



POLITECNICO
MILANO 1863

**SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE**

EXECUTIVE SUMMARY OF THE THESIS

Handling Partial Observability in Non-Stationary Reinforcement Learning via Hidden Markov Models

LAUREA MAGISTRALE IN COMPUTER SCIENCE & ENGINEERING - INGEGNERIA INFORMATICA

Author: CATERINA MOTTI

Advisor: PROF. MARCELLO RESTELLI

Co-advisors: GUSMEROLI ENEA, LAVELLI ALESSANDRO, TENEDINI DAVIDE

Academic year: 2025-2026

1. Introduction

The last decade has witnessed a surge in Reinforcement Learning (RL) research due to its remarkable success in solving complex decision-making problems, from superhuman performance in gaming [4] to sophisticated robotic manipulation [1], driving an unprecedented growth in both academic and industrial interest. Despite this progress, the vast majority of RL research has concentrated on a strong, yet often unrealistic, assumption: the stationarity of the environment. While this simplification has enabled the development of stable, high-performance algorithms within controlled simulations, it remains unrealistic for deployment in real-world scenarios. In fields ranging from finance to robotics to healthcare, environments are inherently dynamic, with underlying processes often shifting due to unobservable latent variables. This creates a state of partial observability, where identical actions taken in identical states may lead to different outcomes. Under these non-stationary conditions, standard RL algorithms suffer from perceptual aliasing, and their learned policies systematically fail. Consequently, bridging the gap between simulated environments and the complexity of real-world applications has emerged as a cen-

tral challenge for the next generation of RL research [2].

1.1. Contributions

To address this gap, our work focuses on environments where non-stationarity arises from latent, unobservable processes. We propose a **decoupled framework** that separates latent estimation of these hidden dynamics from the policy optimisation process. By training a latent estimator offline on historical trajectories, we can augment the environment's observation space online with a belief over the latent modes, effectively reducing the non-stationary Partially Observable Markov Decision Process (POMDP) to a stationary belief Markov Decision Process (MDP). This modularity allows the framework to function independently of the control objective, enabling the use of standard RL agents without requiring structural modifications.

Our approach relies on an **extension of the classical Baum-Welch algorithm**, which we have adapted to handle **input-conditioned emission** for both discrete and continuous domains. Our empirical evaluations, conducted on non-stationary variants of the Cliff Walking task under both deterministic and stochastic latent dy-

namics, demonstrate that this latent-augmented approach **successfully mitigates the catastrophic effects of non-stationarity**. Ultimately, this work enables RL agents to achieve high-performing policies in environments where standard baselines systematically fail.

2. Preliminaries

To formalise the non-stationary problem and establish the mathematical foundation for our proposed framework, we rely on the core concepts of sequential decision-making and probabilistic latent inference.

2.1. Markov Decision Processes

Sequential decision-making problems in RL are traditionally modelled as MDPs. An MDP is a stochastic process defined as a tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma, \mu \rangle$. Starting from a state $s_0 \in \mathcal{S}$ sampled from the initial distribution μ , the agent chooses an action $a \in \mathcal{A}$ according to a decision-making strategy. The environment responds by emitting an immediate reward $r \sim R(s, a)$ and transitioning to the successor state $s' \sim P(\cdot | s, a)$. This framework relies on the fundamental Markov assumption, which states that the current observable state contains all the information necessary to predict future states, or more informally, that the future is independent of the past given the present. Under this assumption, the environment is inherently stationary, allowing the agent to learn an optimal, memoryless policy that maps each state to a specific action, i.e., $\pi(a | s)$. The agent’s goal is to learn a policy that maximises the expected discounted cumulative reward, with the parameter $\gamma \in [0, 1]$ dictating the present value of future observations.

2.2. Partially Observable Markov Decision Processes

When an environment is non-stationary due to unobservable latent factors, the Markov assumption breaks down. This problem can be formulated as a POMDP, in which the true latent state of the environment is hidden, and the agent only receives a partial observation. This limited perspective introduces perceptual aliasing, as identical states can lead to completely different outcomes depending on the hidden mode. Consequently, standard RL algorithms systematically fail, as they attempt to map actions to

physical observations that are no longer uniquely informative.

