Trajectory representation learning for Multi-task NMRDP Planning

Firas JARBOUI

Vianney PERCHET

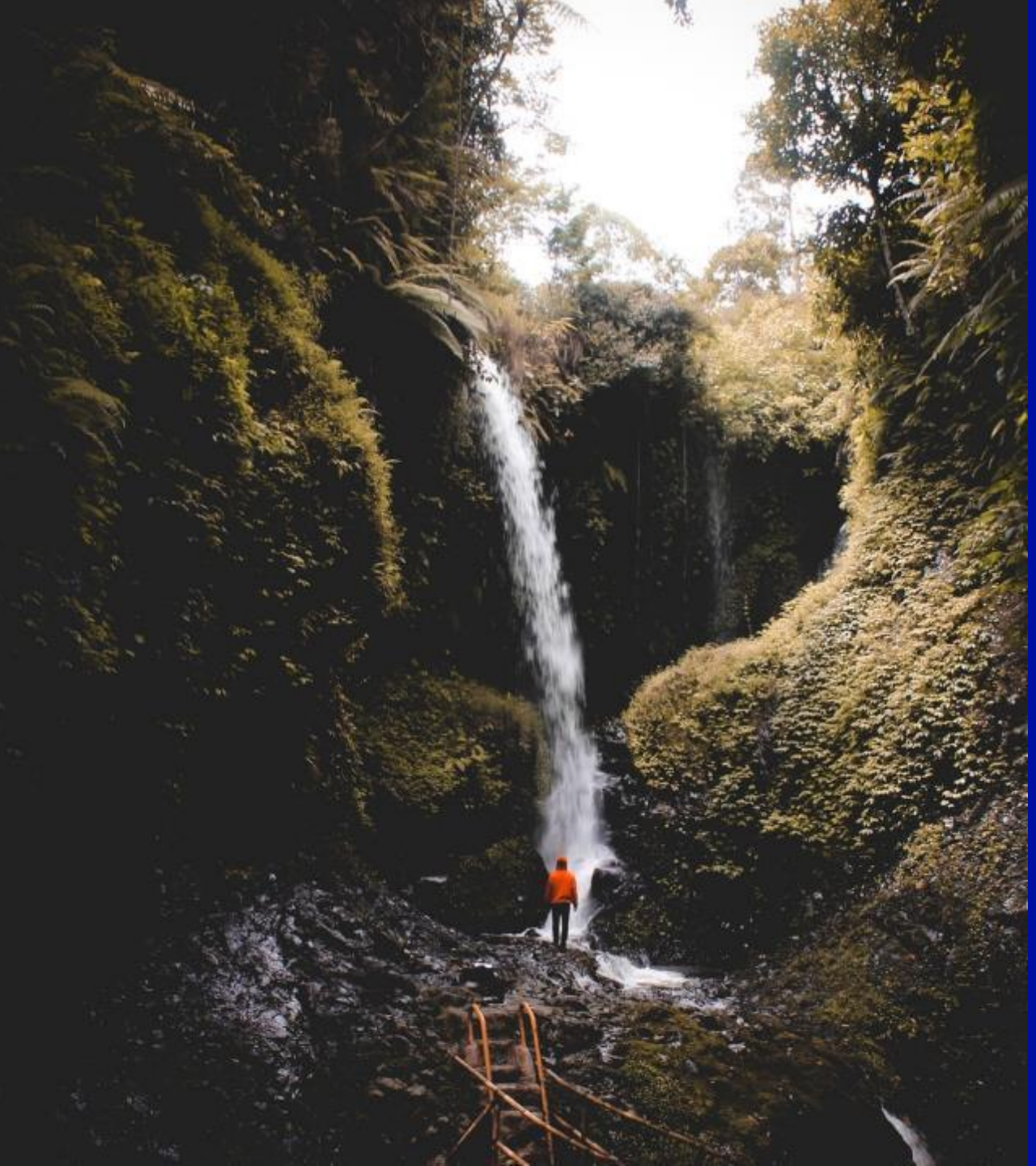
aneo

école normale supérieure paris-saclay

université PARIS-SACLAY

Summary

1. What are NMRDPs?
2. How do we solve NMRDPs?
3. Does it work in practice?



1 NMRDPs: a Non Markovian Reward Decision Process framework

What are NMRDPs?

