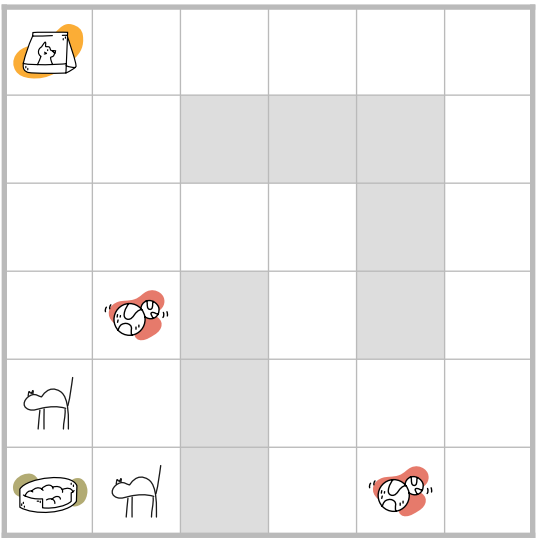
Learning to Remember the Past by Learning to Predict the Future

PhD Defense - April 7th, 2025

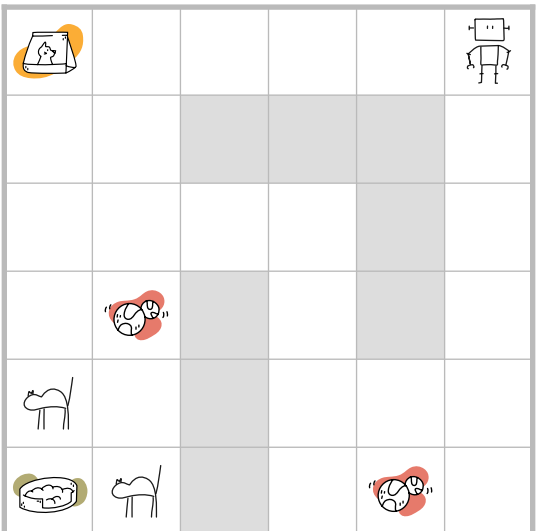
Gaspard Lambrechts

A Matter of Perception

A cat story



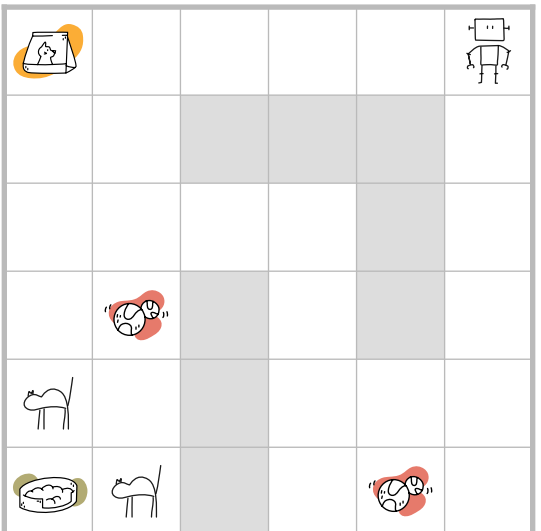
A cat story



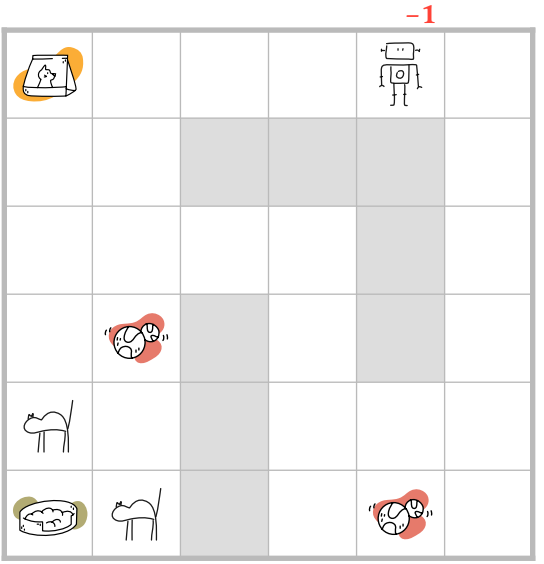
A cat story

What should the robot do?

