MetaReview

The reviewers had some concerns about the incremental nature of the paper and some missing information. They were pleased with the rebuttal although they expressed the concern about whether the authors will uphold their promise of increasing the number of experiments done. If this is the case, they are happy to accept it.

Reviewer-1

Overall Recommendation
1: (weak accept)

Summary
The authors propose an evolutionary reinforcement learning algorithm Evolution-based Soft Actor-Critic (ESAC). The algorithm is evaluated using locomotion tasks. The authors report improved performance compared to other state of the art DRL methods.

Detailed Comments
Pros:
Well written.
Good experimentation.
Of broad interest to AAMAS community.

Cons:
Slightly incremental.

Changes:
Cannot read text in some figures, e.g. Fig 6 axis titles.
No conclusion section. What are the conclusions of the work?
It would furhter strengthen the results to also compare to ERL algorithm in reference [13].

Rebuttal Question
How come you did not compare the results of ESAC to ERL [13]?

Reason for Recommendation
I think this is a good paper. Comparing it to ERL [13] would make it a very good paper.

Reviewer-2

Overall Recommendation
1: (weak accept)

Summary
The paper proposes a new algorithm for calculating optimal policies in situations where function approximation is needed for long-horizon reinforcement learning problems. The algorithm combines features evolution strategies on one hand and another algorithm called soft actor-critic (SAC). The authors explore the effectiveness of their method on a toy problem as well as on some traditional benchmark applications for reinforcement learning problems. Overall the proposed method has promising experimental results.

Detailed Comments
I think that the paper is well-written, relevant to AAMAS, and a big part of it self-contained and easy to follow. However, there are times where I believe that the paper could provide more information and make it a little bit easier for the reader to follow. Perhaps the appendix should be longer and discuss some issues there?
-- For example, one of the main arguments that the authors have is the use of "hindsight crossovers" that allow simultaneous transfer of dominant traits. After reading the paper it is not so clear to me what is so special about the crossover operator that the authors use and I feel that there is something interesting behind this notion that somehow I have not been able to follow fully.

Switching topic, the averages that we see in most of the experiments are the outcome of very few iterations (of the order of 3, or along these lines). This is ridiculously low, especially given the running time that we can see on the plot, where, e.g., the worst-case wall-clock execution of the experiment is of the order of 40 seconds (see, e.g., Figure 2b). Please try to do more experiments in the final version.

After some equations (e.g., 2 and 3) you are leaving an empty space in latex which results in indenting your next paragraph when you in fact continue the same sentence.

Rebuttal Question
Q1. In section 2.2 you mention "mutation robustness". Can you elaborate what you mean by "robustness"?

Q2. Can you provide more information on the notion of "dominant skill transfer" and on its connection to "hindsight crossovers"?

Q3. In Figure 6: you compare ESAC with SAC in terms of the number of updates, but it appears that SAC has slightly better cumulative reward in these two benchmark problems (based on Table 1). This is just a comment and in fact in one of the two cases (HalfCheetah-v2) the difference seems to be significant, or very close to being significant. However, Figure 6 (left) is very strange. Is there some intuitive explanation as to why the reward drops so much when the clip parameter takes the value of 0.1 or 1 (depending on the benchmark problem)?

Q4. I think the source code should be available online. Do you plan to make it public after the paper gets accepted?

Reason for Recommendation
- New algorithm that is parallelizable and moreover it behaves well on several benchmark problems.
- For the largest part, the paper is well-written.

Reviewer-3

Overall Recommendation
2: (accept)

Summary
This paper leverages evolutionary methods for exploration in RL (SAC).
The authors evaluated the sample efficiency, scalability, and the mutation sensitivity of their framework. They introduce AMT to maximize the mutation rate in a clipped region to address the mutation sensitivity.

Detailed Comments
I have not noticed major issues with the paper, except for a couple of concerns that are expressed in the Rebuttal section.

Mino fixes:
- Figure 3 appeared prior to Figure 2
- latter → later

Rebuttal Question
1- Why in Figure 2-left, the ES and DDPG methods do not have any oscillations?
2- Any explanation on why ESAC does not perform the best in 5 environments in Table 1?

Reason for Recommendation
I believe the claims in the paper are well supported by the experiments.

RESPONSE FROM AUTHORS

We thank the reviewers for providing feedback which is of utmost value to our work. Below we address your questions in detail.

Reviewer-1

Cannot read text in some figures, e.g. Fig 6 axis titles.
We apologize. Title and legend texts have now been enlarged for the final version.