A tuple {O,A,R,P,ɣ} where:

O: Observation space
A: Action space
R: Reward function
P: Transition probabilities
ɣ: discount factor

BUT

R maps **trajectories** Γ(O) into rewards
(rather than observations as in MDPs)

NMRDP vs Multi-task MDP

In Multi-task MDPs:

An agent is optimal with respect to each task

In NMRDPs:

An agent is optimal with respect to a **sequence** of tasks

Optimally collecting wood OR stones ≠ Optimally collecting wood THEN stones

2

Solving NMRDPs

Standard approach?

Solving NMRDPs in the general case require specific domain knowledge to build the equivalent MDP

For example, 'key observations' that can lead to a change in the reward function.

The optimal construction of the equivalent MDPs relies heavily on combinatorial schemes.

A Subset of NMRDPs

We consider NMRDPs where:

$$\mathcal{R}((o_t)_{t=0}^{t=T}) = \mathcal{R}(o_T, \mathcal{T}((o_t)_{t=0}^{t=T})) = \mathcal{R}(o_T, h_T) \quad \forall\, T \in \mathbb{N}.$$

$$\mathcal{T}((o_t)_{t=0}^{t=T}) = \mathcal{T}((o_{T-\tau})_{\tau=0}^{\tau=\tau_O}, (h_{T-\tau})_{\tau=1}^{\tau=\tau_T}) \quad \forall\, T \in \mathbb{N}.$$

In particular this coincides with:

$$\mathcal{T}((o_t)_{t=0}^{t=T}) = \mathcal{T}(o_T, h_{T-1}) = h_T \quad \forall\, T \in \mathbb{N}.$$

Solving NMRDPS

$\simeq$

Constructing an equivalent MDP

+

Solving the new MDP

A Subset of NMRDPs

In this specific case, the equivalent MDP can be defined as follows:

$$\mathcal{M}^* = \{\mathcal{S}^*, \mathcal{A}^*, \mathcal{P}^*, \mathcal{R}^*, \gamma\}$$

Where

$$\mathcal{S}^* = \mathcal{H} \times \mathcal{O}$$

And

$$\begin{cases} \mathcal{A}^*(o_t, h_t) = \mathcal{A}(o_t) \\ \mathcal{R}^*(o_t, h_t) = \mathcal{R}(o_{0:t}) \\ \mathcal{P}^*((o_{t+1}, h_{t+1})|(o_t, h_t), a_t) = \\ \quad \mathcal{P}(o_{t+1}|o_t, a_t) \times \mathbb{1}_{h_{t+1}=\mathcal{T}(h_t, o_{t+1})} \end{cases}$$

All these quantities are tractable except the trajectory representation function T and the latent space H

Solving NMRDPS

$\simeq$

Constructing an equivalent MDP

+

Solving the new MDP

Relaxing the equivalent MDP

We propose to construct the equivalent MDP using a feature space instead of the latent space H and a trajectory embedding

$\phi : \Gamma(\mathcal{O}) \to \mathbb{R}^d$ instead of T

Let $\hat{\mathcal{M}} = \{\hat{\mathcal{S}}, \hat{\mathcal{A}}, \hat{\mathcal{P}}, \hat{\mathcal{R}}, \gamma\}$ where:

$$\hat{\mathcal{S}} = \mathbb{R}^d \times \mathcal{O} \supseteq \phi(\Gamma(\mathcal{O})) \times \mathcal{O}.$$

$$\begin{cases} \hat{\mathcal{R}}(o_t, \phi_t) = \mathcal{R}^*(o_t, \mathcal{C}(\phi_t)) = \mathcal{R}^*(o_t, h_t) \\ \hat{\mathcal{P}}((o_{t+1}, \phi_{t+1}) | (o_t, \phi_t), a_t) = \\ \qquad \mathcal{P}^*((o_{t+1}, \mathcal{C}(\phi_{t+1})) | (o_t, \mathcal{C}(\phi_t)), a_t) \end{cases}$$

The equivalence holds if and only if:

$$\mathcal{C} \circ \phi = \mathcal{T}$$

(up to a permutation)

Relaxing the equivalent MDP

We proved that given a feature function that satisfies for all trajectory pairs:

$$\inf_{\gamma_1,\gamma_2,\mathcal{T}(\gamma_1)\neq\mathcal{T}(\gamma_2)} |\phi(\gamma_1)-\phi(\gamma_2)| > \sup_{\gamma_1,\gamma_2,\mathcal{T}(\gamma_1)=\mathcal{T}(\gamma_2)} |\phi(\gamma_1)-\phi(\gamma_2)|.$$

Then the K-means classifier is ensured to verify:

$$\mathcal{C}\circ\phi=\mathcal{T}$$

Thus, expanding the NMRDP boils down to approximating such trajectory feature function.

