Thesis Description

# Background:

In RL, a task () is defined as an MDP ():

Task = or = {}

Conventionally, an RL-agent would have to be re-trained from scratch for each change in the task it was originally trained for. Not only could this be very time-consuming, but also collecting large amounts of data could be impractical or inefficient as it could be not possible or expensive in reality.

Meta learning is an approach where the model distills experience from training on a set of tasks (called “meta-training” tasks) drawn from the same distribution as the downstream tasks expected to be encountered at test time, and leverages this experience to quickly and efficiently adapt to changes in these downstream tasks, and have better generalization capabilities. This distilled knowledge is in the form of parameters called “meta-parameters” which could be e.g. hyperparameters, initial conditions (i.e. priors), exploration policy, optimizer, loss/reward, etc.

Meta reinforcement learning (meta-RL) extends meta learning to the RL setting. For example, the meta-parameters could be priors for the leaned policy or dynamics/transition model which serve to adapt quickly and efficiently to changes in the environment and/or reward function.

In the scope of this document, I define a semantic factorization of the Task: , where:

Environment = {

* Reward function = R =

In this sense, environment-based tasks could refer to settings of varying dynamics or agent parameters. This setting is notably relevant when training a model in simulation to be used later on real robots, since the agent parameters can often vary in reality from those specified in simulation, causing naïve policies trained in simulation to fail when transferred directly. This is the problem setting of “Sim-to-Real Transfer” for which several approaches have been proposed, including varying the agent parameters during training to train robust policies, an approach known as domain randomization.

Finally, meta training tasks has traditionally been sampled uniformly from the given task distribution. Recent research has shown that more sophisticated ways of sampling from the distribution could lead to improvement of the performance. One example is active domain randomization [1], which samples from the given parameter ranges in a way that lets the agent train longer on more difficult environments, and gradually move from easier ones to difficult ones.

# Thesis Title:

Meta Reinforcement Learning with Active Domain Randomization for Sim-to-Real Transfer

# Problem Statement:

In this thesis, I want to address the problem of improving the performance of learned policies for sim-to-real transfer.

To achieve this goal, I want to use a meta-learning approach to extract/learn useful, adaptable priors from the environment and combine it with active sampling of the randomized environment parameters from their predefined ranges.

# Research Questions:

1. Meta-RL so far has relied on uniform sampling from task distribution (including, uniform domain randomization (UDR)). Would combining meta-RL with active domain randomization (ADR) instead yield better performance over the randomization ranges and/or generalization of the meta-RL algorithm in the setting of changing dynamics and fixed rewards?

**Hypothesis**: Since ADR establishes some notion of a curriculum in sampling from the randomization domain and that it has been shown to be superior to UDR when both use a standard/core RL algorithm, combining meta-RL with ADR instead of UDR will also yield similar improvements to the meta-RL algorithm in terms of its performance over the randomization ranges and generalization in the setting of changing dynamics and fixed rewards.

1. Active domain randomization so far has been used with standard/core RL algorithms to train a robust policy. Would combining it with a meta-RL algorithm instead, to train an adaptable policy, yield better performance over the randomization ranges and/or generalization in the setting of changing dynamics and fixed rewards?

**Hypothesis**: Adaptive policies (learned by meta-RL algorithms) are at least as good as, and in many cases superior to, robust policies (learned by standard RL-algorithms) in terms of performance over the randomization ranges in the setting of changing dynamics and fixed rewards, since the meta-parameters of the meta-RL algorithms could be seen as comparable to the robust policies’ parameters, but additionally the perform a fine-tuning step to adapt to the current context which would yield better performance.

1. In the setting of changing dynamics and fixed rewards, so far, the meta-RL algorithms have been model-based. Would using a model-free meta-RL algorithm yield comparable results in terms of performance and generalization?

**Hypothesis**: Since the [optimal] policy is affected by changing dynamics, policy-based/model-free meta-RL algorithms would produce a comparable performance over the randomization ranges and generalization to model-based meta-RL algorithms in the setting of changing dynamics and fixed rewards.

# Workplan:

Desired/Estimated end date: 1/5/2022

|  |  |  |  |
| --- | --- | --- | --- |
| Objectives | Steps | Details | Status |
| Main Objectives | Literature Review | Excel Sheet detailing reviewed papers and choosing ones of interest | Done |
| Implement Meta-RL | MAML algorithm | Done |
| Implement Domain Randomization (DR) | UDR & ADR | Done |
| Milestone 1: Combine Meta-RL + DR | MAML + ADR/UDR | Done |
| Milestone 2: Run Sim2Sim Experiments | Experiments with algorithms on Mujoco envs with varying dynamics | In Progress |
| Run Sim2Sim experiments on simulation of real test env | - | Not started |
| Milestone 3: Run Sim2Real Experiments | Experiments with algorithms on real robot env with varying dynamics | Not started |
| Thesis Finalization | Write Thesis Document & Prepare Presentation | - | Not started |

# References:

[1] Mehta, B., Diaz, M., Golemo, F., Pal, C.J., & Paull, L. (2019). Active Domain Randomization. CoRL.

# Appendix:

# Problem dimensions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task | | | | Data | | | Reward | | Environment | | | | | | Phases | |
| IDs/boundaries | | Source | Distribution | Distribution | Source | | Params | Transition dynamics model | | | observability | State & action spaces |
| Known / defined | discrete | given | i.i.d. | stationary | Trial-&-Error | Demos | Known /dense | Fixed | Fixed | Known | | Deterministic | Full obs. | discrete | 2 phases (fixed) | |
| Semi-known | Continuous | Partially given | predictable | Partial non-stationary | Semi-known / sparse | Changing | Changing | Unknown, unlearned | Unknown, learned | Stochastic | Part obs. | Continuous  (or Mix) | Lifelong, 2 phases (interleaving) | Lifelong, online |
| unknown | curriculum | Full non-stationary | Unknown / non-existent or very sparse |
| Not given | adversarial | Mix | |
| OOD |

|  |  |
| --- | --- |
|  | Train |
|  | Test |
|  | Both |

# Desiderata / Objectives:

|  |  |
| --- | --- |
| Asymptotic Performance | Effective/structured exploration |
| Consistency | Computational, memory and time resources |
| Stability &/or Convergence  (related: Reliability / reproducibility) | Safety |
| Data/sample efficiency | Uncertainty awareness |
| Expressivity | Feasibility / Realizability |
| Robustness | Privacy |
| Learning/Adaptation speed and ability | Interpretability / Explainability |
| Transferability / Portability & Generalization | Complexity |
| Jump-start performance |  |

|  |  |
| --- | --- |
|  | Primary |
|  | Secondary |