2.3. Hidden Markov Models

To overcome perceptual aliasing and restore the Markov property, the agent cannot rely solely on its observations of the environment. Instead, it must maintain a belief state, a probability distribution representing its confidence over the possible hidden latent modes. We model these dynamics using an Hidden Markov Model (HMM) [3] defined by a tuple $\langle A, B, b_1 \rangle$, where A is the transition model of the temporal evolution of latent states, B is the emission model defining how these latent states manifest as observations o , and b_1 is the initial state distribution.

Given a sequence of observations $O = (o_1, \dots, o_T)$ of length T , these parameters can be learned by the Baum-Welch algorithm, an Expectation-Maximisation procedure that alternates between two steps. In the E-step, we compute the forward variable $\alpha_t(z)$, representing the probability of observing the partial sequence up to time t and being in state z , and the backward variable $\beta_t(z)$, representing the probability of the future observations given state z at time t :

$$\begin{aligned} \alpha_1(z) &= b_1(z) \cdot B(o_1 | z), \\ \alpha_t(z) &= B(o_t | z) \sum_{z' \in \mathcal{Z}} \alpha_{t-1}(z') A(z | z'), \quad (1) \\ &\text{for } t > 1; \end{aligned}$$

$$\begin{aligned} \beta_T(z) &= 1, \\ \beta_t(z) &= \sum_{z' \in \mathcal{Z}} A(z' | z) B(o_{t+1} | z') \beta_{t+1}(z'), \quad (2) \\ &\text{for } t < T. \end{aligned}$$

Using these, we estimate the marginal probability $\gamma_t(z)$ of being in state z at time t , and the joint probability $\xi_t(z, z')$ of transitioning from z to z' :

$$\gamma_t(z) = \frac{\alpha_t(z) \beta_t(z)}{\sum_k \alpha_t(k) \beta_t(k)}, \quad (3)$$

$$\xi_t(z, z') = \frac{\alpha_t(z) A(z' | z) B(o_{t+1} | z') \beta_{t+1}(z')}{D_t}, \quad (4)$$

where $D_t = \sum_k \sum_w \alpha_t(k) A(w | k) B(o_{t+1} | w) \beta_{t+1}(w)$ is the normalization factor. In the M-step, these expected values are used to update the transition and emission models, maximising the likelihood of the observed sequence:

$$\hat{a}_{zz'} = \frac{\sum_{t=1}^{T-1} \xi_t(z, z')}{\sum_{t=1}^{T-1} \gamma_t(z)}, \quad (5)$$

$$\hat{b}_{zo} = \frac{\sum_{t=1, o_t=o}^T \gamma_t(z)}{\sum_{t=1}^T \gamma_t(z)}.$$

This two-step procedure is repeated iteratively. A fundamental property of the Baum-Welch algorithm is its convergence guarantee: at each iteration, the updated parameters are mathematically proven to strictly increase, or at least maintain, the log-likelihood of the observed sequence. The process terminates when the change in log-likelihood falls below a predefined threshold. However, while convergence is guaranteed, the algorithm is only proven to reach a local maximum, making the quality of the final model highly dependent on the initial parameter estimates.

3. Problem Formulation

To formally model this problem, we frame the environment as an Hidden Mode Markov Decision Process (HM-MDP) defined by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{Z}, A, P, R, \gamma \rangle$. In this setting, the state and action spaces are fully observable, but the transition dynamics P are governed by a set of unobservable latent modes $\mathcal{Z}$. These hidden modes evolve according to a first-order Markov chain, determined by the transition model A , independently of the agent’s actions (passive non-stationarity). Moreover, we assume that the reward function is mode-independent. Since the environment is once again partially observable, the agent suffers from perceptual aliasing.

To allow the agent to discriminate between different latent modes, we augment its observation space with a sufficient statistic that summarises the historical trajectory, thus restoring the Markov property: the belief. To compute this belief, we leverage an extended version of the Baum-Welch algorithm to account for conditioning the emission model on the state-action pair. Specifically, the emission distribution, that is, the probability of transitioning to a successor state, is conditioned on the current state, action, and latent mode. This design choice resolves the ambiguity introduced by mode-dependent dynamics, providing a highly informative, discriminative signal for latent inference.

By applying Bayesian filtering over this HMM, we recursively estimate the posterior belief over the latent modes at each timestep. We use a compact representation of this inferred belief by sampling a latent mode $\hat{z}_t$ to augment the agent’s

observation.

4. Methodology

To implement this state augmentation, we propose a framework that decouples latent context tracking from policy optimisation. This separation allows the complexity of the non-stationary environment to be handled independently from the agent’s decision-making process. The framework operates in two distinct phases: offline latent estimator model learning and online context-conditioned control.

