Paper ID 10791
Title: Information-Theoretic Task Selection for Meta-Reinforcement Learning

We thank all the reviewers for their thoughtful feedback. Our response can be found below, organized by review.

R1
"It is not yet clear how results on such simple "toy" tasks will, if ever, generalize to practically important task distributions. But this current limitation does and should not stop progress towards such seminal contributions."

Thank you for the positive comments. We agree that scalability to more complex settings is challenging (more on this in response to Reviewer 3), but this is a challenge for all of meta-RL.

We introduce a method that identifies a clear gap in the literature, and that provides a first solution to the problem, which performs reliably well in a number of current meta-RL benchmarks. We don't expect it to be the last word on the subject, quite the opposite, we hope it will spur new research in its theoretical understanding, and that new meta-RL algorithms will incorporate a task selection component inspired by these results.

In this spirit, we agree with your view that *"It may turn out that some of the assumptions may need approximations and the literal implementation of the heuristic algorithm may be inefficient for larger numbers of more complex task distributions with substantially more complex state spaces. However, ITTS may provide a means of designing better meta-training distributions"*.

R2
"there is no theoretical justification for why the two criteria for task distinctiveness and relevance should work"

We state that the method is a heuristic, and acknowledge in the paper that further work is required on the theoretical side. However, many recent successes in machine learning have been driven by empirical results (most notably, deep learning), and we hope that the introduction of task selection into meta-RL can be one of those.

" there are critical aspects of the experimental methodology that do not appear in the main text, supplement, or code. In particular, for each domain, what are the meta-training (un-pruned and pruned), meta-validation, and meta-test tasks? This information is crucial in helping to elucidate what tasks the algorithm prunes and why, as well as ascertaining the validity of the baseline comparisons."

All the tasks, for training, validation, and test, have been generated according to the distribution provided by the environment. Every run had different tasks, amounting to hundreds of them across all domains. Looking at each one would be impossible, and we do not believe it to be crucial to the validity of the results. It is important that, as long as the tasks are extracted from the domain distribution, task selection improves meta-RL regardless of the particular tasks involved.

"It is hard to assess the correctness of the empirical methodology, as each experiment comprise three main stages (i. learning optimal policies via RL for each of the meta-training and meta-validation tasks; ii. ITTS; iii. meta-RL on the pruned meta-training tasks) yet code is only provided for the second stage."

The code for stage 1 and 3 is not ours, and in the readme file we provide links to the code repositories we used for the domains. For RL (stage 1) and meta-RL (stage 3) algorithms (TRPO, PPO, RL^2, and MAML) we used publicly available code by the respective authors, which can be freely downloaded.

R3

“The authors show their method improving meta-RL test-time performance on a small panel of meta-RL challenges from the literature, however these are all very small "toy" problems which are well-known to be prone to overfitting.”

We did not mean to be limited to “toy” problems, but to use a range of tasks from the literature, whose results we could reproduce. It is a current limitation of state-of-the-art meta-RL algorithms that they have only be applied to such toy problems. Results with one domain could be an overfit, but we think that consistently good results over 6 domains can convince the reader that the benefits of ITTS are not the effect of the algorithm overfitting to any particular domain.

“The simulated robotics tasks in particular just do not exhibit enough structural diversity to evaluate the applicability of this method to real robotics problems, or even more complex simulated robotics problems”

This criticism is understandable, but should be directed at the original publications that proposed those domains. Again, we intended to reproduce and improve upon published results, to make clear that we did not design or use a particular domain because it exhibits the task-selection benefit, but rather that it is widespread in meta-RL, and already present in published domains.

“Existing benchmarks exist ([1][2]) which offer much more diversity than the environments used by the authors. The authors even cite [2]. Using better benchmarks would provide the reader with more confidence in the extent to which these limited-scope empirical experiments might extend to her desired application domain.”

We did our best to use as many domains as possible with the required characteristics (access to code, programmatic change of parameters to create the task distribution) and representing a range of challenges in meta-RL, particularly focusing on sparse rewards (Krazy World and MiniGrid) and continuous control (the two locomotion tasks). We also introduced an application-inspired domain, to demonstrate a practical use of ITTS and meta-RL.

[1] was introduced in the multi-task learning context, and we could not find results with meta-RL algorithms on it that we could reproduce and improve upon. If we missed results on such a domain with RL^2, MAML, or other meta-RL algorithms that the reviewer is aware of, we would be grateful if they could point us to them.

[2] was a good candidate, and we did consider it. It did not make the final selection because it is at the edge of what meta-RL can achieve, and current algorithms struggle with this domain. Indeed the authors say “Our experiments show that current meta-RL methods in fact cannot yet generalize effectively to entirely new tasks and do not even learn the meta-training tasks effectively when meta-trained across multiple distinct tasks.” We would have to use the subset of the domain in which current meta-RL algorithms perform well enough, attracting the same criticism the reviewer is making.

As reviewer 1 mentioned, *“This paper shares the weaknesses of other meta-RL papers and benchmarks. It is not yet clear how results on such simple "toy" tasks will, if ever, generalize to*

practically important task distributions." We agree with this view, in that we do share current limitations of meta-RL. We do not introduce ITTS to demonstrate that, only thanks to task selection, current meta-RL algorithms can generalize to much harder domains. We believe that more development will be required on of meta-RL algorithms. However, such future developments should take task selection into account, and ITTS provides a solution applicable to current domains, and a baseline for future work pushing the boundary of meta-RL applicability.

