Non-Markovian Policies for Unsupervised Reinforcement Learning in Multiple Environments

Pietro Maldini [*1] **Mirco Mutti** [*1 2] **Riccardo De Santi** [3] **Marcello Restelli** [1]

Abstract

In recent years, the area of Unsupervised Reinforcement Learning (URL) has gained particular relevance as a way to foster generalization of reinforcement learning agents. In this setting, the agent's policy is first pre-trained in an unknown environment via reward-free interactions, often through a pure exploration objective that drives the agent towards a uniform coverage of the state space. It has been shown that this pre-training leads to improved efficiency in downstream supervised tasks later given to the agent to solve. When dealing with the unsupervised pre-training in multiple environments one should also account for potential trade-offs in the exploration performance within the set of environments, which leads to the following question: Can we pre-train a policy that is simultaneously optimal in all the environments? In this work, we address this question by proposing a novel non-Markovian policy architecture to be pre-trained with the common maximum state entropy objective. This architecture showcases significant empirical advantages when compared to state-of-the-art Markovian agents for URL.

1. Introduction

The unsupervised pre-training of learning models on massive unlabelled datasets has lead to remarkable successes in supervised learning tasks, such as language modelling (e.g., Brown et al., 2020; Alayrac et al., 2022) and image generation (Ramesh et al., 2022). With this inspiration, Unsupervised Reinforcement Learning (URL) (Laskin et al., 2021) brings a similar learning paradigm to sequential decision problems. In this setting, a learning agent first interacts with a reward-free environment to pre-train an exploratory behavior. Then, the pre-trained behavior is exploited to address a wide range of downstream supervised tasks, with the goal of improving the adaptation efficiency w.r.t. learning from scratch (i.e., starting with a randomly initialized behavior). While several approaches have been considered for the unsupervised pre-training phase (e.g., Bellemare et al., 2016; Pathak et al., 2017; Eysenbach et al., 2018), the *Maximum State Entropy* (MSE) objective (Hazan et al., 2019; Mutti & Restelli, 2020) has recently emerged as a powerful alternative. The MSE objective prescribes the agent to maximize the entropy of the state visitations induced by its behavior. In this way, whatever task the agent has to face in the downstream supervised phase, the goal state will be reached with a significant probability under the exploratory behavior, which allows the agent to quickly adapt to the optimal task-specific behavior. This recipe allows to achieve compelling results when tackling the URL problem in both continuous control (Mutti et al., 2021) and visual domains (Liu & Abbeel, 2021; Seo et al., 2021; Yarats et al., 2021).

Recent works (Mutti et al., 2022b; Parisi et al., 2021) have considered a further generalization of the URL problem, in which the unsupervised pre-training is performed over a class of environments instead of a single one, and the supervised phase challenges the agent with any task defined on any environment within the class. Notably, the multiple-environments formulation is not a trivial extension of the single-environment URL setting: One can trade-off the pre-training performance over the environments in any way, making the problem inherently multi-objective (Mutti et al., 2022b; Hayes et al., 2022). Optimizing for the average performance over the class may lead to a pre-trained behavior that is bad for some of the environments, so that the agent may struggle with the subsequent supervised tasks in those environments. Conversely, directly accounting for the tail of the pre-training objective, as in (Mutti et al., 2022b), could limit the efficiency of the adaptation w.r.t. the optimal environment-specific behavior. The previous considerations lead to wonder whether a unique behavior that is simultaneously maximizing the pre-training objective in all the environments of the class can be actually learned. While finding such a behavior within the class of stationary Markovian strategies seems far-fetched, the analysis in (Mutti et al., 2022a) suggests that a non-Markovian strategy of this kind

[*]Equal contribution [1]Politecnico di Milano [2]Università di Bologna [3]ETH Zurich. Correspondence to: Mirco Mutti <mirco.mutti@polimi.it>.

First Workshop of Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022, Baltimore, Maryland, USA, PMLR 162, 2022.