Solving NMRDPS

$\simeq$

Constructing an equivalent MDP

+

Solving the new MDP

Goal?

Learn the trajectory feature function using a semi supervised signal and a contrastive loss.

semi supervised signal?

=

Batches of similar and different trajectories

This can be seen as a relaxation of the domain knowledge requirement in the general case.

Contrastive loss

We can formally satisfy this constraint

$$\inf_{\gamma_1,\gamma_2,\mathcal{T}(\gamma_1)\neq\mathcal{T}(\gamma_2)} |\phi(\gamma_1)-\phi(\gamma_2)| > \sup_{\gamma_1,\gamma_2,\mathcal{T}(\gamma_1)=\mathcal{T}(\gamma_2)} |\phi(\gamma_1)-\phi(\gamma_2)|.$$

Using the contrastive loss

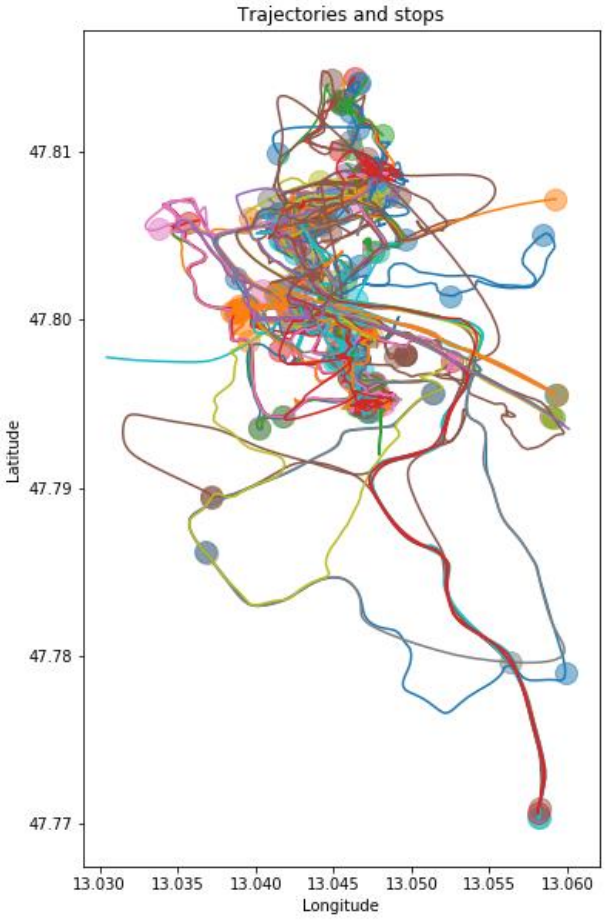
$$\mathcal{L}_{BH}(\theta,\gamma) = \sum_{j=1}^{P}\sum_{i=1}^{K} \text{log1p}\Big(m+ \max_{p\leq K}||\phi(\theta)_i^j-\phi(\theta)_p^j||- \min_{p\leq K,c\leq P,c\neq j}||\phi(\theta)_i^j-\phi(\theta)_p^c||\Big)$$

