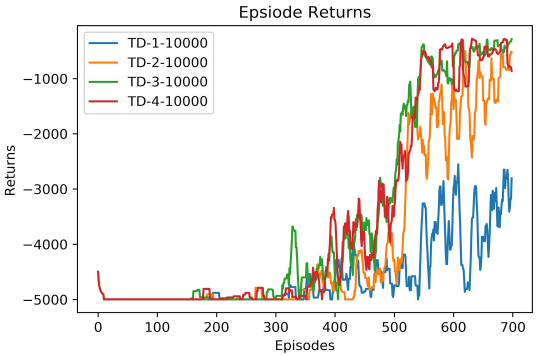


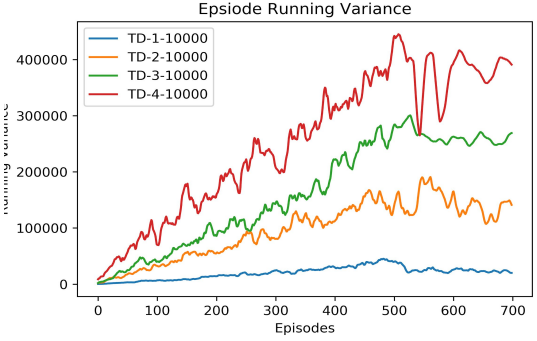
Buf = 1000





Buf = 10000

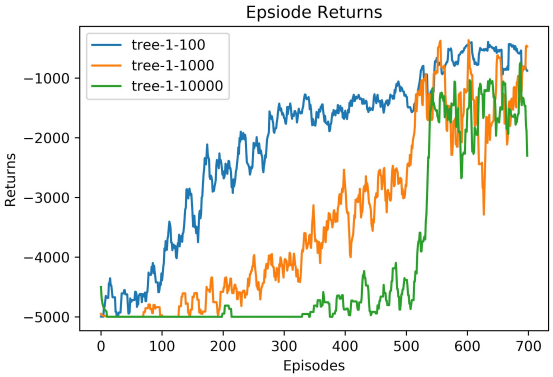


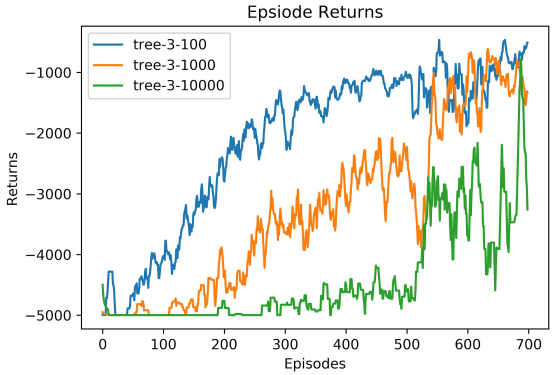


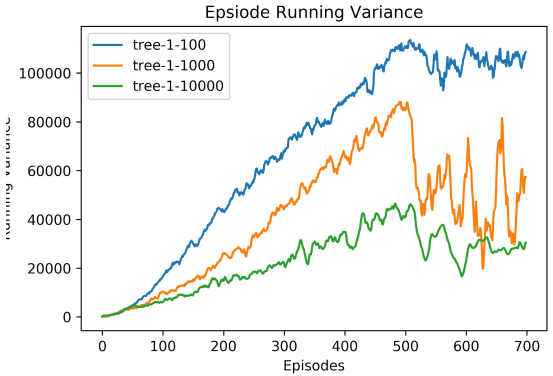
Observations - Uncorrected n-step reward (Const. buffer)

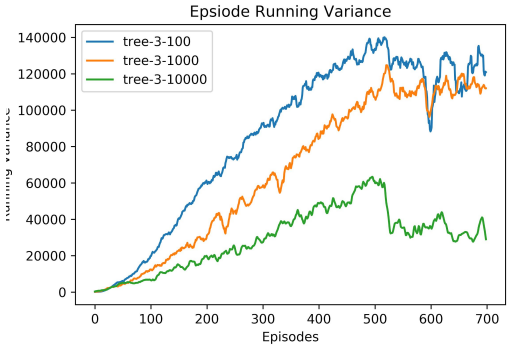
- Larger *n* values lead to faster convergence. This effect diminishes after n=3. This is expected as reward of a terminal state is propagated to more states each time it is encountered until its yield is diminished.
- Larger n values register a sharper drop in variance on approaching an optimal policy / value function.

Comparison - Tree Backup (Const. n)