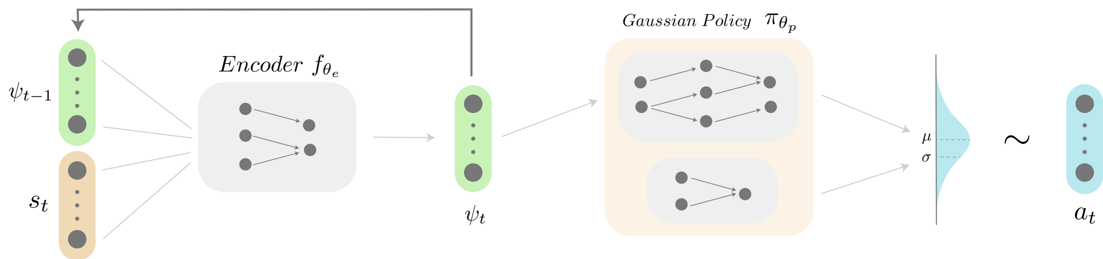


Figure 1. Illustration of the policy architecture learned by MEMENTO. The recursive nature is highlighted with a connection between the output of the history encoder and the input of the history encoder (at the next time step).

exists. In principle, a non-Markovian strategy could identify the environment in a few steps of interaction, then commit to the optimal environment-specific behavior until the end of the episode. Unfortunately, computing the optimal non-Markovian strategy for the MSE objective is known to be impractical in general, as the set of all the histories grows exponentially with the episode horizon (Mutti et al., 2022a).

In this paper, we overcome this computational barrier by expressing a non-Markovian strategy through a convenient recurrent architecture (see Figure 1), which exploits function approximation to recursively compress the history of interactions into a compact representation. Especially, we present a methodology (Section 3), MEmory-based Maximum ENTropy Optimization (MEMENTO), which is adapted from the MEPOL algorithm (Mutti et al., 2021) with the addition of our recurrent architecture, to simultaneously optimize the agent's behavior and the history representation following the same MSE objective over the class of environments. Finally, we provide an illustrative experimental evaluation (Section 4) that shows how the proposed methodology allows to learn a unique non-Markovian strategy that dominates the state-of-the-art Markovian baselines across various classes of illustrative environments.

2. Notation

An episodic *Controlled Markov Process*[1] (CMP) (Puterman, 2014) is defined by a tuple $\mathcal{M} := (\mathcal{S}, \mathcal{A}, P, \mu, T)$, where $\mathcal{S}$ is a continuous state space, $\mathcal{A}$ is a continuous action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$[2] is the transition model, such that $P(s'|a, s)$ denotes the probability of reaching state s' when taking action a in state s, and $\mu \in \Delta(\mathcal{S})$ is the initial state distribution, and T is the horizon of an episode.

In each episode of the CMP, an initial state s_1 is drawn from the initial state distribution μ. The interacting agent observes the state s_1 and picks an action a_1, such that the CMP transitions to the next state s_2 drawn from $P(\cdot|s_1, a_1)$. This interaction process is repeated until the episode ends.

[1] A Markov Decision Process (MDP) without rewards.

[2] The symbol $\Delta(\mathcal{X})$ denotes the simplex over the space $\mathcal{X}$.

A *policy* encodes the behavior of an agent interacting with a CMP. A non-Markovian policy $\pi \in \Pi_{\mathrm{NM}}$ is defined through a function $\pi : \mathcal{H}_T \to \Delta(\mathcal{A})$ that maps any history $h \in \mathcal{H}_T$ to a probability distribution over actions, where $\mathcal{H}_T$ denotes the set of all the histories up to length T. A Markovian policy $\pi \in \Pi_{\mathrm{M}}$ instead maps the last state of the history to a probability over actions $\pi : \mathcal{S} \to \Delta(\mathcal{A})$. Clearly, the set of non-Markovian policies is more general $\Pi_{\mathrm{M}} \subset \Pi_{\mathrm{NM}}$.

Running a policy π over a CMP induces a distribution over the states of the latter (Puterman, 2014), which is called the *marginal state distribution* and it is defined as $d^\pi(s) := \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{P}(s_t = s)$. The MSE objective for unsupervised pre-training asks the agent to find a policy maximizing the entropy of this distribution (Hazan et al., 2019; Mutti et al., 2021), i.e.,[3]

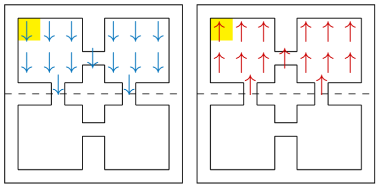
$$\max_{\pi \in \Pi} H(d^\pi) := - \mathop{\mathbb{E}}_{s \sim d^\pi} [\log d^\pi(s)]. \tag{1}$$