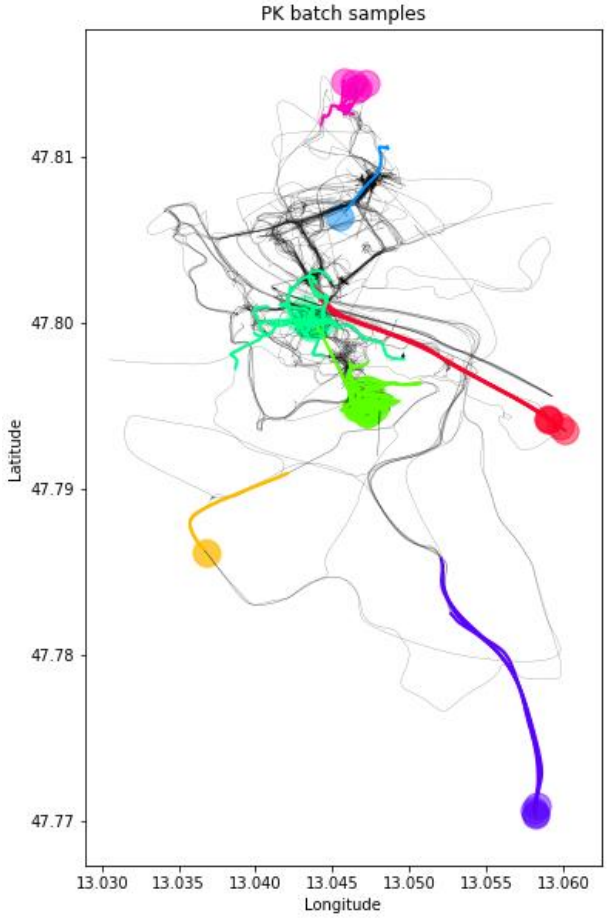
Thus, expanding the NMRDP boils down to minimizing such loss function using trajectory PK batches (K sample from P trajectory class)

3 Experimental results

Does contrastive learning work for trajectories?

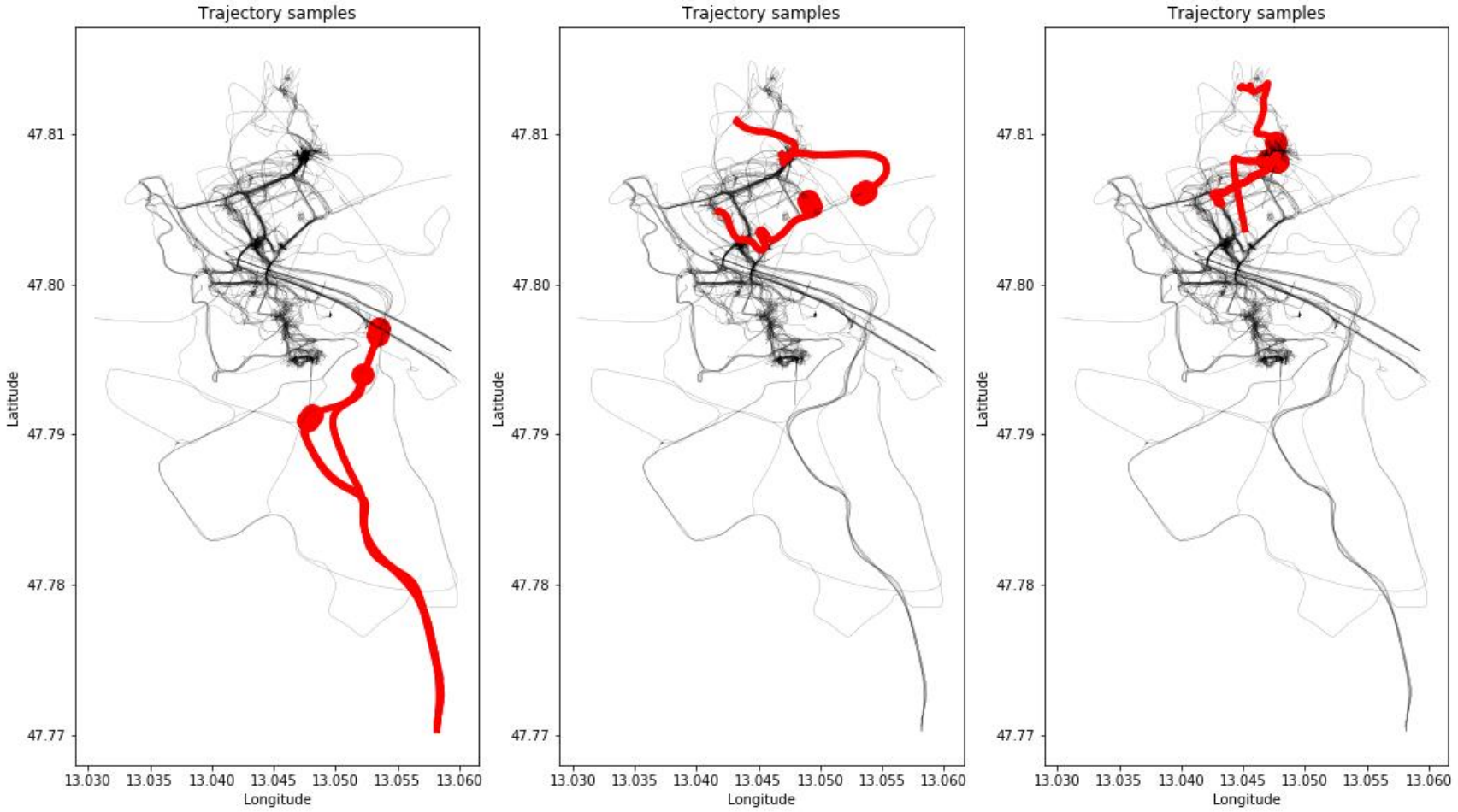
We consider tourist GPS tracks in an open-air museum.
We construct the PK batches using the places where they stopped.