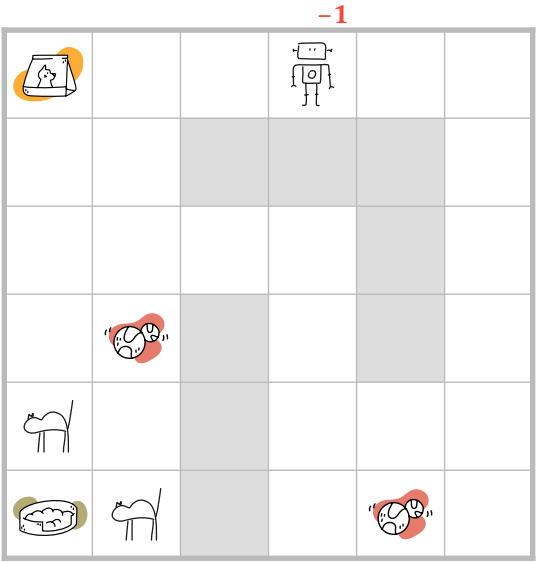
A cat story



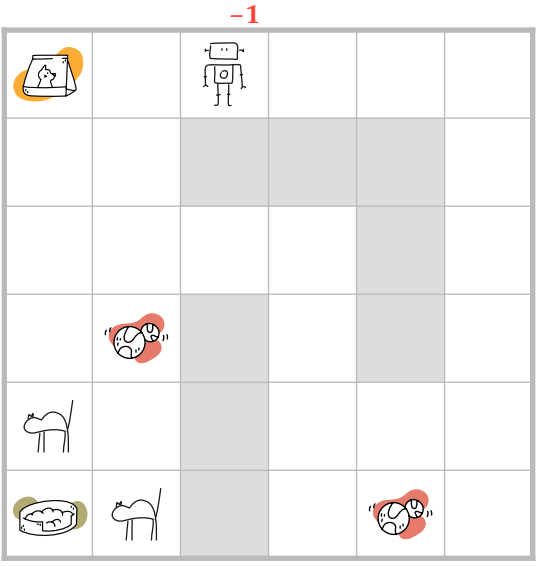
A cat story



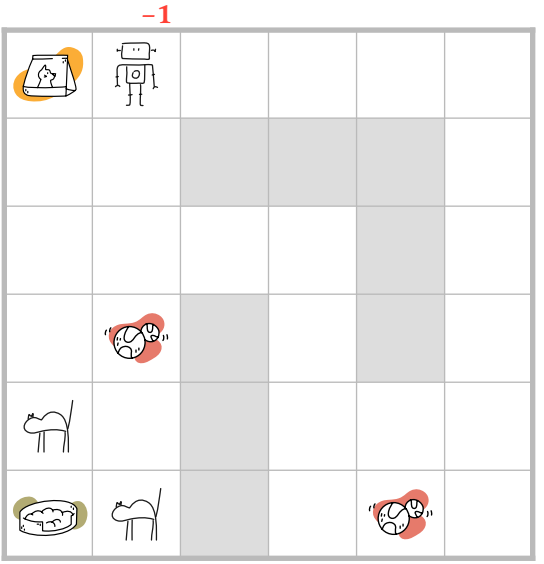
A cat story



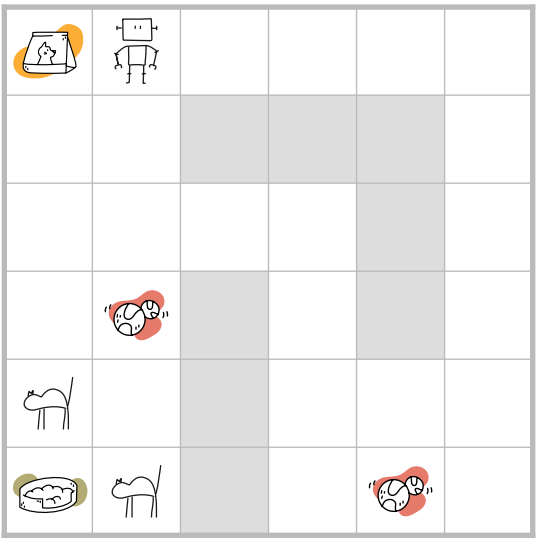
A cat story



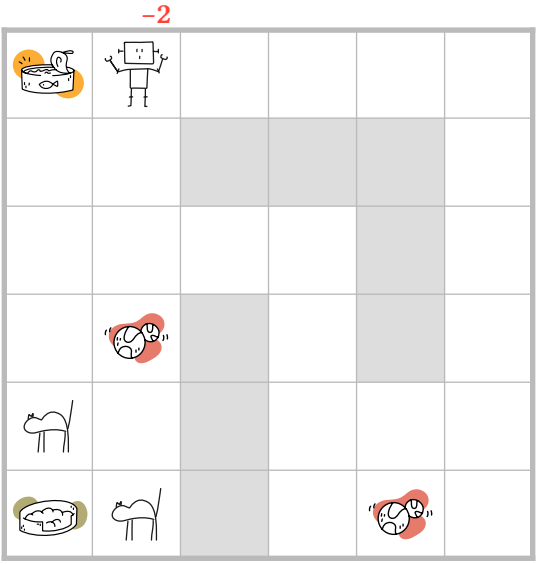
A cat story