3. Algorithm and Architecture

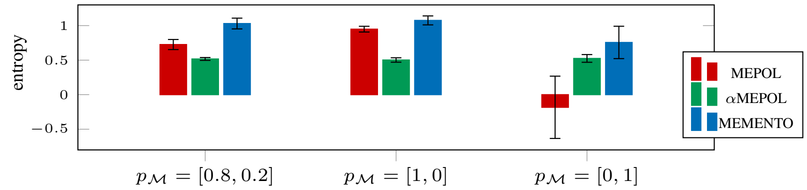
In this work, we aim to learn a non-Markovian policy that allow to address the URL problem over multiple-environments *optimally*, i.e., avoiding dispensable trade-offs between the pre-training performance in the different environments. However, learning such non-Markovian policy exactly is known to be a computationally hard problem (Mutti et al., 2022a), due to the exponential blow-up in the policy representation. Thus, we resort to function approximation in order to learn a compact representation of the desired non-Markovian policy. Especially, we consider a parametric architecture $\pi_\theta, \theta \in \Theta \subseteq \mathbb{R}^q$, which is composed of two modules implemented by separate neural networks: A *recursive history encoder* with parameters θ_e, and a *Gaussian policy* with parameters θ_p, such that $\theta = \theta_e \cup \theta_p$. The complete architecture is depicted in Figure 1.

The recursive history encoder (left-hand side of Figure 1) is used to obtain a compact embedding ψ_t of the current history h_t. Especially, at each time step t, the previous

[3] The set of policies Π denotes either Π_{M} or Π_{NM}.



(a) GridWorld with Slope



(b) Unsupervised pre-training in GridWorld with Slope

Figure 2. Pre-training performance (b) achieved by MEPOL, αMEPOL (α = 0.2), and MEMENTO in the GridWorld with Slope domain, which is depicted in (a). The polices are trained on the class distribution $p_{\mathcal{M}} = [0.8, 0.2]$ and tested on $p_{\mathcal{M}} = [0.8, 0.2]$, $p_{\mathcal{M}} = [1, 0]$, $p_{\mathcal{M}} = [0, 1]$ respectively. For all the experiments, we provide 95% c.i. over 16 runs.

embedding ψ_{t-1} and the current state s_t are encoded into a new history embedding ψ_t according to the following recursive equation

$$\begin{cases} \psi_0 = 0 \\ \psi_t = f_{\theta_e}(s_t, \psi_{t-1}), \end{cases}$$

where f_{θ_e} is a multi-layer perceptron parameterized by θ_e.

The output of the recursive history encoder is then fed into a Gaussian policy (right-hand side of Figure 1) that defines the action selection strategy. Especially, the Gaussian policy is a function of the current history embedding ψ_t that returns the parameters, i.e., the mean μ and the standard deviation σ, of a Gaussian distribution from which the action a_t is sampled. The Gaussian policy is implemented through a two-headed multi-layer perceptron parameterized by θ_p. The mean μ is given by a non-linear transformation of the history embedding ψ_t, while the standard deviation σ is independent from the input ψ_t of the network.

Having introduced the architecture at the base of our methodology, we should now discuss how it can be trained. Similarly as in (Mutti et al., 2022b), we consider the MSE objective for the URL problem in multiple environments. In particular, we propose an algorithm, which we call *MEmory-based Maximum ENTropy Optimization* (MEMENTO), to efficiently optimize the MSE objective from interactions. Taking inspiration from MEPOL (Mutti et al., 2021), MEMENTO combines non-parametric state entropy estimation and policy optimization into a flexible model-free procedure. First, a batch of trajectories is drawn from the class of environments with the current policy. Then, an approximation of the state entropy induced by the current policy is computed with a k-nearest neighbor (k-NN) estimator of the entropy (Singh et al., 2003). Finally, the gradient of the estimated entropy is backpropagated through the policy architecture. Notably, the gradient is not just propagated to the Gaussian policy parameters θ_p, so that to adapt the action selection strategy in the direction of increasing entropy. It is also propagated to the recursive history encoder parameters θ_e, which fosters embeddings of the history that better serve the entropy optimization. Having briefly described the proposed methodology, we provide an empirical evaluation of its merits in the next section. A detailed pseudo-code of the MEMENTO algorithm can be found in Appendix A.

