We wish to thank the reviewers for their time and thorough reviews. We deeply appreciate the unanimous appreciation of the clarity of the paper and the experiments that we curated to study in detail the strengths of our method, disentangling other possible sources of variation. We believe the main contributions of our work have been understood, but there are some points that we wish to explain in more detail, answering reviewers' comments and requests for clarification.

Originality of the method and motivation for DAE. As Reviewer 1 pointed out, regularization of trajectory optimization is a known technique in optimal control and we agree that we should have explained better the originality of our method and especially the motivation for using denoising autoencoders for regularization.

Our goal is to tackle high-dimensional problems (e.g. Ant has an observation space of 111 and action space of 8, we'll add this information in the Appendix) and expressive models of dynamics which not only means that regularization is needed, as we have demonstrated in the paper, but also that most other regularization techniques will fail. As Reviewer 1 points out, neural networks tend to fare better than many other techniques in modeling high-dimensional distributions. However, using a neural network or any other flexible parameterized model to estimate the input distribution poses a dilemma: the regularizing network which is supposed to keep planning from exploiting the inaccuracies of the dynamics model will itself have weaknesses which planning will then exploit. One is trying to patch the leak with something that leaks itself so let us call this the leaking patch problem.

During the early phases of the research which lead to the present paper, we were experimenting with different input density models including ensembles like those in Ref [20]. All of them suffered from the leaking patch problem. Only DAE avoids the problem and, to the best of our knowledge, nobody else has proposed using it for regularizing trajectories. Clearly DAE will also have inaccuracies but planning will not exploit them because unlike basically any other density models, DAE develops an explicit model of the *gradient* of logarithmic probability density.

How this keeps planning from exploiting the inaccuracies of DAE is easiest to see in gradient-based optimization. Let us assume that a density model (e.g. VAE) has a spurious maximum. If the gradients are calculated by backpropagation, the gradients will point towards this spurious maximum and planning will get there even if the spurious values will represent a fraction of the volume of the whole input space. With DAE, the gradients are obtained from the outputs of the network. There are spurious gradients but gradient directions will not correlate with the directions where the anomalies grow higher. This means that, theoretically, DAE should be a uniquely good solution for regularizing trajectory optimization, and the experimental results we presented support this. It is well known to be very hard to get gradient-based optimization to work with flexible models of dynamical systems and input densities because gradient ascent will immediately expose any inaccuracies of those models. Granted, we did not show that all the alternatives fail. Theoretical justification is nevertheless very important because it gives us some guarantees that the good results will extend to other problems, higher dimensions, and so on. Also, this justification is not limited to gradient-based optimization and indeed we showed that CEM benefits just as well.

Previous works. Works like [30, 31] use model-free RL, and use KL divergence of policies to prevent excessive change in output distribution. This is very different to what we do: we use a model to plan using a powerful optimizer, which can degrade performance if optimizing too much due to adversarial effect (Fig. 3). Similarly, [20] guides the evolution of a policy, while we learn a model that can solve novel goals at test time. [17] uses LTV models, which are faster but also generally too simple for many tasks (see [5]). With respect to (Gauss. Mix. Penalty for Traj. Optim. Problems), we scale to high dimensional problems, with more than 100 state variables.

Use of CEM and re-implementation of PETS. We use CEM for comparison with existing methods (PETS). We used a re-implementation of PETS only for the experiments in Section 4.2 to support gradient-based planning (we use the original implementation for all other comparisons). We will update the paper to show that the re-implementation produces the same results as the original implementation.

Exploration. The presented regularization method does penalize exploration and we could not come up with an exploration strategy that would lead to the state-of-the-art asymptotic performance. Experiments with simple exploration strategies (noise injection) show improved asymptotic performance (as we show in the appendix) but they do not reach the state-of-art. The proposed method can fall into a local optimum and this is what happens in our experiments because we do not reach the state-of-the-art asymptotic performance.

Detailed questions by Reviewer 3. 1) Please see the previous paragraph. 2) The transition function is deterministic in all the tasks we consider. 3) The results reported in Table 1 are obtained by evaluating trajectory optimization with different optimizers and regularizers using *pre-trained* dynamics models (that is, there is no learning process in these results). 4) We show that the DAE regularization is effective with both feed-forward (in mujoco experiments) and recurrent (in industrial process control experiments) neural network dynamics models. We used the same network architecture as PETS to avoid external sources of variations other than the DAE regularization. We are certainly very interested in extending our method to image observations using architectures inspired by the World Models paper, but tackling it in this work is out of scope.