We thank all reviewers for their valuable feedback and comments. Please find our responses below.

Reviewer 1 - Explanation in the introduction: we strive for clarity and we appreciate this comment. We will elaborate on the algorithm description accordingly.

- Relationship to Bayesian RL. While we discussed Bayesian RL in the context of posterior sampling, we will elaborate on the relation with Dearden et al. (1998) in the revised version. We thank the reviewer for pointing this out.

Reviewer 2 - Reproducibility: We will release the code upon acceptance of the paper and include the experimental details in the revised version.

- Full-rollouts: The analysis is not limited to planning on full rollouts. Instead, any algorithm that solves the HUCRL objective (7) retains the theoretical guarantees. This can be done in many ways as discussed in Appendix C.

- Selection of β: As stated in Lemmas 10 and 11, β has to grow sub-logarithmically with time in order for the confidence intervals to be correct. The theoretical value used for the bounds is rather conservative however. In practice, we simply set $\beta = 1$ (without tuning), as it is often done in bandit algorithms. We will clarify this in the experimental setup.

- Comparison to competing methods: Our competing methods are Thompson Sampling and Greedy. SAC/PPO/POPLIN can be used to solve the greedy planning problem by simulating transitions with the learned model. For example, Greedy with NN ensembles is the PETS-DS algorithm (which we compare against). POLO and POPLIN are approximate planning algorithms given a model and, as such, could be used to optimize the HUCRL objective (7).

- Calibration Assumption: although this is crucial for the theory, in practice we verified that the resulting confidence intervals contained the true predictions (not necessarily calibrated) but did not employ any re-calibration procedure.

- Tasks: We evaluate on the same tasks as in Chua et al. (2018), which have randomly sampled initial conditions. Also, the sparse inverted pendulum (which is not a Mujoco task) has transitions with additive Gaussian noise.

Reviewer 3 - Scaling: in lines 190-201 we clarify that we parameterize π and η and optimize them jointly using a policy search algorithm. To address instabilities, we could restart the optimization algorithm for π and η as there is no need for such policies to be close to each other between episodes. We did not encounter such instabilities in practice.

- Difference to Gopalan & Chowdhury (2019): as per your review "The main extension is to make this optimization tractable", we find this a big contribution (the previous method is not implementable/practical). Furthermore, we discuss the differences in Appendix H.3, and in lines 73-78.

- Stochasticity: In the HUCRL algorithm (7), J is the expectation w.r.t. the stochastic uncertainty, we will clarify this.

- Multi-modality: When the next-state distribution is multi-modal the choice of the model likelihood is crucial. We can still be optimistic w.r.t. the epistemic uncertainty in the model and use a multi-modal stochastic noise model.

- "Dyna" name: We use Monte Carlo sampling with bootstrapping in a receding-horizon fashion to solve objective (7), which is common as you mention. The difference is that we learn π_θ using samples from the model (Dyna) and then guide the Monte Carlo sampling for MPC with such a policy, hence Dyna-MPC.

- Discontinuous-reward functions: We meant that discontinuous rewards are known to be hard to optimize (even in the bandit setting it is NP-hard). We will clarify that this does not mean that discontinuous rewards are impossible to solve. Note that Atari has discrete states and that rewards are linear functions of the tabular encoding of the state. Furthermore, in most games there is local reward information, whereas "hard" exploration games were recently solved using optimism.

- Experiments: We will use the extra space of the revised version to include the learning curves of environments.

- Action-penalties: All environments have action penalties. There is a typo in the cheetah reward (l. 681), it is $r = \min(v, 10) - \rho \sum_i a_i^2$. This is common in continuous control as it shapes the intended behaviour to consume less energy.

- Colors: Thanks for catching this. We will augment the colors with line-styles and markers.

Reviewer 4 - Choice of action penalties/environments: We used the same environments as in PETS by Chua et al. (2018), which used action penalties. Higher action penalties are commonly used to reduce the energy consumption while achieving the same goal. Simultaneously, higher action penalties favour zero-action against random actions. Thus they also penalize exploration when the greedy exploration objective is used.

- Reduction to Greedy: when considering the joint inputs (a, η), (7) can be solved with standard planning algorithms.

- Significance of Thm 1: i) Having sublinear regret implies that as $T \to \infty$ the average incurred cost converges to the optimal cost. ii) In Fig. 1, optimism finds a good policy whereas TS and Greedy do not. iii) Assumptions: After each assumption we clarify whether it is met in practice.

- Reference: This was published after our submission. We will include a reference to it as contemporary work.