4. Experiments

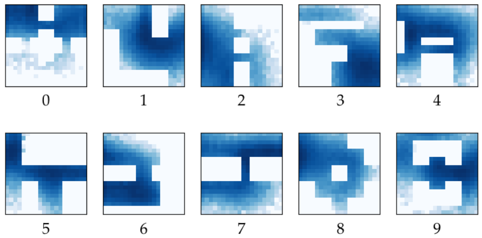
In this section, we provide an empirical validation of the proposed methodology and the superiority of non-Markovian policies for the URL problem in multiple-environments. To this end, we compare the pre-training performance achieved by a non-Markovian policy trained with MEMENTO against Markovian policies trained with MEPOL (Mutti et al., 2021) and αMEPOL (Mutti et al., 2022b). The former is an algorithm designed for optimizing the MSE objective with Markovian policies that is agnostic to the multiple-environments setting, i.e., it maximizes the average entropy across the class. The latter is a retooled version of MEPOL that looks for a conservative solution to the multiple-environments setting, by maximizing the entropy within a pre-specified percentile (controlled with the parameter α) of its distribution, rather than the plain average. Especially, we are able to show that

4.1) The non-Markovian policies Pareto dominates their Markovian counterparts over a class of two illustrative continuous gridworld domains;
4.2) Similar considerations extends to larger classes of environments, such as a set of ten continuous gridworlds;
4.3) A non-Markovian policy can efficiently deal with the *identification-exploitation dilemma*, i.e., deciding when to gather information about the environment, and when to exploit this information with the bets environment-specific strategy.

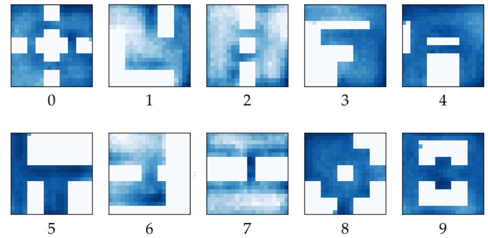
Additional details about the experimental parameters can be found in Appendix B.

4.1. The Pareto Dominance of Non-Markovian Policies

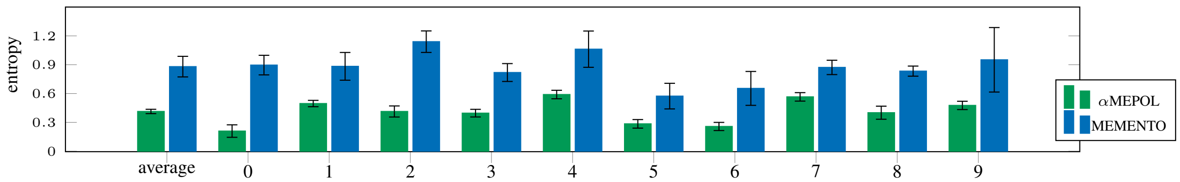
In this section, we show that a non-Markovian policy learned with MEMENTO Pareto dominates their Markovian coun-



(a) Heatmap of the state visitations with αMEPOL



(b) Heatmap of the state visitations with MEMENTO



(c) Unsupervised pre-training in MultiGrid

Figure 3. Heatmap of the state visitations induced by a policy trained with αMEPOL (a) and MEMENTO (b) in all the configurations of the MultiGrid domain. Pre-training performance (c) achieved by αMEPOL ($\alpha = 0.1$) and MEMENTO in the MultiGrid domain. Each environment of the class is sampled with probability 0.1. In all the experiments, we provide 95% c.i. over 16 runs.

terparts, i.e., it achieves a superior pre-training performance in all the environments of the class simultaneously.

To this end, we consider a class $\mathcal{M}$ of two continuous gridworld domains that was presented in (Mutti et al., 2022b) under the name of *GridWorld with Slope* (see Figure 2a). The two gridworlds have a similar configuration, as they are both composed of four rooms connected by narrow hallways. The state of the environment is represented by the 2D spatial coordinates of the agent, while a 2D action defines the motion of the agent along the coordinate directions. The main difference between the two environments lies in the transition dynamics. As shown in Figure 2a, the two configurations have specific slopes (represented by colored arrows) that oppose (or favor) the agent's motion in the upper rooms. Considering the yellow area as the initial position of the agent, the gridworld on the left-hand side is easier to explore, as the slope helps the agent reaching the lower rooms, w.r.t. the one on the right-hand side, which has an unfavorable slope. Moreover, the former is sampled with probability 0.8, whereas the latter with probability 0.2.

