Probabilistic Active Meta-Learning

Jean Kaddour*, Steindór Sæmundsson*, Marc Peter Deisenroth

*Equal contribution

UCL · Imperial College London

Motivation

- Meta-Learning takes advantage of prior experience in a domain to learn new tasks efficiently
- Training tasks are often given or randomly chosen
- When training a model from scratch in real-life: **how do we collect training tasks data-efficiently?**

Background: Probabilistic Meta-Learning

- Meta-Learning deals with task-specific datasets $\mathcal{D}_{\mathcal{T}_i} = \{(\boldsymbol{x}_j^i, \boldsymbol{y}_j^i)\}$ corresponding to tasks $\mathcal{T}_i \sim p(\mathcal{T})$
- We can model the task specification by means of a latent variable $\boldsymbol{h}_i$ distinct from global model parameters $\boldsymbol{\theta}$, which are shared among all tasks

$$p(\mathbf{Y}, \mathbf{H}, \boldsymbol{\theta} | \mathbf{X}) = \prod_{i=1}^{N} p(\boldsymbol{h}_i) \prod_{j=1}^{M_i} p(\boldsymbol{y}_j^i | \boldsymbol{x}_j^i, \boldsymbol{h}_i, \boldsymbol{\theta}) p(\boldsymbol{\theta}),$$

  where $\mathbf{H}$ collects the task-specific embeddings
- At test time we are faced with an unseen task $\mathcal{T}_*$ and our aim is to use the meta-model to make predictions $p_{\boldsymbol{\theta}}(\mathbf{Y}_* | \mathbf{X}_*)$

Extending the Meta-Learning Model

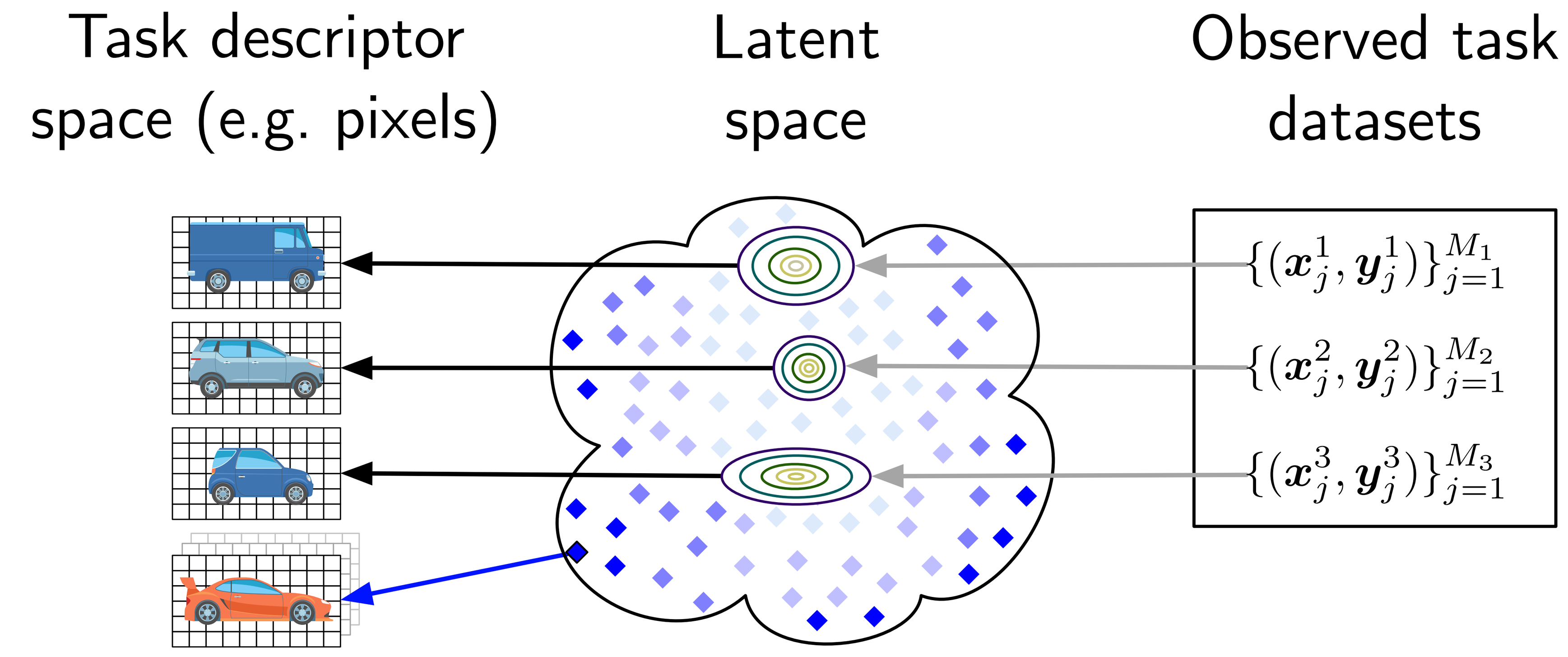
- We learn the relationship between $\boldsymbol{h}$ and $\boldsymbol{\psi}$,