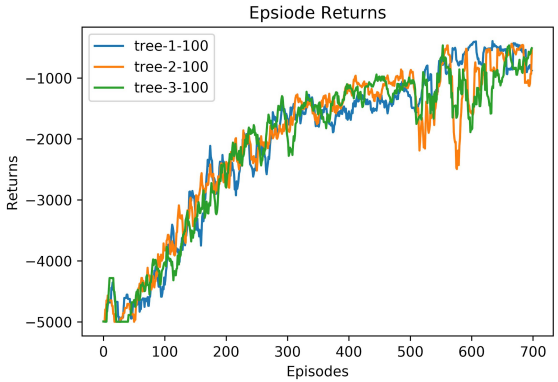


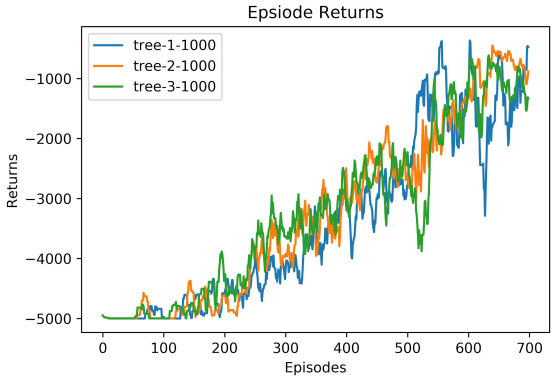


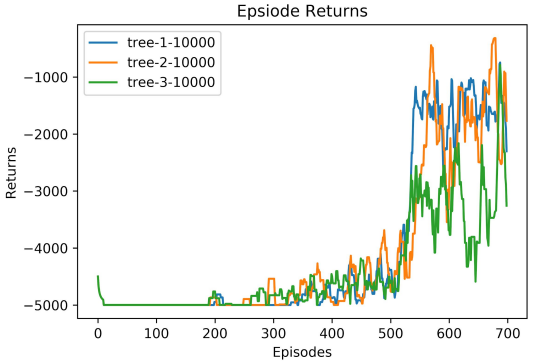


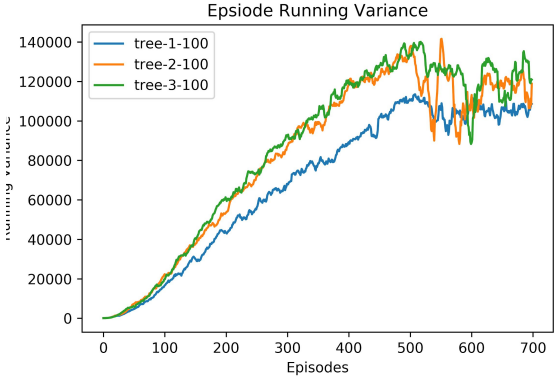


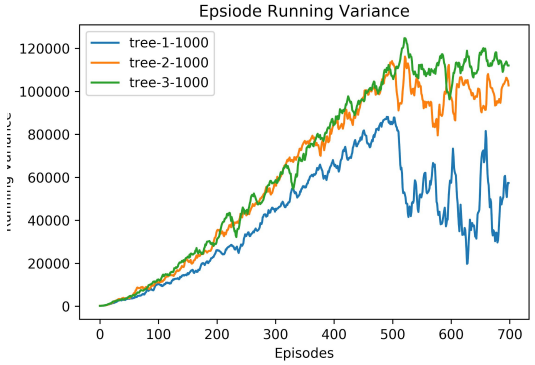
Comparison - Tree Backup (Const. Buffer Size)

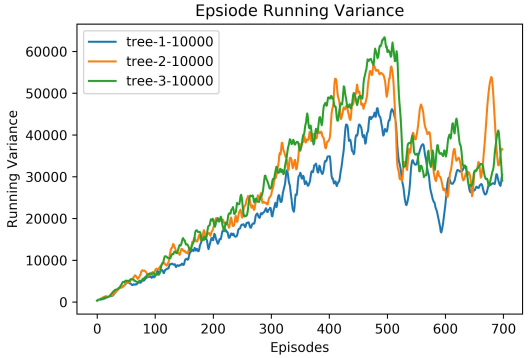








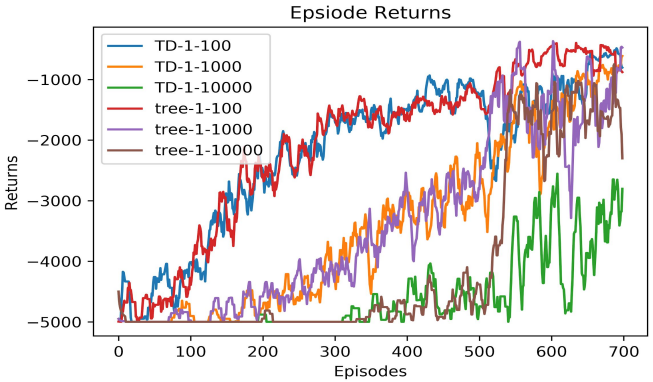


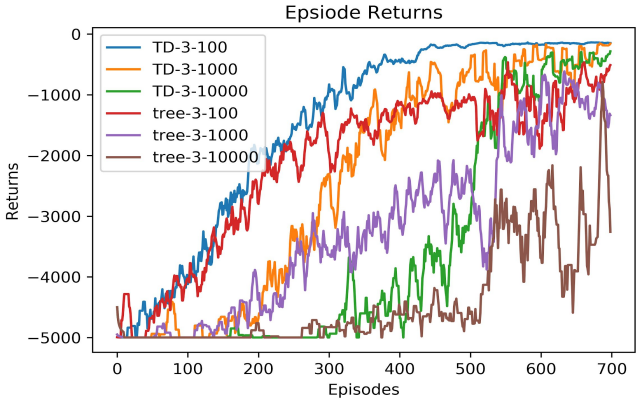


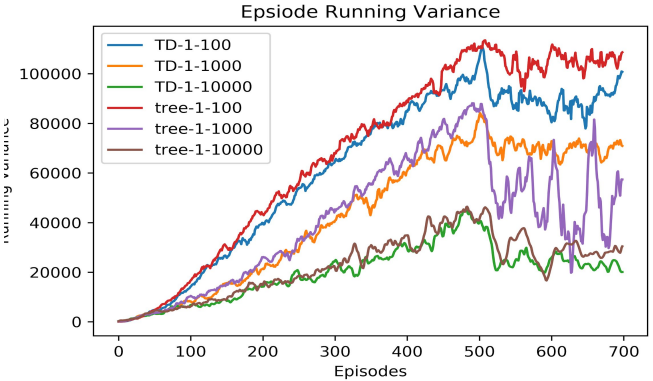
Observations - Tree Backup

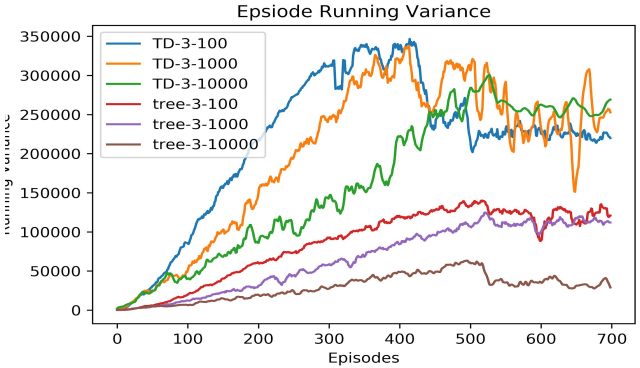
- This section yields interesting results. As hypothesized earlier, the lower variance of the tree backup return as compared to the uncorrected n step return diminishes the gains from a larger replay buffer on moving from a 1 step to a 3 step return.
- While the resulting running variance of a 3 step is higher than a 1 step return the gap is quite small and more importantly, the episode returns are nearly the same.