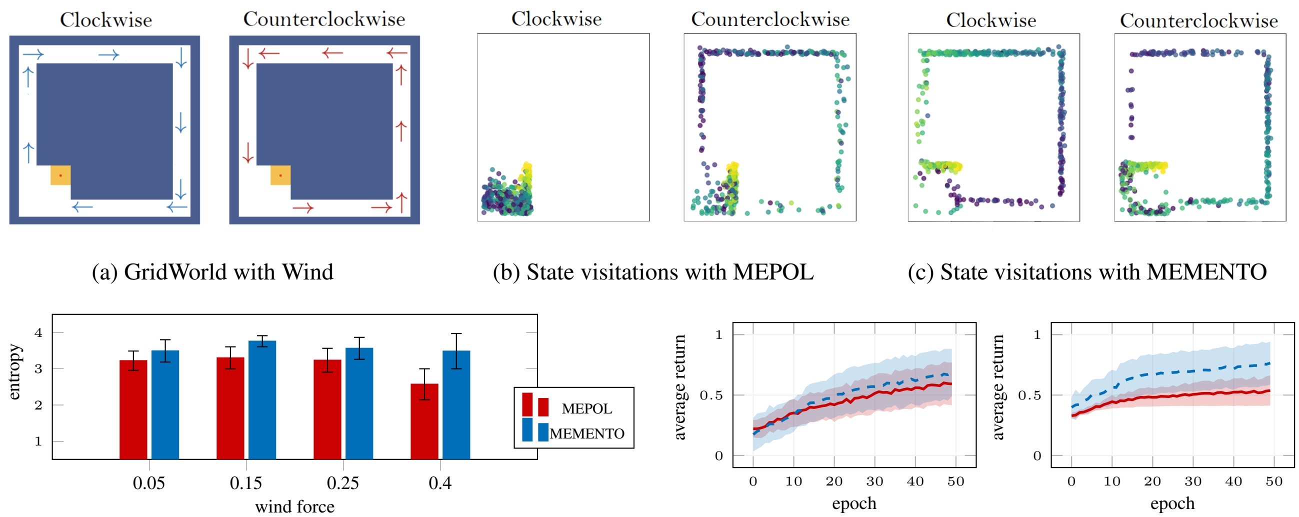
In the described setting, we compare the unsupervised pre-training performance achieved by MEPOL, αMEPOL ($\alpha = 0.2$), and MEMENTO after 500 training epochs. Unsurprisingly, MEPOL achieves a good average performance over the class of environments (Figure 2b left), but it struggles in the most unfavorable configuration (Figure 2b right). Instead, αMEPOL is competitive in the most unfavorable configuration at the expenses of the performance in the easier configuration (Figure 2b center). The non-Markovian policy learned by MEMENTO outperforms the baselines in both the average over the class, the easier configuration, and the unfavorable configuration. Thus, we can say that the non-Markovian policy Pareto dominates the Markovian alternatives in this setting.

4.2. Scaling to Larger Classes of Environments

The careful reader could argue that a class of just two environments does not really challenge the proposed methodology, as the non-Markovian policy has an easy time discriminating between the two configurations. In this section, we instead show that similar results transfer to bigger classes of environments. Especially, we consider the *MultiGrid* domain proposed in (Mutti et al., 2022b), which includes ten different configurations of a 2D continuous gridworld. A crude visualization of these configurations can be inferred by the heatmaps in Figures 3a, 3b.

In this setting, we compare the unsupervised pre-training performance achieved by αMEPOL ($\alpha = 0.1$) and MEMENTO after 500 training epochs.[4] Similarly as in the GridWorld with Slope domain, MEMENTO is able to achieve a good pre-training performance across the boards, and it outmatches αMEPOL in all the environments simultaneously (see Figure 3c). The superior performance is visually reflected by the heatmaps of the state visitations reported in Figures 3a, 3b. Especially, in the configurations $0, 2, 9$ the policy trained with MEMENTO is able to cover regions that are barely visited by the policy trained with αMEPOL.

[4]We avoid comparing with MEPOL in this experiment, as it is dominated by αMEPOL in this setting (Mutti et al., 2022b).