4.1. Offline Latent Model Learning

In this first phase, the parameters of the HMM are learned offline from a static dataset of historical trajectories, collected with a random policy. To ensure broad applicability, we developed two variants of the input-conditioned Baum-Welch algorithm. For discrete environments, we derived an exact tabular variant where the emission model is stored and updated as a multidimensional tensor. For continuous state spaces, where tabular representations are intractable, we derived a neural variant, in which the emission model is approximated by a deep neural network, and the M-step optimisation is performed via gradient ascent using a weighted likelihood on stochastically sampled latent assignments.

4.1.1 Tabular Variant for Discrete Domains

For discrete environments, we store the emission model as a tensor $\mathbf{B}$ of dimensions $|\mathcal{S}| \times |\mathcal{A}| \times |\mathcal{Z}| \times |\mathcal{S}|$.

In the E-step, the forward and the backward variables are computed following Equations 1 and 2, by substituting the standard emission $B(o | z)$ with our state-action-conditioned term $B(s_{t+1} | s, a, z)$. Using these, $\gamma_t(z)$ is computed as in Equation 3, while for $\xi_t(z, z')$ the same substitution for the emission model is performed in Equation 4.

In the M-step, the update of $\mathbf{A}$ remains the same as in Equation 5, while the update of $\mathbf{B}$ proceeds as follows:

$$\hat{b}_{s,a,z,s'} = \frac{\sum_{t:(s_t, a_t, s_{t+1})=(s, a, s')} \gamma_t(z)}{\sum_{t:(s_t, a_t)=(s, a)} \gamma_t(z)}.$$

4.1.2 Neural Variant for Continuous Domains

For continuous environments, we approximate the emission model using a deep neural network $B_\theta(s_{t+1} | s_t, a_t, z)$, which takes as input the state-action pair and outputs the parameters (means μ_θ and standard deviation σ_θ) of a Gaussian distribution over the next state for each latent mode $z \in \mathcal{Z}$. Assuming independence across the d observation dimensions, the emission probability is factorised as:

$$B_\theta(s_{t+1} | s_t, a_t, z) = \prod_{i=1}^d \mathcal{N}(s_{t+1}^{(i)}; \mu_\theta^{(i)}, \sigma_\theta^{(i)}),$$

where $\mu_\theta^{(i)}$ and $\sigma_\theta^{(i)}$ are functions of the state-action pair (s_t, a_t) and the latent mode z .

In the E-step, the forward and backward variables are computed exactly as in the tabular case, substituting the tensor entries with the probability density values evaluated by the network B_θ . This yields the posterior marginals $\gamma_t(z)$ and $\xi_t(z, z')$.

In the M-step, instead of exact frequency counting, the emission parameters θ are updated via gradient ascent. To reduce the computational and memory burden of evaluating the network and accumulating gradients for all possible latent modes at every timestep, a latent mode $\hat{z}_t$ is stochastically sampled from the current posterior $\gamma_t(z)$. The network is then updated to maximise the log-likelihood of the observed transitions:

$$\mathcal{L}_{\text{emit}}(\theta) = \frac{1}{T} \sum_{t=1}^{T-1} \log B_\theta(s_{t+1} | s_t, a_t, \hat{z}_t).$$

4.2. Online Mode-Conditioned Policy Optimisation

In the second phase, the pre-trained latent estimator is embedded into the environment as a real-time tracking wrapper. As the agent interacts with the environment, each observed transition (s_t, a_t, s_{t+1}) is fed into the latent estimator. The estimator performs a Bayesian filtering procedure consisting of two steps: a correction step, which refines the current belief using the observed input-conditioned emission, followed by a prediction step, which propagates the belief through the latent transition matrix to forecast the next hidden mode.

Crucially, passing the full continuous belief distribution to the RL agent would introduce severe exploration challenges due to the curse of dimensionality. To maintain computational efficiency, we extract a compact, scalar forecast $\hat{z}_t$ (the expected latent mode) to construct the augmented state representation $x_t = (s_t, \hat{z}_t)$. This transformation effectively reduces the underlying non-stationary POMDP to a fully observable MDP, enabling the deployment of standard, off-the-shelf RL algorithms without any modifications.