$$p_{\boldsymbol{\theta}}(\mathbf{Y}, \mathbf{H}, \boldsymbol{\Psi} | \mathbf{X}) = \prod_{i=1}^{N} p_{\boldsymbol{\theta}}(\boldsymbol{\psi}_i | \boldsymbol{h}_i) p(\boldsymbol{h}_i) \prod_{j=1}^{M_i} p_{\boldsymbol{\theta}}(\boldsymbol{y}_j^i | \boldsymbol{x}_j^i, \boldsymbol{h}_i),$$

  where $\boldsymbol{\Psi}$ denotes a matrix of task-descriptors $\boldsymbol{\psi}_i$
- Maps latent embeddings to task-descriptor space to generate/choose new tasks

Key idea

Training task selection based on prior experience



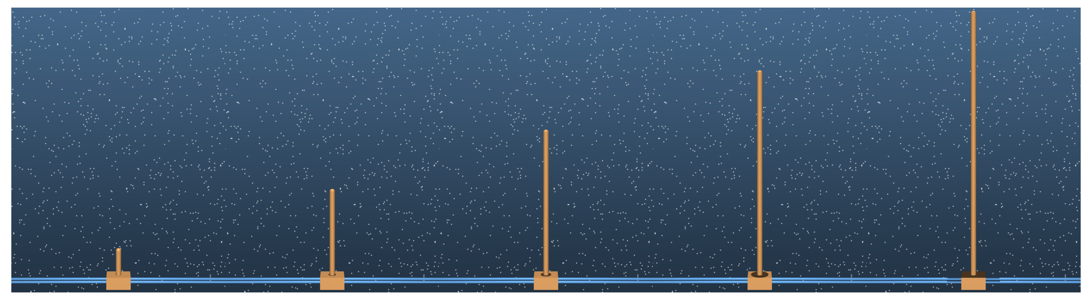
- Infer latent task embeddings (Gaussian-shaped distributions) of observed tasks
- Learn mapping from latent to the task descriptor space
- Rank candidate tasks (diamonds) in the latent space by quantifying their utility (the higher, the darker)
- Select the candidate task with the highest utility