(a) GridWorld with Wind (b) State visitations with MEPOL (c) State visitations with MEMENTO

(d) Unsupervised pre-training with different wind forces (e) Fine-tuning in Clockwise (left) and Counterclockwise (right)

Figure 4. Experiments in the GridWorld with Wind domain, which is depicted in (a). Point-map of the state visitations induced by a policy trained with MEPOL (b) and MEMENTO (c) with wind force 0.4. Pre-training performance (d) achieved by MEPOL and MEMENTO for different configurations of the GridWorld with Wind, in which the wind force varies (95% c.i. over 16 runs). Supervised fine-tuning performance achieved by TRPO (Schulman et al., 2015) with MEPOL and MEMENTO initializations (95% c.i. over 5 goal locations).

4.3. The Identification-Exploitation Dilemma

In the previous sections, we considered settings in which the different environments within a class can be discriminated in a few steps, either by experiencing a particular slope, or by the walls' configuration. In other domains, identifying the environment in order to exploit the best environment-specific strategy could come at a significant cost in terms of the pre-training performance. This poses a problem that we call the *identification-exploitation dilemma*, i.e., understanding when it is convenient to seek for the environment identification rather than directly optimizing the pre-training objective under uncertainty. A Markovian policy can be essentially placed at one extreme of the spectrum, as it always exploits the interactions to increase the entropy without knowledge about the specific environment. Instead, it is interesting to see how a non-Markovian policy can cope with the uncertainty about the environment, and alternatively decide to exploit or to gather additional information in relation to the expected identification cost.

To study the identification-exploitation dilemma, we consider a specific domain composed of two 2D continuous gridworlds, which we call the GridWorld with Wind. As one can see from Figure 4a, the two gridworlds consist of a wide room at the bottom left, and a narrow hallway that connect the two entrances of the room. In the two configurations, which are sampled with equal probability, a wind flows in opposite directions, such that it is easier to explore the one on the left-hand side (Clockwise) going in one direction, and the other (Counterclockwise) going in the other direction. The higher is the force of the wind, the higher is the value of identifying the wind direction before selecting which direction to take. In Figure 4d, we show that a non-Markovian policy trained with MEMENTO is able to handle the identification-exploitation trade-off, whereas the pure exploitation of MEPOL fails with an high wind. As it is evident from the point-maps in Figure 4b, MEPOL commits to a direction irrespective of the actual environment, which lead to a poor performance in the Clockwise configuration. Instead, MEMENTO first identifies the environment and then proceeds in the optimal direction (Figure 4c). Finally, we show that this superior pre-training performance achieved by MEMENTO benefits the subsequent fine-tuning to supervised tasks defined in the same class of environments. Especially, we show that the TRPO (Schulman et al., 2015) algorithm equipped with the MEMENTO initialization outperforms TRPO with the MEPOL initialization (Figure 4e).

5. Conclusion

In this work, we proposed a computationally efficient methodology to pre-train a non-Markovian policy across multiple environments. We showed that this policy outperforms state-of-the-art Markovian agents in the URL setting.

Acknowledgements

We would like to thank Hao Liu and Ivan Ovinnikov for providing useful feedback in an early stage of this work.

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2018.

Hayes, C. F., Rădulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., Verstraeten, T., Zintgraf, L. M., Dazeley, R., Heintz, F., et al. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):1–59, 2022.

Hazan, E., Kakade, S., Singh, K., and Van Soest, A. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pp. 2681–2691. PMLR, 2019.

Laskin, M., Yarats, D., Liu, H., Lee, K., Zhan, A., Lu, K., Cang, C., Pinto, L., and Abbeel, P. Urlb: Unsupervised reinforcement learning benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Liu, H. and Abbeel, P. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34, 2021.

Mutti, M. and Restelli, M. An intrinsically-motivated approach for learning highly exploring and fast mixing policies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Mutti, M., Pratissoli, L., and Restelli, M. Task-agnostic exploration via policy gradient of a non-parametric state entropy estimate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Mutti, M., De Santi, R., and Restelli, M. The importance of non-markovianity in maximum state entropy exploration. *arXiv preprint arXiv:2202.03060*, 2022a.

Mutti, M., Mancassola, M., and Restelli, M. Unsupervised reinforcement learning in multiple environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022b.

Parisi, S., Dean, V., Pathak, D., and Gupta, A. Interesting object, curious agent: Learning task-agnostic exploration. *Advances in Neural Information Processing Systems*, 34, 2021.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

Seo, Y., Chen, L., Shin, J., Lee, H., Abbeel, P., and Lee, K. State entropy maximization with random encoders for efficient exploration. In *International Conference on Machine Learning*, pp. 9443–9454. PMLR, 2021.