Experiments - Across methods (Const N)



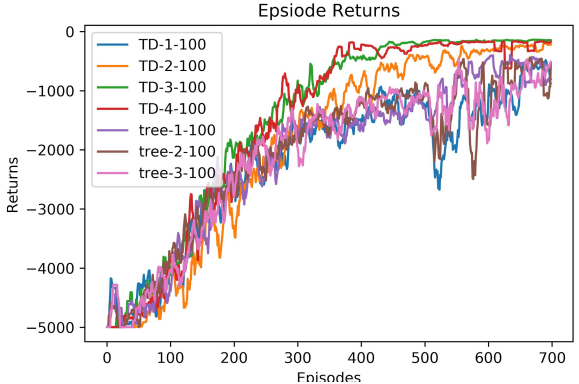


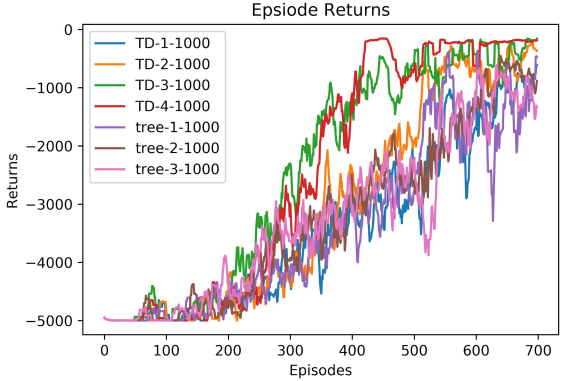


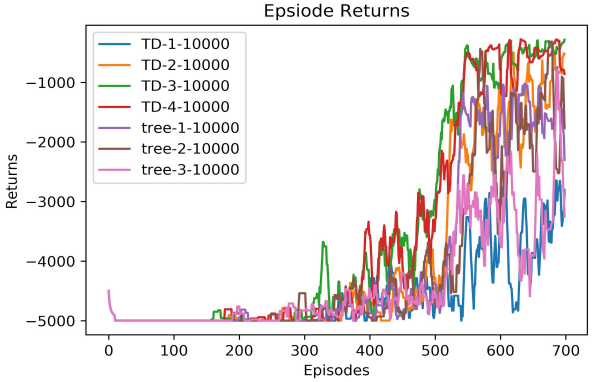


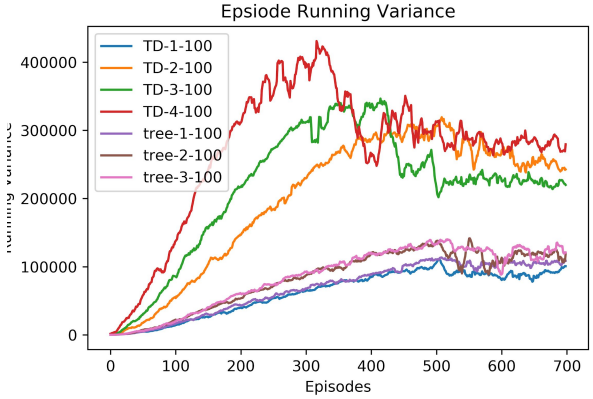
Larger n appear to slow down tree backup in comparison to uncorrected n step. The reduced variance appears to play a role in this.

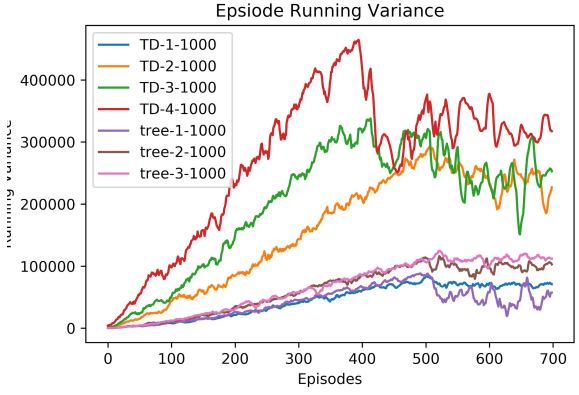
Experiments - Across methods (Const Buf)

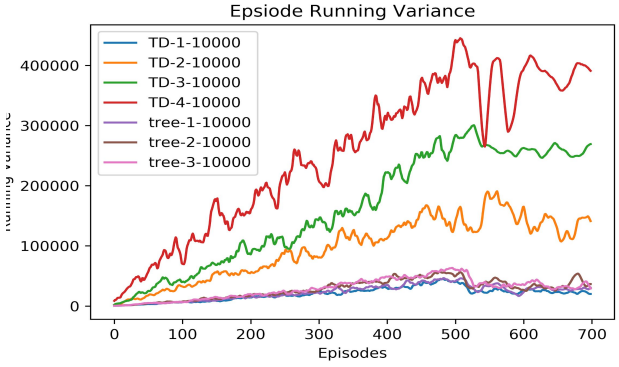












Observations - Across methods (Const Buf)

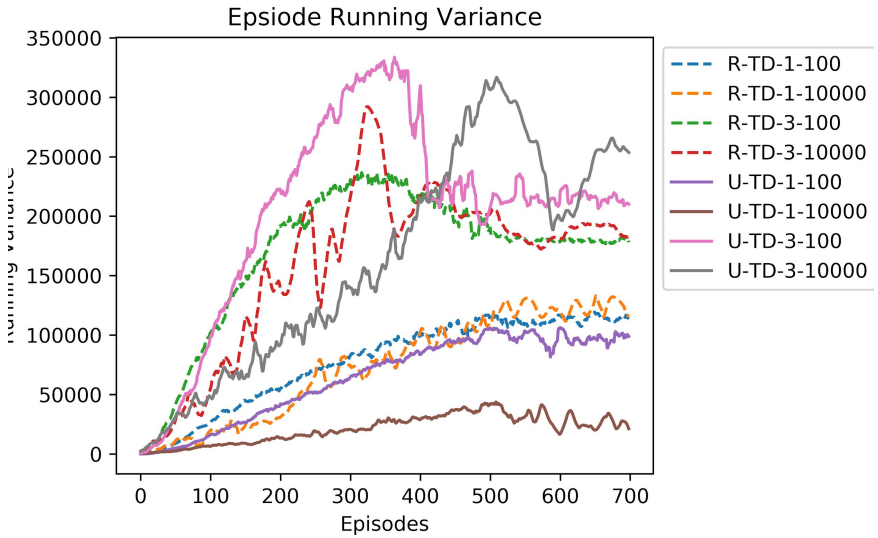
- The initial hypothesis are held - lower variance in tree back vs uncorrected n step.
- This comes at the cost of slower convergence.
- More variance in the initial stages leads to more exploration. Better not to curb it as it helps in faster convergence.