*"By my reading of Algorithms (1,2) suggest that the time/sample complexity of the proposed algorithm is somewhere in the neighborhood of O(T^2 * train()) or O(T^3 * train()), where T is the number of meta-test environments and train() is the amount of time (or number of environment samples) necessary to train an agent on a single instance*
of t \in T"

The agent is trained once on every training task, with a cost of O(T * train()). This happens before Algorithm 1 is executed, so that it has access to \pi^*_t for every t \in T. The computation of \ delta_c is O(T^2) since the difference is computed T*C times, and C is a subset of T. This computation, however, is the KL divergence of the policies on the samples of the validation tasks, and does not depend on the training time of the tasks. It is proportional to the number of states used for the estimate of the KL divergence, which is up to the user, with a more accurate estimate requiring more samples. The last step is the computation of relevance which requires O(T*F*l) steps, where |F| << |T| is the set of validation tasks. The parameter l is the number of episodes for which the transfer policy is learned, to give an estimate of the speed-up that transferring from that training task gives. This value is, again, much lower than the number of episodes required to converge to the optimal policy (the value of all parameters used in the experiments is in the supplementary material).

The complexity is dominated by the initial O(T*train()) computation, that is, by learning the optimal policy for all training tasks. Even in complex tasks, where the estimate of the KL divergence may require a large number of samples, learning the optimal policy for those tasks will still be significantly more expensive. For complex training tasks this is a significant limitation, but we believe that it can be accelerated, for instance not learning each training task from scratch as we did in this paper, but using transfer learning, or curriculum learning. However, we wanted these results to be independent of specific optimizations which may be required on complex domains.

To show the immediate applicability of the method, we also introduce the MGEnv domain, which is simple enough to allow the application of ITTS as presented (without further optimizations, for instance on learning the optimal policies) but is a realistic application scenario, with real data of energy generation and consumption. Admittedly, learning policies for micro grid control is less complex than many robotics tasks, but still economically significant. The simulation of the learned policies uses data of real buildings, and is as close as possible to actually running the simulated device in that building, at that time (data are from January 2016 to December 2018; the PecanStreet database is publicly available).

We would also like to thank the reviewer for the suggestions on clarifying figure naming, adding an "oracle" line, adding option papers to the related work section, and for catching the missing citation of the garage library. The last point, in particular, was indeed an oversight on our part, for which we apologize. We'll incorporate the suggestions in the next version of the paper.

R4
"Though the proposed method seems effective and reasonable in general, more details regarding the problem setup can help better understand the performance of the algorithm and improve reproducibility of the work."

We would be grateful if the reviewer could elaborate on what details they feel are missing, so that we can improve the paper. We answer the questions in Section 8 of the review below, hoping that this provides all the required clarification.

“In addition, comparing the difference between each pair of the training tasks seems expensive. Some discussions about the scalability of the method could be helpful.”

Please see our response to Reviewer 3. The main computation bottleneck is learning the optimal policies for all training tasks. This limitation is discussed in the paper, and is the main barrier to the applicability of the method. Whether or not all training tasks can be learned in a reasonable amount of time, from the user’s perspective, is indeed domain-dependent.

“1) What is the intuition behind using entropy for measuring relevant, in contrast to using kl divergence between the two policies?”

We did try the KL divergence for that comparison as well, but it does not lead to results as good as entropy. The problem is that the learned policy may differ substantially from the transfer policy, while not being the result of any useful learning. For instance, if the transfer policy is not useful in the target task, this may lead to catastrophic forgetting, and the learned policy degenerating to close to random exploration. This effectively corresponds to erasing all transferred knowledge and starting from scratch, which is not desirable, but gives a high KL divergence. However, if learning leads to a decrease in entropy, the learned policy is “sharper” than the transfer one in making decisions, which indicates progress towards an (at least locally) optimal policy. A reduction in entropy is a good indication of transfer being beneficial, corresponding to an information gain.

“2) How are validation and testing tasks selected in the experiments? It is mentioned that the training tasks are sampled uniformly in certain range, but it’s not clear how the validation and testing tasks are selected. Are they from the same distribution that is far away from the training tasks? It would be nice to study how the performance varies when the distribution of training, validation, and testing tasks have different distances.”

We assume the standard meta-RL framework, with only one task distribution. All tasks are generated by sampling from this distribution. For each domain we specify which parameters vary among the tasks, giving raise to the task distribution for that domain.

“3) What is the baseline ‘all’? Does it contain the validation set? If not, why does having relevance or difference alone hurt the performance (Figure 2)? How would it perform if the meta-training is performed for the union of training and validation tasks?”

The baseline “all” is all the training tasks T. This name confused another reviewer, so it is clear that we need a better label for it. It does not contain the validation tasks. Training and validation tasks are extracted from the same distribution, so the union of both is just a larger training set from that distribution. We already show that larger training sets are not necessarily better. The use of the validation set for training demonstrates that those tasks are not “special” or more informative.

“4) In most tasks it seems that the learned policy before adaptation is already better than the baseline methods, which is a bit surprising. Are there any intuitions for that?”

RL^2 does not perform online learning, but the first few episodes are used to fill in the memory of the recurrent neural network with the new context. The network has been entirely learned during meta-learning, so it is perhaps less surprising. For what concerns MAML there is indeed learning in

the test tasks, and the policy learned with ITTS does have a better starting value. We do not have any particular intuition about why this is the case. We can only note in the results that the learned policy does indeed generalize better to new tasks from the first episode.