We thank all the reviewers for the constructive feedback. We will incorporate the valuable suggestions in the revised version. Below, we respond to all of the reviewer comments, including multiple new experiments as requested:

R1: *"fairly limited in terms of applicability... the ability to extend this work to more general settings?"* Since submission, we have tested MOPO on a non-MuJoCo environment: an HIV treatment simulator slightly modified from the one in the whynot package. The task simulates the sequential decision making in HIV treatment. We evaluated MOPO with the data generated from the first 200k steps of training an online SAC agent on this environment. We show results in Table 1, where MOPO outperforms BEAR and achieves almost the buffer max score. While the particular choice of $u(s,a)$ that we used in our experiments makes use of the Gaussianity of the dynamics model, this is not a fundamental requirement – one could eschew Gaussian models and estimate model uncertainty some other way, such as model ensemble disagreement (which we tried; see Appendix E).

Buffer Max	**Buffer Mean**	**SAC (online)**	**BEAR**	**MOPO**
15986.2	6747.2	25716.3 ± 254.3	11709.1±1292.1	$\mathbf{13484.6 \pm 3900.7}$

Table 1: HIV treatment results, averaged over 3 random seeds.

R4: *"Try 1) mean variance as compared to max variance for penalizing the reward or 2) disagreement b/w different model predictions"* 1) We added comparison between max variance and mean variance as the reward penalty in the halfcheetah-jump domain. MOPO with max variance achieves **4140.6**±88 average return while MOPO with mean variance achieves **4166.3**±228.8. The two methods did similarly, suggesting that using either mean variance or max variance would be a reasonable choice for penalizing uncertainty. 2) Table 3 in Appendix E of the paper show the results of using model ensemble disagreement without Lipschitz regularization (denoted as MOPO, no Lip, ens. Pen.). It performs similarly to MOPO in D4RL experiments but worse than MOPO on out-of-distribution generalization tasks.

R2: *"intuition for how far the model generalizes?"* We added experiments in Table 2 that show that MOPO generalizes to Ant running at a $45°$ angle (achieving almost buffer max score), beyond the $30°$ shown in the paper, while failing to generalize to a 60 and $90°$ degree angle. This suggests that if the new task requires to explore states that are completely out of the data support, i.e. the buffer max and buffer mean both fairly bad, MOPO is unable to generalize.

R2: *"How were 'true pen.' and 'ensemble pen.' in the appendix computed?"* As explained on line 593-595 in Appendix E, "true pen." is computed as the model prediction error $\|T(s,a) - \widehat{T}(s,a)\|$ using the ground truth dynamics T. The "ensemble pen." measures disagreement among the ensemble: precisely, if the models' mean predictions are denoted $\mu_1, \ldots, \mu_N$, we compute the average $\bar{\mu} = 1/N \sum_{i=1}^N \mu_i$ and then take $\max_i \|\mu_i - \bar{\mu}\|$ as the ensemble penalty. We will make sure these explanations appear prominently in the main paper.

Environment	**Buffer Max**	**Buffer Mean**	**MOPO**
`ant-angle-45`	3168.7	1105.5	2571.3±598.1
`ant-angle-60`	1953.7	846.7	840.5±1103.7
`ant-angle-90`	838.8	-901.6	-503.2±803.4

Table 2: Limit of generalization on `ant-angle`.

R2: *"How did you apply MBPO to the problem?"* As discussed on line 140-149, we first use the full offline dataset to train the model and then use the trained model for model rollouts to optimize the policy. There is no explicit regularization that forces MBPO to stay close to the offline data, but maximizing the expectation over the reward of the trajectories generated from the rollouts of the ensemble model can be viewed as some sort of implicit regularization since the learned model learns the transition dynamics induced by the offline data.

R2: *"It would be nice to compare against something... that relies only on model-rollouts to optimize the policy."* In our experiments, when sampling from the replay buffers, only a small fraction (5%) comes from the real data, and the rest from the model-generated data. For further comparison, we re-ran MBPO with only model-generated data on the D4RL tasks and found that its performance was not significantly affected: no-real-data MBPO outperforms 5%-real-data MBPO on 6/12 tasks and lies within one SD of 5%-real-data MBPO on 9/12 tasks.

R2, R3: *"The practical algorithm is fairly disconnected from the theoretical motivation... The vast chasm between the theory and the actual MOPO?"* We would argue that the theory motivates and justifies the particular way of penalizing the reward using the uncertainty estimates of the dynamics. Indeed, we didn't provide any theory for the uncertainty estimate of the dynamics, but provable uncertainty quantification for nonlinear supervised learning is a major and modular open question in statistics and ML, which is beyond the scope of this paper.

R2: *"A more fine-grained analysis that incorporates the effect that model errors have on the difference in value function would likely lead to more interesting results?"* This is true – certainly $R_{\max}/(1-\gamma)$ is a loose bound. The main difficulty seems to be that without any assumptions on the value function (other than boundedness), the difference could theoretically be arbitrary if the model has any error. If the value function is Lipschitz, we get a bound that involves the 1-Wasserstein distance, which is more fine-grained than total variation distance in the sense that it incorporates the magnitude of error according to the geometry of the state space. However, we do not expect the value function to be Lipschitz in general. A possible strategy would be to use $V^\pi_{\widehat{M}}$, which we can approximate using only samples from the model, to estimate a bound on the difference in V^π_M. We leave this for future work.