5. Experimental Results

To validate the decoupled framework, we developed non-stationary variants of the *Cliff Walking* environment, adapting the task for both discrete and continuous control. The setting consists of a grid where the agent must navigate to reach a goal while avoiding the cliff. The agent receives a reward of -1 for each step, -100 if it falls into the cliff (resulting in a reset to the initial state), and $+10$ upon reaching the goal. To further complicate the task, we design a narrow safe path the agent must traverse to minimise negative reward. For computational simplicity, we limit each episode in our simulations to a finite horizon of 500 timesteps.

5.1. Latent Dynamics

The non-stationarity is introduced via an unobservable wind factor consisting of five distinct modes: the four cardinal directions and a windless state. These modes are arranged in a fixed cyclic sequence, and at each timestep there is a probability p that the environment advances to the next mode in the cycle and a corresponding probability $1 - p$ that the current mode persists. The active wind mode acts as an additive perturbation to the agent's actions: the new position is computed by summing the current position, the chosen action, and the active wind, followed by a clipping operation to enforce the map's boundary constraints.

It is crucial to note that while the environment exhibits only five distinct physical wind modes, the underlying transition model must be structured using eight latent states to preserve the Markovian property of the transition chain.

5.2. Evaluation Metrics

We evaluate the performance of an agent via the mean cumulative reward over episodes. A theoretical upper bound is established by an Oracle baseline that employs the latent estimator mechanism to compute the belief state over the latent modes, but is equipped with the ground truth model parameters. Conversely, a lower bound is established using a standard agent trained on the observable state space without any latent inference mechanism.

To represent the performance at convergence, all reported metrics are computed exclusively over the last 100 episodes. For the latent estimate accuracy, we additionally retain only high-confidence predictions, applying a confidence threshold of 0.85 to the belief state.

5.3. Discrete Environment

To evaluate the framework in a controlled setting, we first analyse the discrete domain under both deterministic ($p = 1$) and stochastic ($p = 0.9$) latent dynamics. As illustrated in Figure 1a and 1b, the standard agent fails to converge to a goal-reaching policy, as without access to the latent mode, it cannot resolve the perceptual aliasing induced by the wind. Conversely, the Oracle converges to an optimal policy, and our proposed Latent Estimator closely bridges the gap, demonstrating that the inferred belief state effectively guides the agent.

Regarding the inference quality, the Latent Estimator achieves an overall accuracy of 83.45% in the deterministic setting, compared to the Oracle’s perfect 100%. In the stochastic setting, the environment’s uncertainty significantly degrades inference, with the Latent Estimator’s accuracy dropping to 46.62% compared to an Oracle of 87.56%. In both cases, the HMM correctly captured the cyclical structure of the latent chain, assigning distinct latent states to physical modes that appear multiple times in the sequence.

Ablation studies reveal that in this discrete formulation, the tabular emission model acts as the primary bottleneck for inference.

5.4. Continuous Environment

Scaling the framework to continuous state and action spaces inherently increases the complexity of the task. As illustrated in Figure 1c and 1d, consistent with the discrete domain, the stan-

dard agent completely collapses, while the Latent Estimator successfully navigates the safe corridor, substantially outperforming the standard baseline. In the stochastic setting, we observe a notably high variance across random seeds for both the Oracle and the Latent Estimator. This indicates that the combination of continuous kinematics and unpredictable wind shifts creates a highly challenging exploration landscape, though the inference mechanism itself remains stable.

A key challenge specific to the continuous domain is the observation aliasing induced by boundary interactions, which account for over 91% of transitions and prevent the HMM from disambiguating the active wind mode. Excluding these aliased transitions, the Latent Estimator achieves an accuracy of 64.26% in the deterministic setting against a perfect Oracle. Moreover, the HMM produces a degenerate state mapping, assigning four out of eight latent states to a single wind mode. Despite this, the Latent Estimator still converges to a near-optimal policy, suggesting that perfect alignment between learned latent states and human-readable environmental variables is not strictly necessary: the belief state provides sufficient context for effective decision-making even when the latent structure is only partially recovered. In the stochastic setting, the Latent estimator achieves 79.21% while the Oracle itself reaches only 78.11%.

6. Conclusions

This work provides two main contributions to the field of RL in non-stationary, partially observable environments. First, we **extended the Baum-Welch algorithm** to handle **input-conditioned emissions** in both discrete and continuous domains. Second, we successfully validated a **decoupled framework** that separates latent state inference from policy optimisation, enabling standard agents to solve tasks where baselines systematically fail. Furthermore, our empirical evaluation revealed a crucial insight: perfect inference of the hidden variables is not strictly necessary. The proposed framework succeeds by encoding environmental uncertainty into a robust belief state that provides sufficient context for effective decision-making.