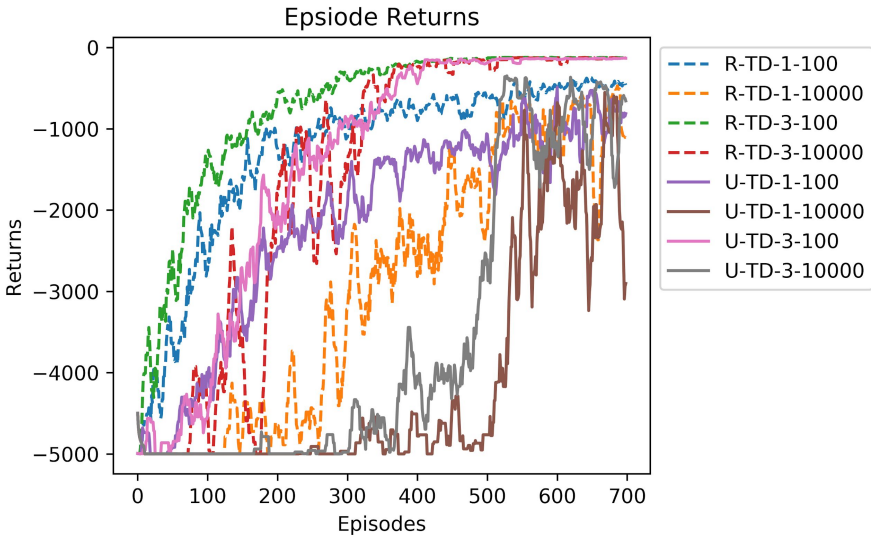
Question - Can refreshing transitions while replaying them speedup convergence?

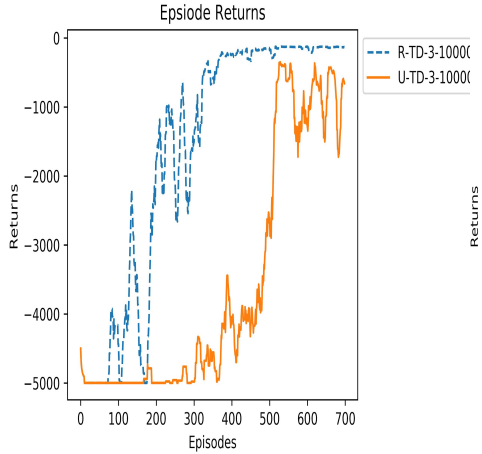
Hypotheses

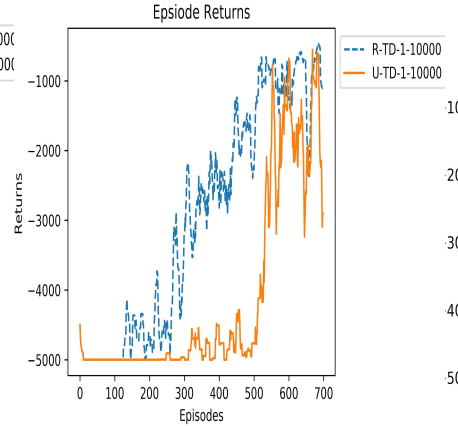
- Yes it should. While the transition itself may have a low likelihood of being selected by the current policy, the action and update will be more on policy.
- The variance of the update should increase as each update of a given transition will be from a changed policy

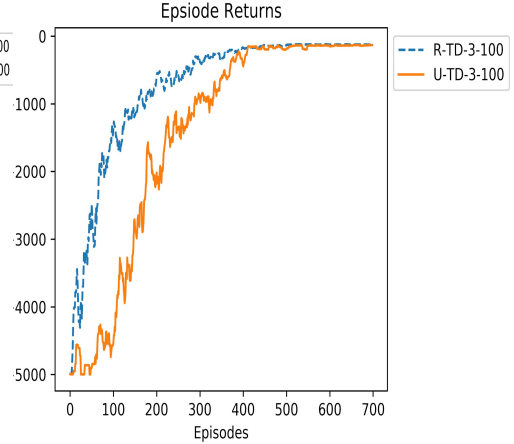
Experiment - Refresh Vs Uniform Sampling



