Responses to Review Comments (Paper #1720)

We thank all the reviewers for their valuable and helpful comments. Our responses are presented as follows:

Responses to Reviewer #1 1. Weaknesses and Correctness: 1) We believe the novelty of our work lies on that we propose a framework that enables a single DRL agent to achieve expert-level performance on multiple different tasks by learning from task-specific teachers. In general, it is much easier to obtain a set of actors for each learning task; 2) In some scenarios KTM-DRL works better than the ideal solution. Related works [13] (Actor-Mimic) and [14] (policy distillation) also made such observation. One motivating argument from [24] is that the distillation encourages the agent to explore more potential states than teachers, thus leading to better performance ([24] Czarnecki et al., *Distilling Policy Distillation*). 3) In KTM-DRL, we assume a teacher that has good performance on the task will be given. As far as we know, it remains an open question that whether we can learn policies from in-perfect export or sub-optimal policies (Imitation Learning). This problem is quite challenging and is beyond the scope of our discussion, we leave it for future work; 4) While we believe fine-grained tuning to neural network settings (e.g., reducing the number of layers and the number of neurons) may lead to better performance on its task, we just followed up standard settings with previous work [3] (TD3).

2. Clarity, Relation, additional Comments: 1) We follow a typical setting for the DRL problem, which is a MDP; 2) For the term "supervised manner" used in Line 150, we want to emphasize that the DRL agent updates its policy only based on a pre-collected dataset without interactions with the environment; 3) For Equations 1 and 2, "We follow the structure of two critic networks as TD3, to reduce the overestimation bias and improve the stability of the algorithm. In Equ. 1, we aim at mimic the outputs of two critic networks of each task-specific teacher. In Equ. 2, we use one of the critic network of teacher to estimate a precise Q-value, which can provide proper guidance for updating the actor network of the student."; 4) Since we can not find any public implementation of the multi-task DDPG [19], we implemented it and evaluate it by ourselves. However, we found its performance is quite bad. For example, on the benchmark B, the multi-task DDPG [19] only achieves negative rewards on all the tasks. Thus we omit the results of multi-task DDPG; 5) We set the learning rate to 3×10^{-4} for both actor and critic network, the total buffer size is set to 1×10^{6}. We have two independent runs for the training plots. We will narrow down the scope and discuss more specifically in the broader impact section. Besides, we will have a further proofreading and fix the typos.

Responses to Reviewer #2 1. Related works: 1) We thank you for the suggestion to compare with Imitation Learning (IL) methods [1, 2, 3]. But we respectfully disagree with you. The major goal of our paper is multi-task learning, i.e., train a single DRL agent to achieve expert-level performance in multiple different tasks. However, these IL methods focus more on how to leverage demonstrations to effectively learn the control policy for the same task. Specifically, none of [1, 2, 3] on RL+IL are for multi-task with single agent. It is also not trivial to extend existing methods into the continuous-control multi-task DRL setting, which is demonstrated by our experiments (i.e., TD3-MT and SAC-MT can not work well on the multi-task setting). 2) Thanks for your suggestion about the references, we will certainly take a step further to investigate more related works. But it is good to mention that we already include some most related works (i.e., multi-task DRL for continuous control tasks) and compare with them in our experiments.

2. Others: 1) The expert are pre-trained. In general, it is much easier to train an expert on a single task, there are quite a few DRL methods for continuous control that can achieve state-of-the-art performance, like DDPG, TD3, SAC, and PPO. Thus in this paper, we didn't discuss much about how to train the expert on a single task, which is beyond the scope of our discussion. 2) Thanks for your suggestion on wording. We will replace them with more accurate words in the paper and have a further proofreading. 3) We will add more descriptions about the two critic networks in the paper.

Responses to Reviewer #3 We would like to thank the reviewer for the positive comments and encouragement. For the questions: 1) Sorry for the confusion. In the paper, we use the term "epoch" to represents one single interaction with the environment, which has the same meaning of one "step". Thus the x-axis of the learning curves represent the total number of steps. 2) The x-axis in the learning curves **are** cumulative samples across all offline/online learning for KTM-DRL. We will emphasize this in the paper. 3) Thanks for your suggestion. We will add more discussion in the paper. Basically, KTM-DRL actually takes much less time to finish the 1M training compared with some baselines, e.g., it takes 19 hours to finish the end-to-end training, compared with 26 hours for SAC-MT and 28 hours for G-Surgery on NVIDIA GTX 960.

Responses to Reviewer #4 1. 1) Thanks for your suggestion. We had two independent runs and used 10 seeds to evaluate each methods to get the max average rewards. For example, in Table 1, the standard deviation for KTM-DRL, Ideal, TD3-MT, SAC-MT, SharedNet, G-Surgery, and A2C-MT on the task HalfCheetahSmallTorso are 10348 ± 476, 8743 ± 547, 7898 ± 347, 7805 ± 899, 4140 ± 907, 7189 ± 608, 2276 ± 46.14, respectively. 2) Since KTM-DRL learns from task-specific teachers, the performance of the teachers will affect the performance of KTM-DRL. KTM-DRL will fail to find a good control policy if the teachers suffer from performance degradation in their tasks.

2. We will release all the training and evaluation code for the purpose of reproducing the results.