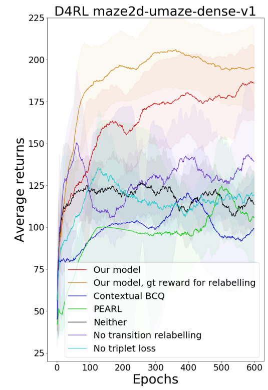
We thank all reviewers for their feedback. Answers to reviews are denoted R2, R3, R5, R6.

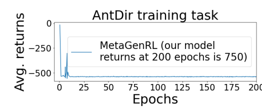
R2: We feel our technical contribution is significant. Since offline data is essentially free for many applications, RL methods should use it whenever possible. This is especially true because practical deployments of RL are bottle-necked by its poor sample efficiency. In particular, results in Sec. 5.3, where we use our policy to initialize an RL algorithm show a substantial gain in performance, even in the complex HumanoidDir environment (64% improvement). As far as we know, we are the first to demonstrate such large gain, using only offline data from other tasks and without knowledge of identity and reward of the test task.

R2: Concerning readability, we will increase figure size.

R2, R3, R5: We performed new experiments. In response to R2, MetaGenRL is designed for online RL while we focus on batch (offline) RL. MetaGenRL relies on DDPG to learn accurate value estimates, which are known to diverge in batch RL (as shown by the BCQ paper). This means that MetaGenRL is not a strong baseline, as confirmed by our experiment in Fig. b, where its performance quickly plummets and does not recover with more training epochs. Combining MetaGenRL and our method would be interesting since MetaGenRL generalizes to out-of-distribution tasks, but is beyond the scope of the paper. As suggested by R3, we add results on D4RL. We didn't know about D4RL when writing the paper (it is a recent preprint), but we ran the experiment on maze2d-umaze now (Fig. a). In this experiment, we train with offline data and evaluate their performance without further training on unseen navigation targets. Our model significantly outperforms the baselines and the ablations. We will provide more analysis on this environment in the paper as R5 suggests.

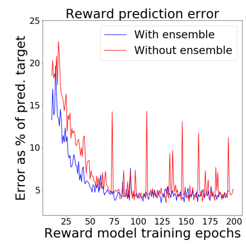


(a) Our model outperforms baselines and ablations.



(b) Meta-GenRL's poor performance even on training task. Results obtain from official MetaGenRL code.

R2, R3, R5: We are happy to extend the related work section and discuss all mentioned papers. Regarding CQL and BEAR, they are single-task Batch RL algorithms and as such are not directly applicable to multi-task Batch RL. We will discuss topics from the deep metric learning paper: embedded samplers, the effect of mini-batch diversity and the correlation between embedding space compression and generalization in RL. Also, since R3 mentions novelty as a relative weakness, we would be grateful if R3 could provide us with more references. The use of the triplet loss in this context is novel and opens up new research directions to determine what the best metric learning loss for RL is.



(c) Error on unseen task.

R6: It is in fact possible to learn a good reward model. Existing model-based RL algorithms necessarily rely on the ability to learn good reward models that generalize. Since the reward function is a mapping from state-action pairs to scalar reward, it is in general much simpler than the task identity function whose inputs are complex high dimensional sets and which maps to a high dimensional embedding space. Moreover, unlike task inference, reward learning can be accomplished for each task independently. Empirically, in Figure c, we show that our reward model indeed achieves low error on state-action pairs from another task, both with and without an ensemble. Moreover, we did an ablation with the ground truth reward you suggested on the D4RL maze2d-umaze environment (Fig. a). While using an oracle for the ground truth reward produces a performance improvement, final performance is close to using our method with learned reward.

R6: We are concerned the original HumanoidDir environment is not suitable as a benchmark for multi-task RL because a single-task policy already obtains high performance on unseen tasks. In particular, we train BCQ with transitions from one task and it obtains a similar return, as measured on unseen tasks, (993 ± 33) to SAC trained from scratch separately for each task (988 ± 19). You are right that we should have used a different name for the environment. We will change the name and show results for both versions in the final version of the paper.

R6: Concerning performance, on AntDir, AntGoal and HumanoidDir, we outperform the best baseline Contextual BCQ by 25%, 26%, 28% in terms of mean return. On those 3 tasks, we outperform the best ablation no_transition_relabelling by 20%, 26%, 14%. Our experiment on D4RL also shows clear improvement over baselines and ablations (Fig a.). On WalkerParam, we agree with your analysis and will clarify in the paper that the performance improvement in WalkerParam comes from distillation. We hypothesize that WalkerParam and HalfCheetahVel do not benefit from reward relabelling because they are lower-dimensional, hence random sampling will lead to lower divergence in state-action distribution compared to higher dimensional tasks.

R6 (other points): We did not use relabeled data to train the critic since we focus on task inference. The connection with structural causal models is an interesting avenue for further work, but beyond scope of this submission. Our method is not specific to BCQ. We will explain it more clearly in the final version. We will explain the tasks in the main text. Finally, while our models are trained from MuJoCo states, they are high-dimensional. In HumanoidDir, the state has 376 dimensions. The task inference model input has 98560 dimensions.