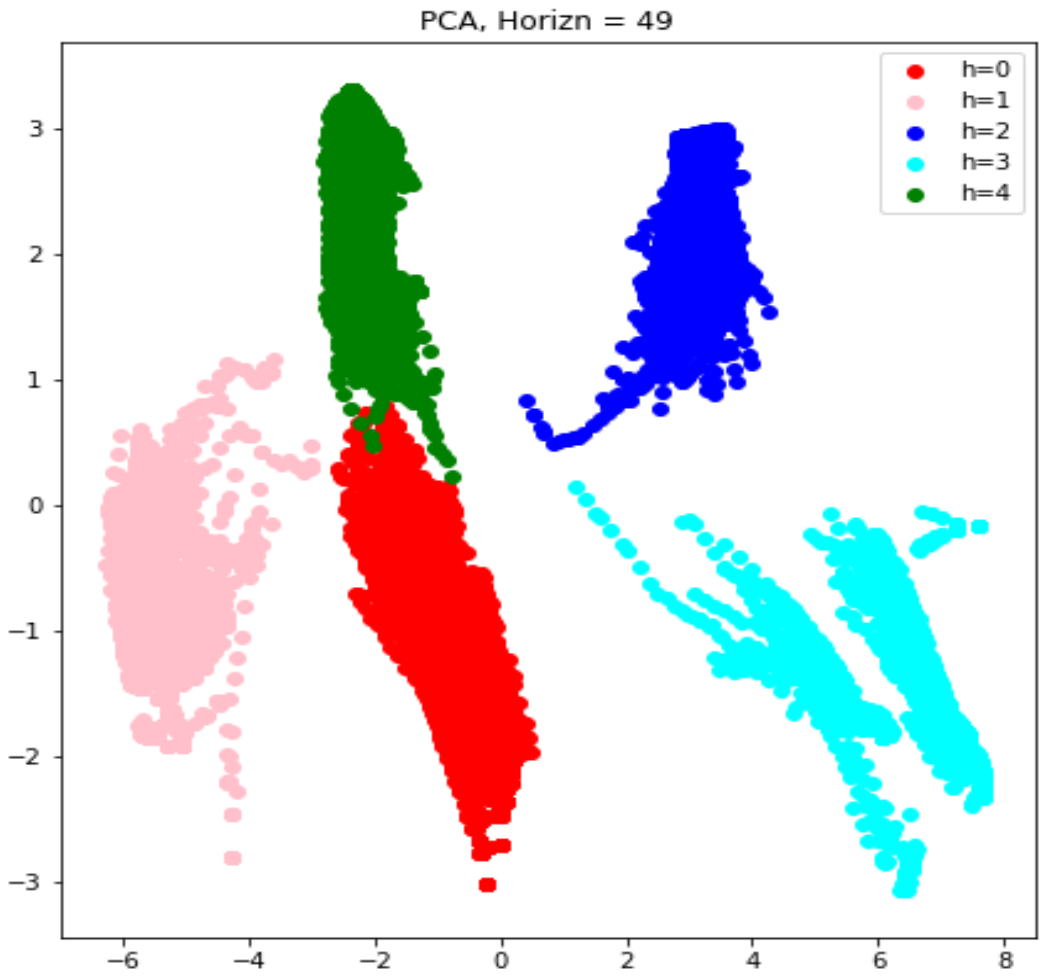
Does contrastive learning work for trajectories?