Singh, H., Misra, N., Hnizdo, V., Fedorowicz, A., and Demchuk, E. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23 (3-4):301–321, 2003.

Yarats, D., Fergus, R., Lazaric, A., and Pinto, L. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pp. 11920–11931. PMLR, 2021.

A. Algorithm

In this section, we present the pseudo-code of MEMENTO (Algorithm 1). The learning process starts from an initial policy with parameters θ. At each epoch, the algorithm collects N batches of B trajectories of length T. Hence, the samples collected are used to perform off-policy updates of the policy parameters. First, an estimate of the KL divergence between the distributions induced by the sampling policy and the updated policy is computed. Then, the algorithm perform an off-policy update as long as the policy remains within a trust region (Schulman et al., 2015) around the sampling policy (or until a fixed number of off-policy iterations is reached). At convergence, MEMENTO outputs both an history encoder and a diagonal Gaussian policy that together represents the learned non-Markovian behaviour. In the following pseudo-code Σ_{epoch} stands for the number of off-policy iterations performed in each epoch, while Σ_{max} stands for the maximum number of off-policy iterations that can be performed in an epoch.

Algorithm 1 MEMENTO

Input: Horizon T, number of batches N, batch-size B, trust-region threshold δ, learning rate α, nearest neighbors k, max off-policy iterations Σ_{max}
initialize θ
for $epoch = 0, 1, 2, ...$ until convergence **do**
  draw N batches of B trajectories of length T
  $\theta^{'} \leftarrow \theta, \Sigma_{epoch} \leftarrow 0$
  while $D_{KL}(\hat{d}_T(\theta)||\hat{d}_T(\theta^{'})) \leq \delta$ & $\Sigma_{epoch} \leq \Sigma_{max}$ **do**
    $\theta^{'} = \theta^{'} + \alpha \nabla_{\theta^{'}} \hat{H}_k(\hat{d}_T(\theta^{'})|\hat{d}_T(\theta))$
    $\Sigma_{epoch} \leftarrow \Sigma_{epoch} + 1$
  end while
  $\theta \leftarrow \theta^{'}$
end for
Output: Recursive history encoder f_{θ_e}, Gaussian policy π_{θ_p}

B. Experimental details

Here we specify the experimental details of the methods run within the experimental section, namely MEMENTO, MEPOL, and αMEPOL.

B.1. MEMENTO

Parameters	GridWorld with Slope	GridWorld with Wind	MultiGrid
Number of epochs	500	500	500
Samples collected for each	200	100	200
Horizon(T)	400	400	400
Batch Size	5	1	5
KL threshold (δ)	15	0.1	15
Learning rate	10^{-4}	$4 \cdot 10^{-5}$	10^{-5}
Max off policy iterations	30	30	30
Number of neighbors (k)	30	30	30
recursive encoder layer sizes	(300,300)	(300,300)	(300,300)
history vector length	100	100	100
Policy hidden layers sizes	(300,300)	(300,300)	(300,300)
History encoder output act. function	ReLU	ReLU	ReLU
Policy and History encoder act. function	ReLU	ReLU	ReLU
Number of seeds	8	8	4

Table 1. Hyperparameters of MEMENTO.

B.2. MEPOL

Parameters	GridWorld with Slope	GridWorld with Wind
Number of epochs	500	500
Samples collected for each	200	100
Horizon(T)	400	400
Batch Size	5	1
KL threshold (δ)	15	0.1
Learning rate	10^{-5}	$4 \cdot 10^{-5}$
Max off policy iterations	30	30
Number of neighbors (k)	30	30
Policy hidden layers sizes	(300,300)	(300,300)
Policy hidden layers act. function	ReLU	ReLU
Number of seeds	8	8

Table 2. Hyperparameters of MEPOL.

B.3. αMEPOL

Parameters	GridWorld with Slope	MultiGrid
Number of epochs	500	500
Samples collected for each	200	200
Horizon(T)	400	400
Batch Size	5	5
KL threshold (δ)	15	15
Learning rate	10^{-5}	10^{-5}
Max off policy iterations	30	30
Number of neighbors (k)	30	30
α	0.35	0.1
Policy hidden layers sizes	(300,300)	(300,300)
Policy hidden layers act. function	ReLU	ReLU
Number of seeds	8	4

Table 3. Hyperparameters of αMEPOL.