6.1. Limitations and Future Work

Building upon these findings, several promising directions for future research have been identified:

- **Online refinement of the emission model:** Updating the emission model incrementally during deployment via online Baum-Welch variants to compensate for initial data sparsity.
- **Uncertainty-aware policies:** Handing the full multidimensional belief vector directly to the RL agent, enabling the policy to explicitly act upon inference uncertainty.
- **Handling an unknown number of latent modes:** Relaxing the assumption of a known number of latent modes by deliberately overestimating the number of states. This relies on the HMM to merge redundant modes, provided the risk of overfitting to noise is mitigated.

References

- [1] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.
- [2] Gabriel Dulac-Arnold, Nir Levine, Daniel J. Mankowitz, Jerry Li, Cosmin Paduraru, Sven Goyal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021.
- [3] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

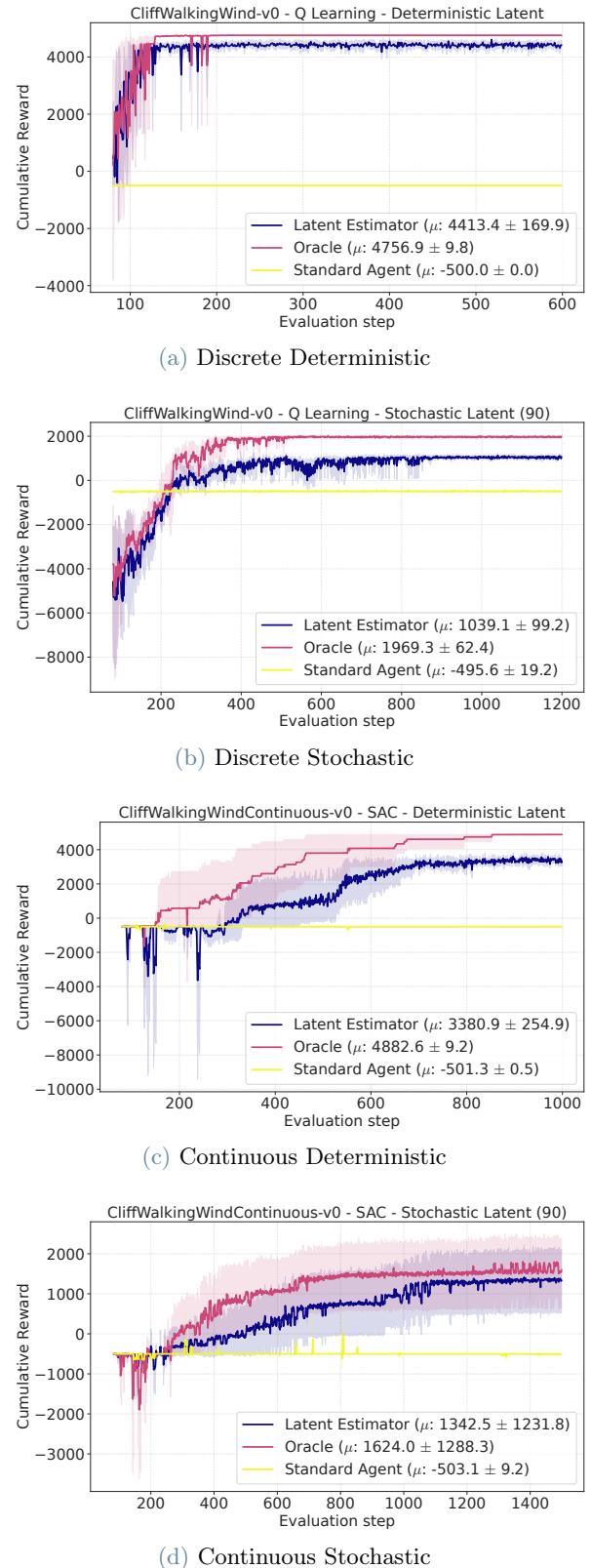


Figure 1: Learning curves across all experimental domains. Solid lines indicate the mean cumulative reward evaluated during independent rollouts, averaged over 5 episodes, and shaded areas represent the 95% confidence intervals across 5 different random seeds.