latent dim.	return
1	-10.58 ± 1.27
3	-14.13 ± 1.21
5	-15.41 ± 1.40

Table 1: PEMIRL is robust to various latent dimensions.

method	return
PEMIRL w/o MI	-39.24 ± 3.48
PEMIRL	-14.13 ± 1.21

Table 2: The MI term is important for training PEMIRL.

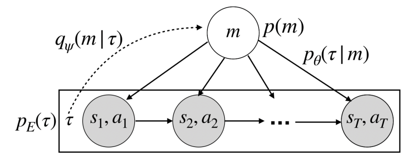

Figure 1: Graphical model underlying PEMIRL

We thank all the reviewers for the constructive feedback. We will incorporate the valuable suggestions in the revised version. We have conducted more experiments and addressed all of the comments below:

To Reviewer #2:

Q1: The importance of mutual information (MI) term? We conducted an ablation study on the MI term with the Point-Maze-Shift environment. The reward function learned without MI failed to induce a good policy in the reward adaptation setting. Results in Table 2 (on the top) demonstrates the importance of our MI term. Theoretically, without MI regularization, the resulting method indeed resembles a VAE. As analyzed in [36], the ELBO of VAE can be interpreted as enforcing consistency between $p(m)p_\theta(\tau|m)$ and $p_E(\tau)q_\psi(m|\tau)$ by minimizing the KL divergence between these joint distributions. Without maximizing the MI between m and τ, a simple degenerate case is $p_\theta(\tau|m) = p_E(\tau)$ and $q_\psi(m|\tau) = p(m)$, which satisfies the consistency constraints, yet completely fails to capture the dependencies between m and τ.

Q2: What if latent dimension is mis-specified? We conducted additional experiments with the Point-Maze-Shift environment (where the ground-truth latent dimension is 3). See the results in Table 1 (on the top). We can observe that PEMIRL with various latent dimension specifications all outperform the best baseline (return -28.61) stably and is hence robust to dimension mis-specifications.

Q3: Performance on a stochastic environment? We create a stochastic version of Point-Maze-Shift (maze size: 60×100 cm) by changing its deterministic transition dynamics into a stochastic one. Specifically, $p(s_{t+1}|s_t, a_t)$ is now realized as a Gaussian with standard deviation being 1 cm. The average return of PEMIRL in reward adaptation is -17.39 ± 0.84, which outperforms the best baseline (average return -30.58) by a large margin.

Q4: Test generalization in more realistic environments? We will add an experiment with a simulated Sawyer robot button pressing task to the revised version, which we were unable to complete during the rebuttal period.

To Reviewer #3:

Q1: Discussion on data efficiency? We would like to clarify that in reward adaptation, we use the inferred reward function to train a policy from scratch rather than finetuning the learned policy. Although efficiency is not the focus of this work, we are happy to provide more discussions on this aspect in the revised version. The sample complexity of PEMIRL at meta-testing phase is comparable to RL training with the oracle ground-truth reward, *e.g.* (PEMIRL vs RL with oracle reward): Point-Maze-Shift: 5.4M vs 4M simulation steps; Disabled-Ant: 15M vs 18M simulation steps.

Q2: Can the tasks also change the dynamics during training? In principle, our algorithm can also handle changes in dynamics during meta-training. We leave this as an interesting avenue for future work.

Q3: The meaning of unstructured demonstrations? As described in line 58-59, "unstructured" means the demonstrations are not grouped according to the task or labeled by task-specific variables. To elaborate, as discussed in line 196-199, previous Meta-IRL methods [12, 32] make simplifying assumptions that each provided expert demonstration contains its corresponding task information (hence "structured"), while PEMIRL has to learn to infer the underlying task corresponding to each demonstration. We will rephrase corresponding parts to clarify it in the revised version.

Q4: Minor comments (1) We will revise the captions to make them more informative. Policy generalization examines if the policy learned by Meta-IL is able to generalize to new tasks with new dynamics. **(2 & 3)** [11, 23] focus on standard IRL and meta-RL respectively rather than Meta-IRL as in PEMIRL. Although [32] focuses on Meta-IRL, their method derivation (*e.g.* Eq 5) requires a tabular MDP. We will rephrase corresponding parts to make this clear.

To Reviewer #5

Q1: Discussion on the efficiency of the proposed method? Although efficiency is not the focus of this work, we are happy to provide more discussions on this aspect in the revised version. During meta-training, for the Point-Maze environment, it takes about 32M simulation steps to converge (similar to other methods such as Meta-InfoGAIL that takes 28M), which amounts to about 2 hours on one Nvidia Titan-Xp GPU; for the Ant environment, it takes about 13.8M simulation steps (Meta-InfoGAIL takes 12M) and about 40 hours on the same hardware (the state-action dimension is much larger than that of Point-Maze). For the sample complexity of meta-testing phase, please refer to the response to Q1 for reviewer #3.

Q2: Graphical model illustration? We will add the graphical model illustration in Figure 1 to the revised version.