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, in model-based meta-RL, the meta-parameters are priors for the dynamics/transition model which serve to adapt quickly and efficiently to changes in the environment and reward function. Moreover, research has recently been done to make meta-RL even more autonomous by making the agent acquire the meta-training tasks on its own without having to require them to be provided manually (i.e. acquire them in an unsupervised way), in a setting named unsupervised meta-RL.

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

Environment = {

* Reward function = R =

# Preliminary title suggestion:

Unsupervised Model-based Meta-RL

# Problem Statement:

In this thesis, I want to address the problem of autonomously learning a consistent, optimal policy for each new & generally different downstream task in a fast and data-efficient way.

To achieve this goal, I want to use a meta-learning approach to extract/learn useful priors from generally continuous and high dimensional or partially observable environments and combine it with unsupervised task acquisition for automating meta-tasks design.

# Research Questions:

## Main Objectives:

1. Could automatic/unsupervised task acquisition be combined with model-based meta-RL in a way that maintains benefits of both worlds? How?  
     
   **Hypothesis**: Combining Unsupervised Meta-learning with Model-based Meta-RL would simultaneously address the problems of:
2. Better adaptation to changes in tasks. Changes in tasks could be split according to:

* Changes in the reward function: Model-based RL methods would be useful in this case because they are more data-efficient and reusable.
* Changes in the environment: Model-based Meta-RL (i.e. learning dynamics model priors) would be useful for this because it enables fast adaptation to changes in the environment.

1. Increased autonomy in learning, because meta-learning combined with unsupervised task acquisition automate the burdens of manual algorithm and tasks design, respectively, for better adaptation to downstream tasks/environments.

Diagram

Description automatically generated

Figure 1: Unsupervised Model-Based Meta-RL. Adapted from [1] & [6]

1. Could additional priors be learned jointly over the proposed tasks? How? And how useful are they w.r.t. adapting to downstream tasks quickly and efficiently? (e.g. for Embedding/Latent representations, Rewards, Exploration policy and/or Exploitation/Execution policy)  
     
   **Hypothesis**: Learning more priors, jointly with the dynamics model prior, would address the problem of pushing further the boundaries of efficiency, learning speed, performance and generalization capabilities of meta-RL, because:
2. Exploration policy priors would allow to learn more accurate models efficiently, which leads to even faster and more efficient adaptation to potentially more different downstream tasks.
3. Embedding/representation priors would lead to learning quickly more compact and informative task representations.
4. Rewards priors for learned skills would speed up learning by selecting for only valuable skills based on their usefulness for downstream tasks.

## Stretch Objectives

1. Does learning a factorized dynamics model into a split causal and natural dynamics help the agent to adapt better to new tasks and environments?  
     
   **Hypothesis**: Learning a split dynamics model would address the problem of quick and efficient adaptation to potentially OOD downstream tasks because the factorized dynamics would allow the agent to reason about invariant and causal structures of the tasks
2. How can the solution be extended to continual/lifelong/online learning setting while additionally minimizing negative and maximizing positive transfer?  
     
   **Hypothesis**: Extending the solution to lifelong learning setting addresses the problem of making learning more autonomous and the solution more feasible (i.e. more fully applicable to the real world) where tasks are learned and performed together continuously.
3. Does active curriculum design at meta-training time help the agent to adapt better to different and difficult/complex downstream tasks?  
     
   **Hypothesis**: By allowing the agent to determine and optimize for the order and difficulty in which tasks are encountered, the agent would be more proficient and efficient at learning more complex downstream tasks.

# Preliminary Workplan:

## Literature Review is almost done

## Milestones:

Estimated time for each activity is 1 month with an estimated total duration of 9 months (1/8/2021 – 1/5/2022)

|  |  |  |  |
| --- | --- | --- | --- |
| Phases | Steps | Output | Activities |
| Study | Reading | Summary of read topics | Learn about model-based RL & meta-RL in more depth |
| Learn about Information-theoretic view & relevant deep generative models |
| Learn about GVFs, options, skills & aux tasks |
| Implementation | Implementation and demo of algorithms on a toy RL problem (e.g. from Mujoco, like Half-cheetah, ant, etc) | Core & PG: PPO, TRPO, SAC |
| Model-based: PETS [7] |
| Meta-RL: |
| Work | Reading | Summary of read topics & written “related works” section | Reading relevant papers in detail [1], [2], [3], [4], [5], etc |
| Implementation | Implementation and demo of algorithms on a toy RL problem | Implement unsupervised meta-RL (e.g. DIAYN) [1] |
| Implement model-based meta-RL (GrBAL and/or ReBAL) [2]. |
| Combine both methods |
| Implement other priors |
| Writing | thesis documents and resources | Write and finalize thesis |

# References:

[1] Gupta, A., Eysenbach, B., Finn, C., & Levine, S. (2018). Unsupervised Meta-Learning for Reinforcement Learning. ArXiv, abs/1806.04640.

[2] Nagabandi, A., Clavera, I., Liu, S., Fearing, R., Abbeel, P., Levine, S., & Finn, C. (2019). Learning to Adapt in Dynamic, Real-World Environments through Meta-Reinforcement Learning. arXiv: Learning.

[3] Hsu, K., Levine, S., & Finn, C. (2019). Unsupervised Learning via Meta-Learning. ArXiv, abs/1810.02334.

[4] Eysenbach, B., Gupta, A., Ibarz, J., & Levine, S. (2019). Diversity is All You Need: Learning Skills without a Reward Function. ArXiv, abs/1802.06070.

[5] Sharma, A., Gu, S., Levine, S., Kumar, V., & Hausman, K. (2020). Dynamics-Aware Unsupervised Discovery of Skills. ArXiv, abs/1907.01657.

[6] Finn, C. (2020) Meta Reinforcement Learning: Adaptable Models & Policies [PowerPoint slides]. Computer Science Department, Stanford University. <https://cs330.stanford.edu/slides/cs330_metarl1_2020.pdf>

[7] Chua, K., Calandra, R., McAllister, R., & Levine, S. (2018). Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. NeurIPS.

# Appendix:

## Dimensions:

### Problem:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Tasks ID/boundaries | | Train Tasks | Task distribution | Data distribution | | Data Source | Task Labels/rewards | Environment | | | Setting |
| Known / defined | discrete | given | i.i.d. | stationary | discrete | Data is Given | Known/dense | same | Known model | Full obs. | Offline / 2 phases |
| Semi-known | Continuous | Not given | predictable | Partial non-stationary | Continuous  (or Mix) | Trial-&-Error  (Active exploration / data collection) | Semi-known / sparse rewards | different | Unknown, unlearned | Part obs. | Online / continual |
| unknown | curriculum | Full non-stationary | Demonstrations | Unknown / non-existent or very sparse | Unknown, learned |
| adversarial | Mix |
| OOD |

### Desiderata:

|  |  |
| --- | --- |
| Performance | Effective/structured exploration |
| Consistency\* | Computational and time resources |
| Safety, Stability &/or Convergence | Reliability / reproducibility |
| Data/sample efficiency | Uncertainty awareness |
| Expressivity\*\* | Feasibility / Realizability |
| Robustness | Privacy |
| Adaptation speed and ability | Interpretability / Explainability |
| Transferability & Generalization | Complexity |

|  |  |
| --- | --- |
|  | Main Objectives |
|  | Stretch Objectives |

\* The learned learning procedure will solve the task given enough data at test time. Important because it reduces reliance on meta-training tasks & gives good OOD performance

\*\* Expressive power: the ability to represent a range of learning procedures. Important because it helps with scalability, applicability to a range of domains, etc. E.g. NNs are expressive.