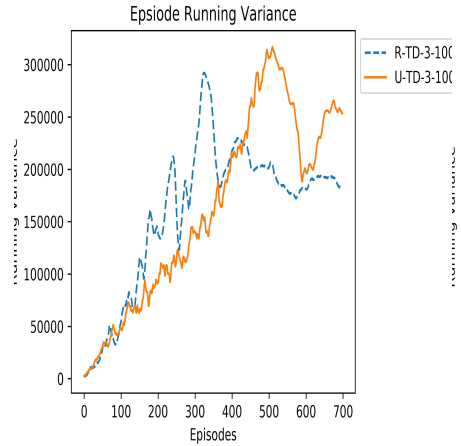


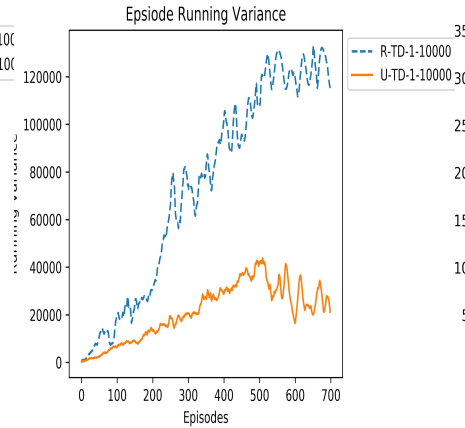

Epsiode Returns
R-TD-3-1000(
U-TD-3-1000(
Returns
Episodes

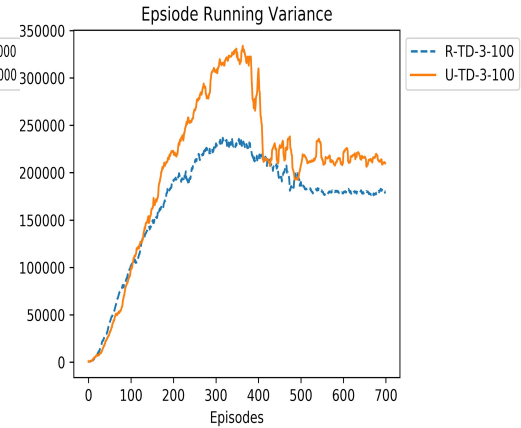

Epsiode Returns
R-TD-1-10000
U-TD-1-10000
Returns
Episodes

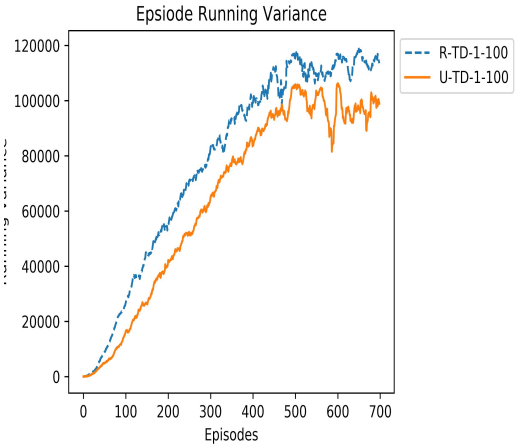

Epsiode Returns
R-TD-3-100
U-TD-3-100
Episodes


Epsiode Returns
R-TD-1-100
U-TD-1-100
Returns
Episodes


Epsiode Running Variance
R-TD-3-10(
U-TD-3-10(
Episodes


Epsiode Running Variance
R-TD-1-10000
U-TD-1-10000
Episodes


Epsiode Running Variance
R-TD-3-100
U-TD-3-100
Episodes


Epsiode Running Variance
R-TD-1-100
U-TD-1-100
Episodes

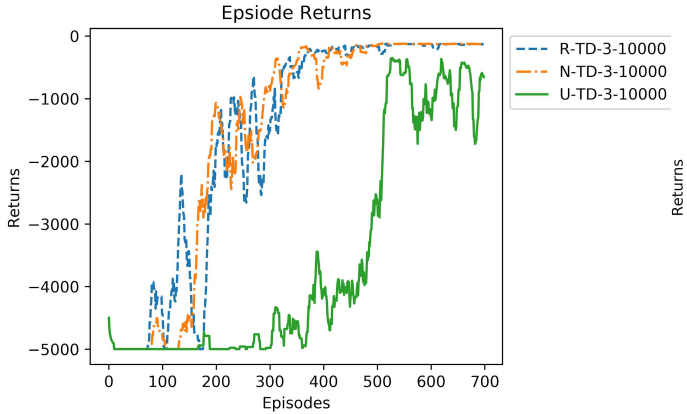
Observations - Refresh vs Uniform

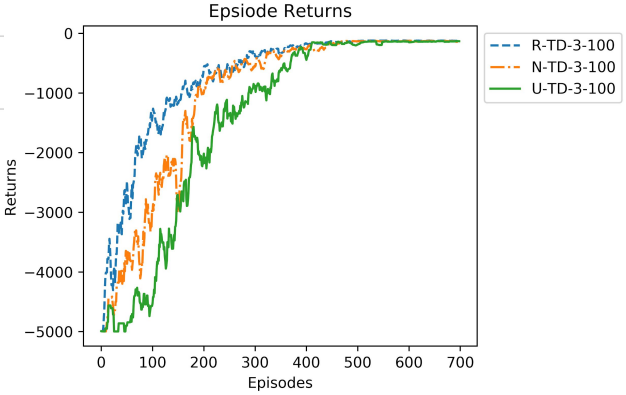
- There is a considerable speedup in the convergence across buffer size and n.
- However there is no concrete variation in the running variance. Though it does appear to have a higher variance during the early stages of learning for higher buffer sizes.
- This makes sense as larger replay buffer will contain highly off policy transitions for which the policy could undergo significant change in the early stages leading to higher variance.
- The impact of Refresh is more noticeable in these larger buffers.

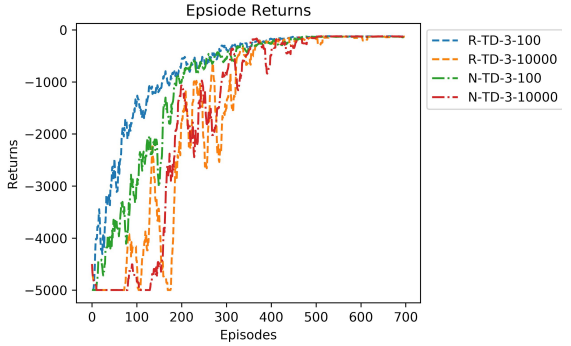
Caveats - Refresh

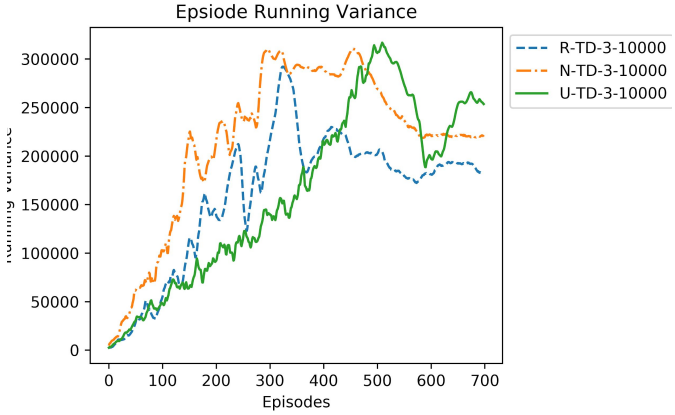
- While promising, Refresh cannot immediately be applied to complex environments as replaying an arbitrary state may not be possible as a consequence both the next state and the new reward would be difficult to obtain.
- A workaround this would be to train a forward dynamic model to predict the next state as well as some way of generating a reward signal (such as intrinsic motivation).
- This however would be noisy. To test a somewhat similar condition we compare a noisy version of the Refresh as well.