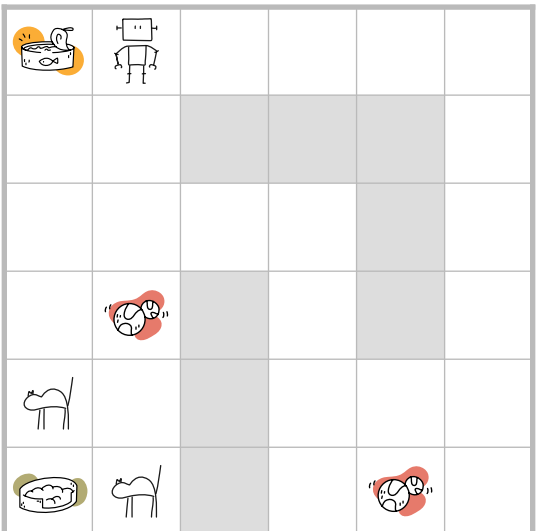
A cat story



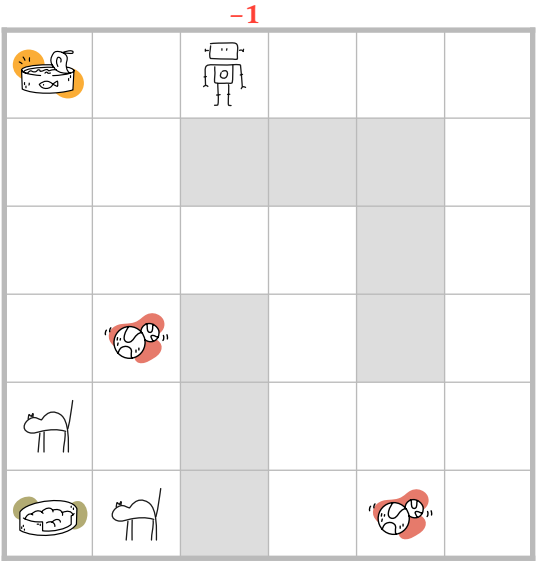
A cat story



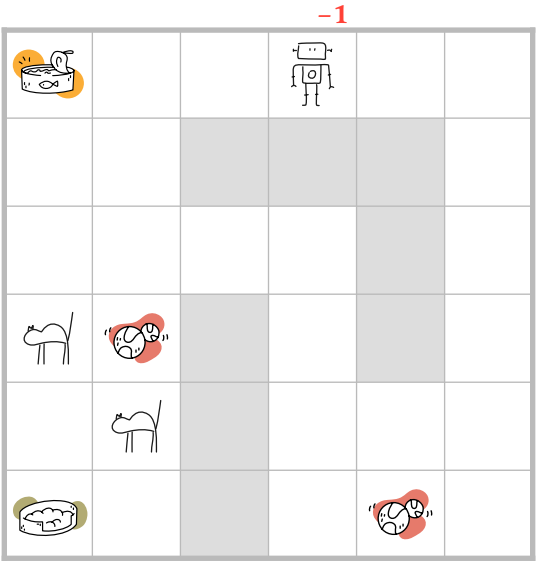
A cat story



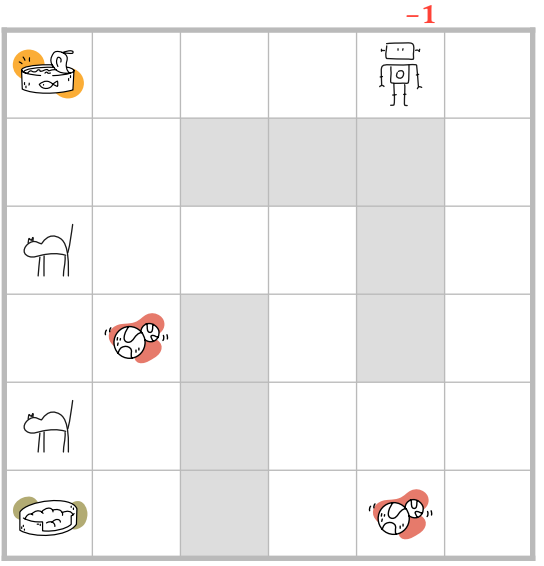
A cat story



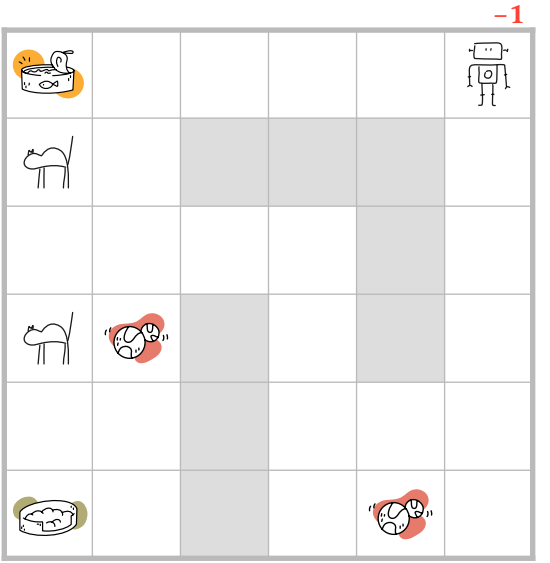