No conclusion section. What are the conclusions of the work?
We provide our conclusions in Section 9.1 under the "Summary" sections heading. To make our conclusions more explicit we have renamed this section to "Conclusions" in the final version. Thank you for the suggestion.

How come you did not compare the results of ESAC to ERL [13]?
Thank you for pointing this out. Following your suggestion, we have started running additional experiments for ERL baseline. These experiments will be included along with details of ERL hyperparameters in Appendix.

Reviewer-2

Please try to do more experiments in the final version.

Thank you for pointing this out. We have started addressing your concerns by running more seeds of our algorithm. Specifically, we are evaluating the CyclicMDP experiments for more runs, given its fast computational times. Additionally, we aim to add more seeds (over 5 random seeds) for our locomotion experiments as well.

After some equations (e.g., 2 and 3) you are leaving an empty space in latex which results in indenting your next paragraph when you in fact continue the same sentence.
We apologize for this. Blank spaces and indentations have now been corrected.

In section 2.2 you mention "mutation robustness". Can you elaborate what you mean by "robustness"?
Mutation robustness here refers to an agent's ability to resist unnecessary mutations when the agent is already evolved to be optimal. In a population of offsprings, some actors may evolve quickly and demonstrate optimal returns when compared to the rest. However, additional rounds of mutation may inject noise in these actors and hurt performance. These actors, thus, must be robust to mutations when convergence has taken place. Our final version provides an explicit definition of mutation robustness "the ability to resist mutation noise when converged" in Section 2.2.

Can you provide more information on the notion of "dominant skill transfer" and on its connection to "hindsight crossovers"?
Dominant skill transfer indicates the usage and transfer of skills acquired by the winners of the population. These skills are dominant as the dominant actors (i.e-winners) select these skills to improve performance. Transfer of dominant skills from winners to other offsprings would improve overall performance of the population. This process is carried out in two stages; (1) soft-winner selection and (2) hindsight crossovers. In the first stage, we select top k winners of the population and save their parameters in a set W. In the second stage, these parameters are used to carry out crossovers with the main policy parameters. This way, skills from the winners of previous generation are transferred to the offsprings of the next generation, i.e- crossovers taking place in hindsight. Our final version will provide the above details on dominant skill transfer and hindsight crossovers in appendix.

Is there some intuitive explanation as to why the reward drops so much when the clip parameter takes the value of 0.1 or 1 (depending on the benchmark problem)?
For low values of the clip parameter (1e-4 to 1e-2), the population is found robust to excessive mutations. This is because the method clips updates to smaller values and no additional noise is injected in the parameters. For high values of clip parameter (> 1), significant noise is injected which hurts the performance of weak learners. However, strong learners, being robust to excessive mutations, converge and improve the policy. In the special case of moderate clip parameter values (1e-1 to 1), actors are sensitive to perturbations. Mutation updates are sufficient to inject noise in weak learners but, at the same time, these values are small enough to prevent exploration among strong learners. This way, both the strong and weak learners of the population suffer.

I think the source code should be available online. Do you plan to make it public after the paper gets accepted?
We firmly believe in the reproducibility of our work. To this end, we will be open-sourcing our code implementation along with detailed notes on reproducibility. Further, we have made

our code modular and easy-to-read along with a short blog post which would help the community better navigate and build upon it.

Reviewer-3

Figure 3 appeared prior to Figure 2.
We apologize. This is now corrected for the final version.

latter → later
Once again, apologies. This is now corrected.

Why in Figure 2-left, the ES and DDPG methods do not have any oscillations?
ES and DDPG converge very quickly on the CyclicMDP task due to its simplicity. The task only consists of 3 states and 3 actions. This allows ES and DDPG to converge by only learning from a handful of transitions. PPO, on the other hand, learns using a proximal update rule. The update can be noisy due to the presence of two surrogate objectives and an entropy penalty in PPO. This leads to frequent oscillations in PPO's returns. We further note that the plot is averaged over 3 random runs. This leads the oscillations to average out as well.

Any explanation on why ESAC does not perform the best in 5 environments in Table 1?
ESAC's improved performance is a result of its simultaneous exploration and exploitation using mutations. In scenarios where the magnitude of mutations is small, ESAC may not get a good generalization signal. This leads the method to demonstrate comparable performance as SAC. While we note that this does not hurt ESAC's performance, it could still lead us to answer further questions on improving mutations. We mention this in the limitations section (section 9.2) and leave it as an interesting direction for future work.