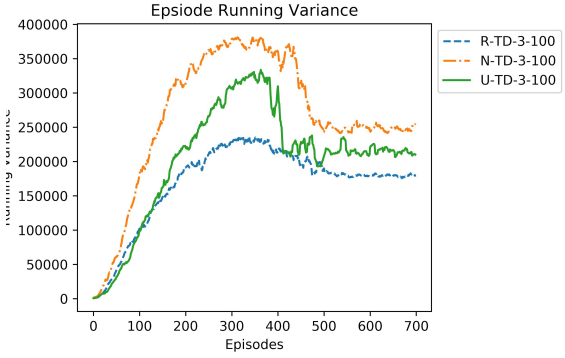
Refresh (R) vs Refresh Noisy (N) vs Uniform(U)

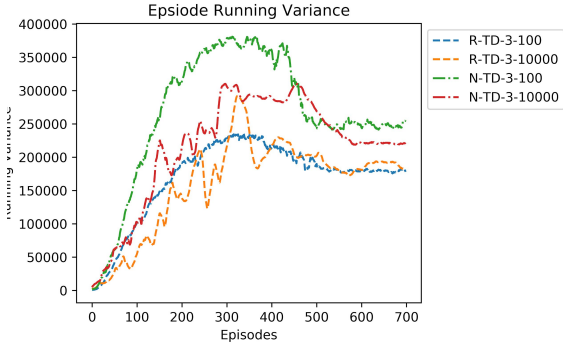








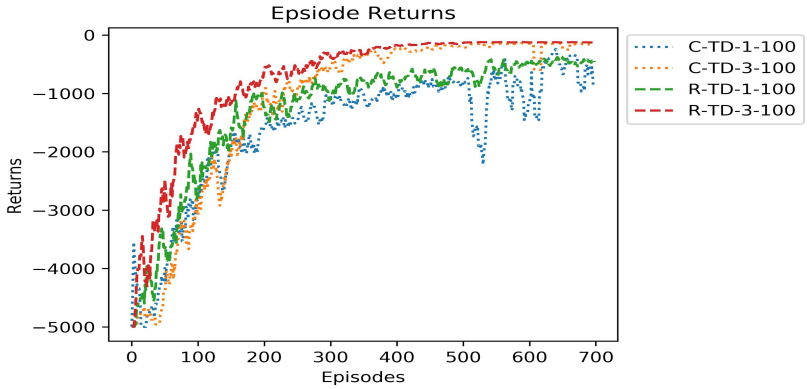


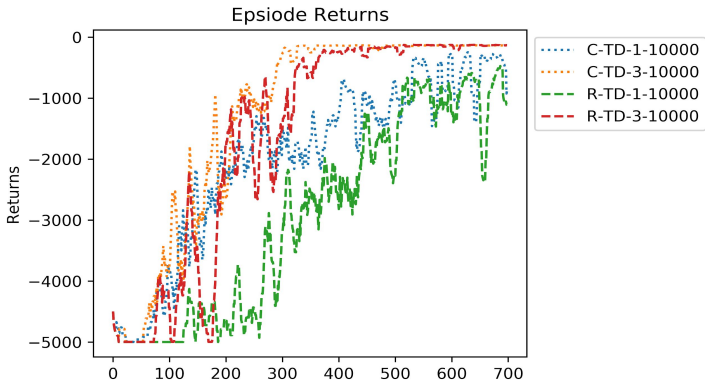


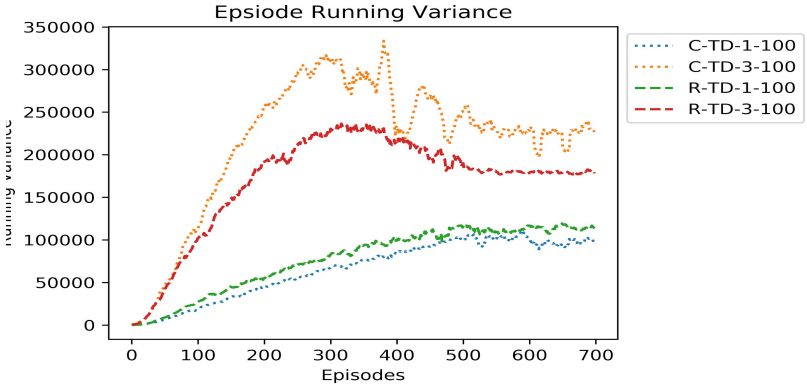
Observations

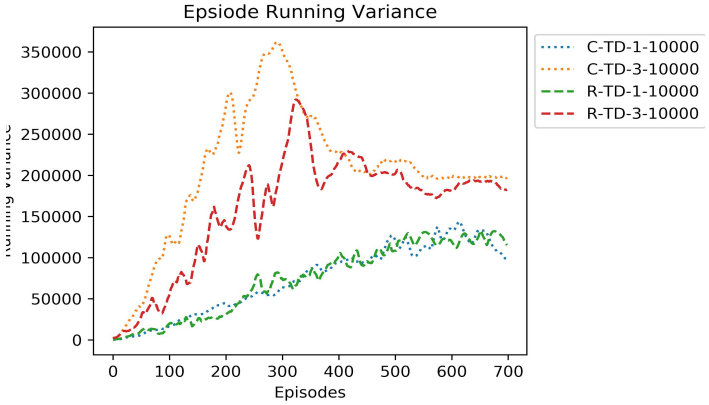
- The Noisy refresh performs comparably to the noise free version.
- Moreover its running variance is at par or higher than the other two.
- This indicates that such an approach may prove useful even on more complex settings.
- A consequence of the high variance would indicate that this approach would also benefit from larger replay buffers