A cat story



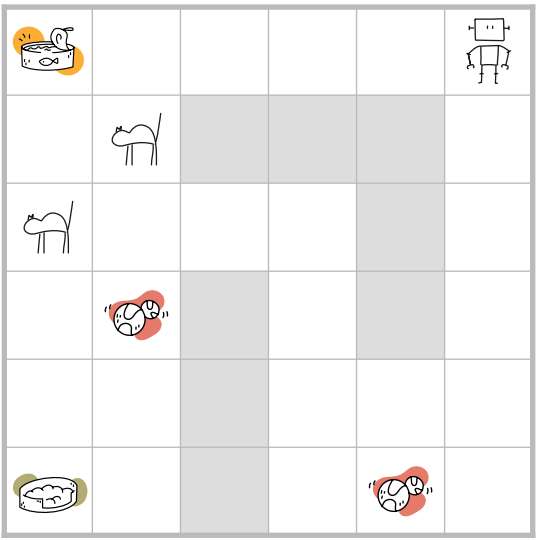
A cat story



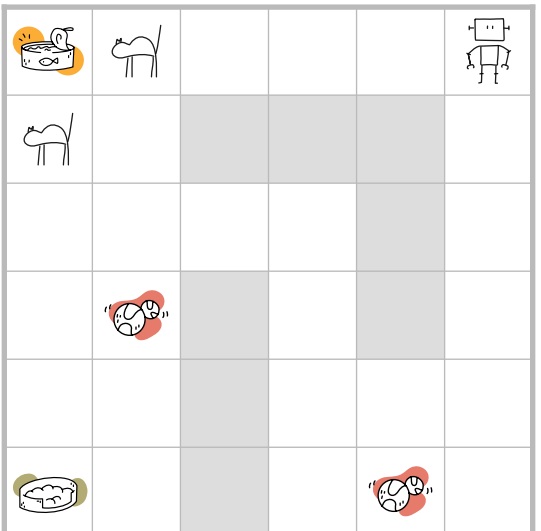
A cat story



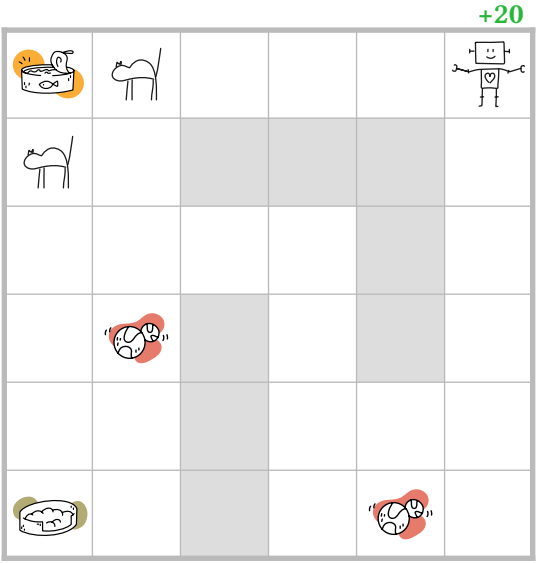
A cat story



A cat story

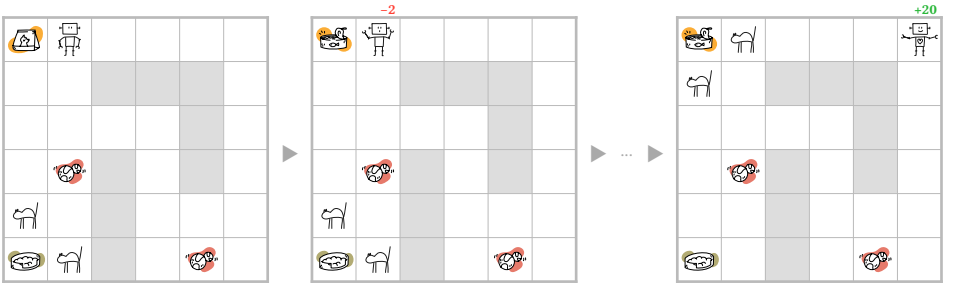


A cat story

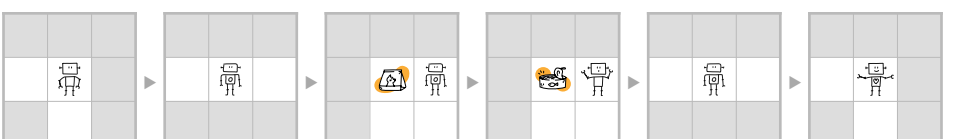