We sample trajectories with similar representation to check that they are indeed separated



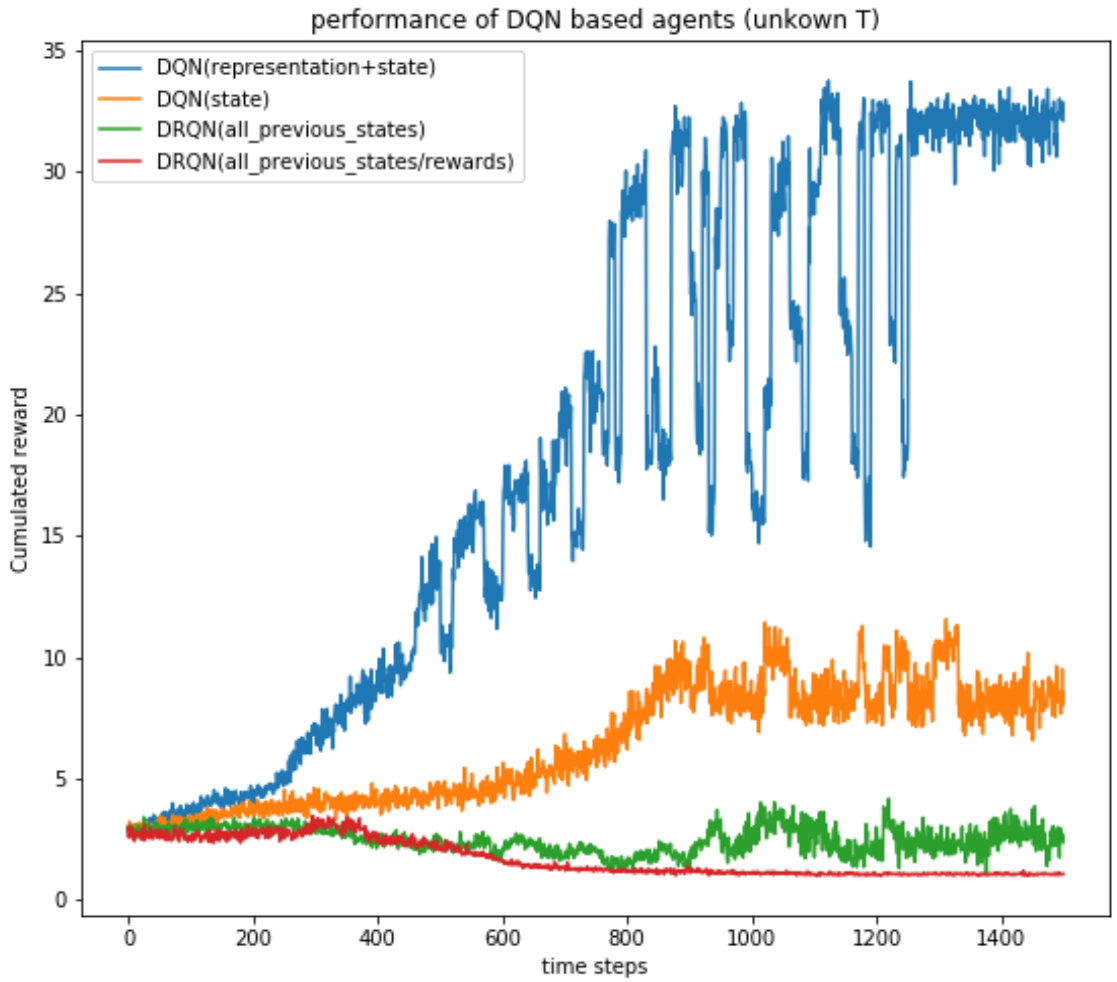
Can contrastive learning expand efficiently NMRDPs?

- We consider an Object-world environment where the task is to collect a succession of objects
- We represent the trajectories' features color-codded according to the associated latent tasks



Is it easy to learn the optimal policy of the NMRDP?

- We consider an Object-world environment where the task is to collect a succession of objects
- We compare DQN performance with or without the latent representation
- We also use as a baseline the DRQN architecture used to learn policies in Attari games



Thanks for watching !