Experiment - Refresh Vs CER





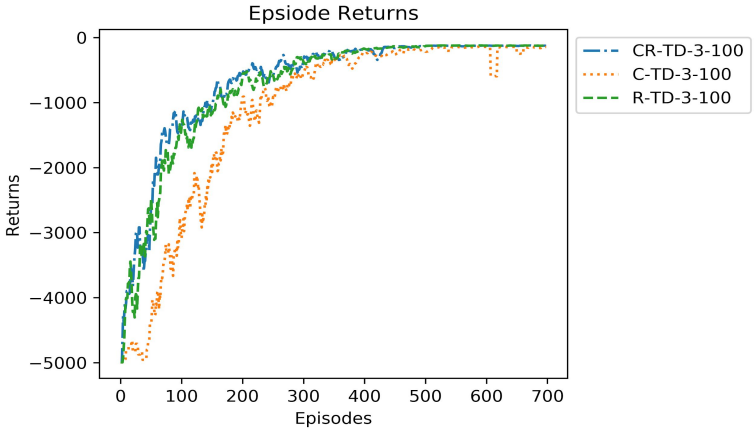


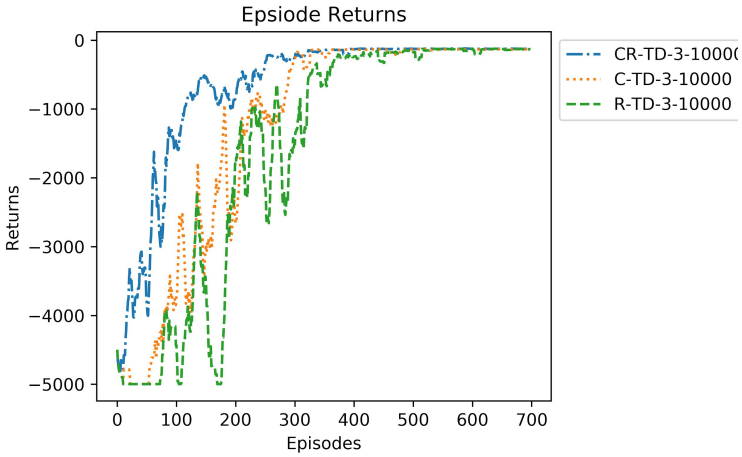


Observations

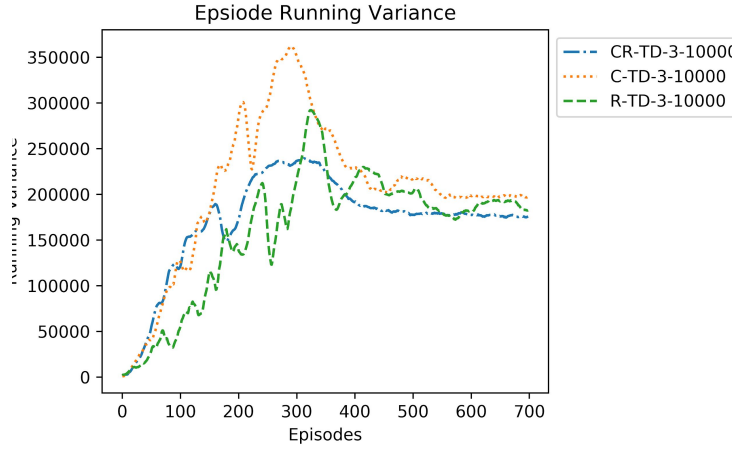
- CER performs at par or slightly better than Refresh.
- It can be seen that for n>1, CER exhibits a higher variance than Refresh. This is puzzling because CER just replaces 1/10 transitions in each batch with the latest transition. This however seems to have a notable impact on the initial running variance.
- It perhaps allows more exploration in early phases.
- A natural question then arises is whether the combination of CER and Refresh can achieve faster convergence.

