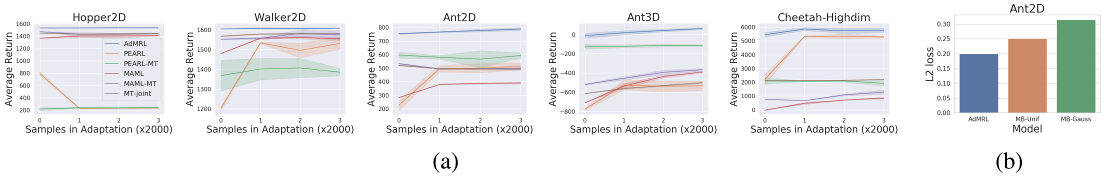

Figure 1: (a) Returns of adapted policies averaged over 3 random seeds. Additional baselines MT-joint, MAML-MT, PEARL-MT are included. Our approach substantially outperforms baselines in adaptation sample-efficiency. (b) The L2 loss of model predictions on OOD `Ant2D` tasks by AdMRL, MB-Unif, MB-Gauss.

We thank the reviewers for the insightful feedback. The reviewers noted that the paper studies "an interesting and relevant setting to meta-RL"[R1] and is "well motivated"[R2], "address the distribution shift issue of meta-RL"[R3], that the idea is "novel" [R2,R4] and "solid"[R4], and that "the results are significant and would be appealing to the wider RL community"[R2]. We will address the major points below and others in the next revision. We respectfully ask the reviewers to consider increasing the scores if our clarification and additional baselines addressed their concerns.

[R1] Q1 (Baselines). A1: We thank the reviewer for suggesting to compare with multi-task policy approaches which also leverage the task parameters explicitly. We didn't compare with them in the paper partly because the original MAML paper [Finn at el.'17] compared MAML with the multi-task policy ("oracle" in Fig. 5 of [Finn at el.'17]) and found that MAML is on par with it. However, we do strongly agree that we should add the comparisons in our paper. We added the experiments (with more details below) as shown in Fig. 1(a) above. Indeed, letting the policy take the task parameters as inputs does not lead to significantly better results, and is still consistently worse than our method. The results also show that multi-task policy is far from "already perfect", a possibility raised by the reviewer, and that MAML and PEARL are also strong baselines for our setting (even though they are not designed to leverage the task parameters). The authors' understanding is that because the optimal policy is a very complex function of the task parameters that cannot necessarily be expressed by neural nets, the multi-task policy is not consistently helpful.

Details. We experimented on three more baselines that use a multi-task policy $\pi(a|s,\psi)$ that takes in the task parameters ψ as inputs. (A) MT-joint: train multi-task policy π jointly on all training tasks. (B) MAML-MT and (C) PEARL-MT: replace the policies in MAML and PEARL by a multi-task policy, respectively. We maintain the number of training samples and tasks. As shown in Fig. 1(a), AdMRL outperforms these baselines consistently, although it is trained on 100X fewer samples than MT-joint and MAML-MT and 3X fewer than PEARL-MT. Note that MAML and PEARL do not explicitly leverage additional task parameters and thus a multi-task policy does not necessarily help them.

[R1] Q2 (Relation to meta-RL). A2: Indeed, our method assumes the knowledge of the task parameters and is different from the standard meta-RL setting, and we will clarify more prominently about it in the paper. However, we also believe that our setting (a) is practically relevant and (b) provides new opportunities for more sample-efficient and robust algorithms. Handcrafted families of rewards functions are reasonable in practical applications, if not common. Moreover, if we don't even know the family of test tasks, it's challenging, if not impossible, to be robust to task shifts in the test time. Our more *restricted* setting makes it possible to be robust to worst-case task shifts. Some intermediate formulations may also be possible, e.g., it's possible to adapt AdMRL to settings where the task family is known in the training time but the task parameters are unknown but inferred in the test time. We leave them as future work.

[R2] Q1 (Oracle access to optimal policy). A1: R2 seems to be mainly concerned with that we assume an oracle access to the optimal policy of a task. We clarify that we do not require such an oracle. Instead, for a training task, we use MBRL to compute the optimal policy ourselves (line 7 of Alg. 1 in the paper) and the optimal performance refers to the convergent post-adaptation performance that the algo. computes. The samples used by MBRL is counted towards the total number of samples. (Actually, PEARL or MAML also have similar components that aim to maximize the return of the adapted policy on training tasks.) we do use an oracle in the final evaluation of the sub-optimality gap. **Q2**: *"...faster adaptation at test-time [with] some other proxy metrics ...?"* **A2**: We did not seriously consider other metrics, but it could be a very interesting direction for future work! One may ideally optimize for the post-adaptation performance, which is challenging and left for future work. **Q3**: *Open-source code.* **A3**: Yes, the authors are committed to open-sourcing the code once the paper is published. We also submitted the code and checkpoints in the supp. material. **Q4**: *The quality of learned models.* **A4**: To measure the model error, we collect samples from true dynamics from OOD `Ant2D` tasks and then evaluate the prediction errors of learned models by L2 loss. As shown in Fig. 1(b), the model learned by AdMRL is more accurate than those learned by MB-Unif and MB-Gauss.

[R3] Q1: *"The effects of each design choice."* **A1**: Most design choices such as conjugate gradient, REINFORCE trick are important as discussed in L198. We expect other MBRL algo. can replace SLBO but we didn't try any. The tuning of hyperparameters is recorded in Appendix D.

[R4] Q1: *Does using advantage function of the learned dynamics incur error?* **A1**: The advantage function is computed by Monte-Carlo estimates of the return—which is accurate because we sample from real dynamics in the training time—minus a parameterized value function—which likely has an error as the reviewer suggested. However, the error does not change the mean of the gradient estimator, because any subtracted "baseline" in the policy gradient estimator that is a function of past states/actions only changes the variance but not the mean of the gradient estimator. **Q2**: *"Is [the comparison with model-based methods] equivalent to Alg 1 without the task parameter optimization?"* **A2**: Yes, model-based methods (MB-Unif and MB-Gauss that we compared with) sample tasks randomly in Alg. 1 (instead of optimizing the tasks). **Q3**: Comparison to multi-task policy. **A3**: Please refer to answer to R1:Q1. **Q4**: "More results on the generalization ability of the method." **A4**: Due to space limit, we didn't include all combinations of evaluation metrics and environments. We will include more in the next revision.