Algorithm

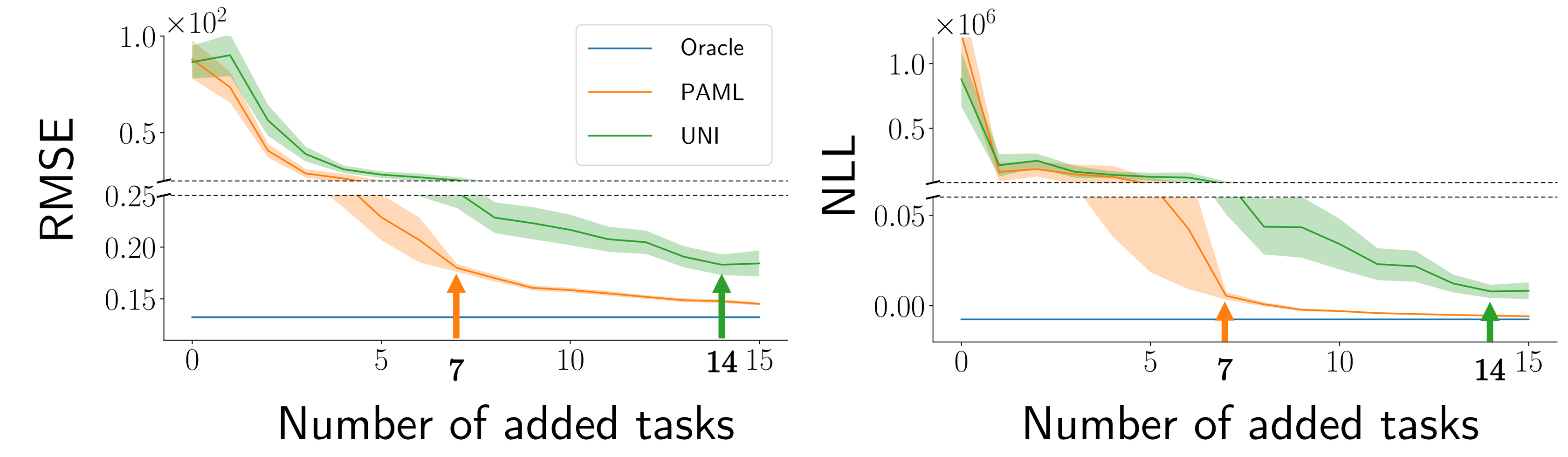
input: Task descriptors (distribution $p(\boldsymbol{\psi})$ or fixed set $\{\boldsymbol{\psi}_i\}_{i=1}^{N}$), active meta-learner $\{p_{\boldsymbol{\theta}}, q_{\boldsymbol{\phi}}\}$, utility function $u(\cdot)$ and N_{init}
Sample initial Ψ_{init} and task datasets $\mathcal{D} = \mathcal{D}_{\text{init}}$
while meta-training **do**
    Train active meta-model $p_{\boldsymbol{\theta}}$ and infer task embeddings $q_{\boldsymbol{\phi}}(\mathbf{H})$
    Select candidate $\boldsymbol{\psi}^*$ by ranking in latent space $\boldsymbol{\psi}^* = \operatorname{argmax}_{\boldsymbol{h}_*} u(\boldsymbol{h}_*)$
    Observe new task $\mathcal{D}_{\boldsymbol{\psi}^*} \sim p(\boldsymbol{y} | \boldsymbol{x}, \boldsymbol{\psi}^*)$
    Add new task to dataset $\mathcal{D} = \mathcal{D} \cup \mathcal{D}_{\boldsymbol{\psi}^*}$
end while

Experiments with Pixel Task Descriptors

- Measure model's performance on test tasks as a function of tasks added by each method
- Baselines: Uniform sampling (UNI), Oracle
- Tasks: Learning dynamics of robotic environments
- Only access to pixel descriptors, e.g., images of cart-pole systems with varying lengths



Results



~50% reduction in added tasks to achieve the same performance

Related work

- Probabilistic Meta-Learning
  - Sæmundsson et al. "Meta reinforcement learning with latent variable Gaussian processes" (2018)
  - Gordon et al. "Meta-learning probabilistic inference for prediction" (2019)
- Automatic Curriculum Learning
  - Portelas et al. "Automatic curriculum learning for deep RL: A short survey" (2020)
  - Jabri et al. "Unsupervised curricula for visual meta-reinforcement learning" (2019)
- (Automatic) Domain Randomization
  - Akkaya et al. "Solving Rubik's cube with a robot hand" (2019)
  - Mehta et al. "Active domain randomization" (2020)

Code

https://github.com/JeanKaddour/PAML