CER+Refresh

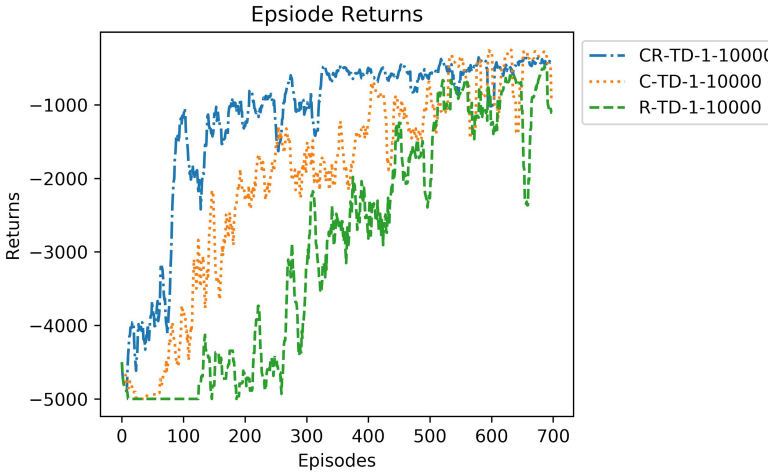

Epsiode Returns
Returns
Episodes
CR-TD-3-100
C-TD-3-100
R-TD-3-100

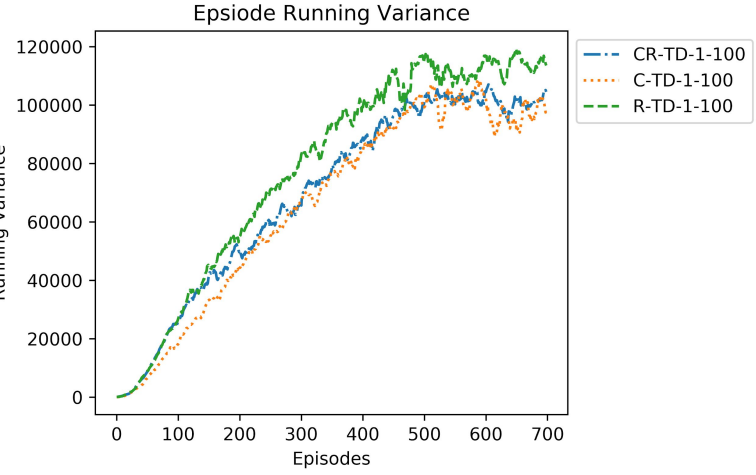

Epsiode Returns
Returns
Episodes
CR-TD-3-10000
C-TD-3-10000
R-TD-3-10000

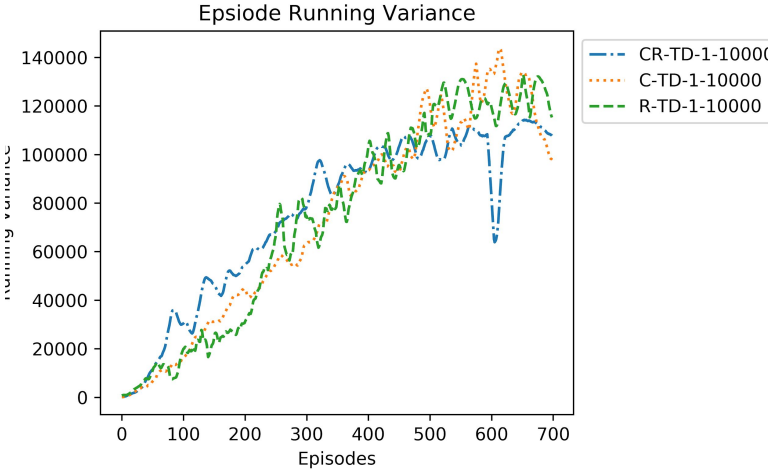

Epsiode Running Variance
Episodes
CR-TD-3-100
C-TD-3-100
R-TD-3-100


Epsiode Running Variance
Episodes
CR-TD-3-10000
C-TD-3-10000
R-TD-3-10000


Epsiode Returns
Returns
Episodes
CR-TD-1-100
C-TD-1-100
R-TD-1-100


Epsiode Returns
Returns
Episodes
CR-TD-1-10000
C-TD-1-10000
R-TD-1-10000


Epsiode Running Variance
Episodes
CR-TD-1-100
C-TD-1-100
R-TD-1-100


Epsiode Running Variance
Episodes
CR-TD-1-10000
C-TD-1-10000
R-TD-1-10000

Observations

- This is indeed true. CER + Refresh does show benefits by speeding up convergence especially in a larger buffer as compared to the individual variants.
- The variance profile for n>1 is similar to Refresh rater than CER.

Insights?

- What does Refresh solve - Does it make exploration more efficient or does it learn a better policy to navigate the explored state space.
- Would large state spaces with sparse rewards benefit from a method that could learn meaningful information from rewardless episodes? Such as learn the shortest path from start to an intermediate point this would enable faster navigation to the boundary of the explored region.
- A combination of high early variance and such an intermediate reward scheme could enable exploration as well the learning of useful behaviour simultaneously.

How does the replay buffer look over training?

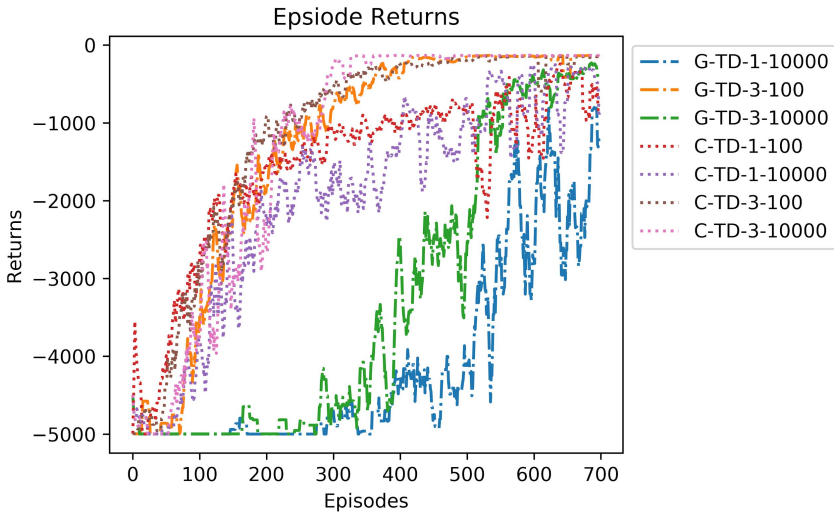
- The replay buffer shows oscillating behaviour of states. It goes one way for a while, the returns gradually reduce and then it goes the other way.
- This trend continues until the policy nears the optimal. As this happens the replay buffer consists of a near shortest path from the source to goal.
- CER appears to become more path like faster than uniform sampling.
- This may indicate that it might be useful to actually learn optimal policy within a region while exploration happens at its frontiers.
- Lends credence to intermediate rewards to learn from otherwise no reward episodes

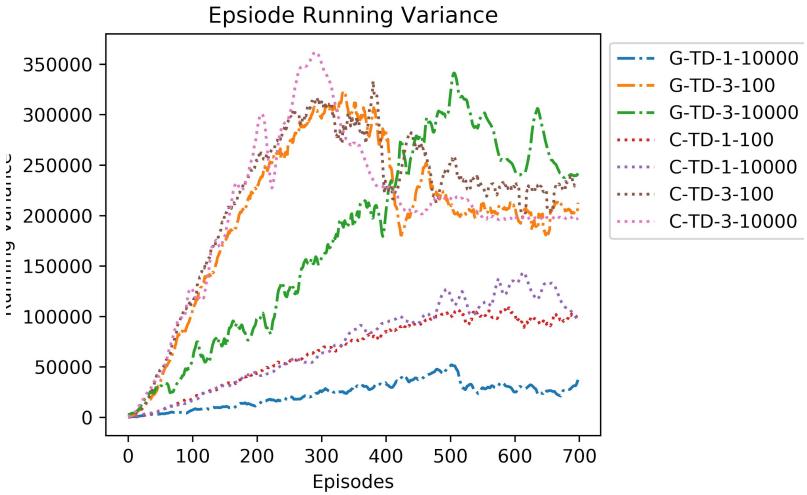
Hypothesis

CER shows that immediately replaying the current transition greatly improves convergence.

- Could this be reasoned as replaying recent transitions early hastens convergence?
- Would having an exponential distribution over time in the replay buffer be a natural extension of this?

Geometric Decay Buffer





The hypothesis does not hold. This buffers is similar to the uniform sampling scheme. (p = 0.11 and buckets = 20)