Partial observability

- The agent does not observe the full state of the environment.

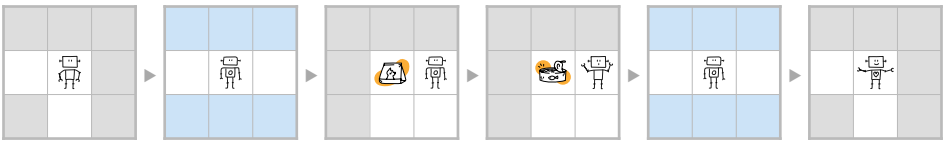


- Instead, its perception is limited to a partial observation.

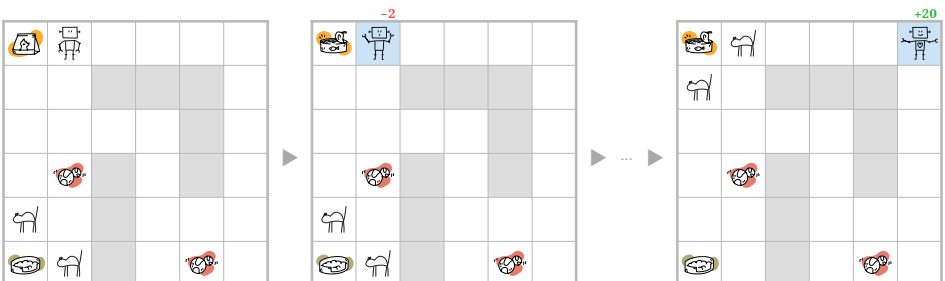


Past and future

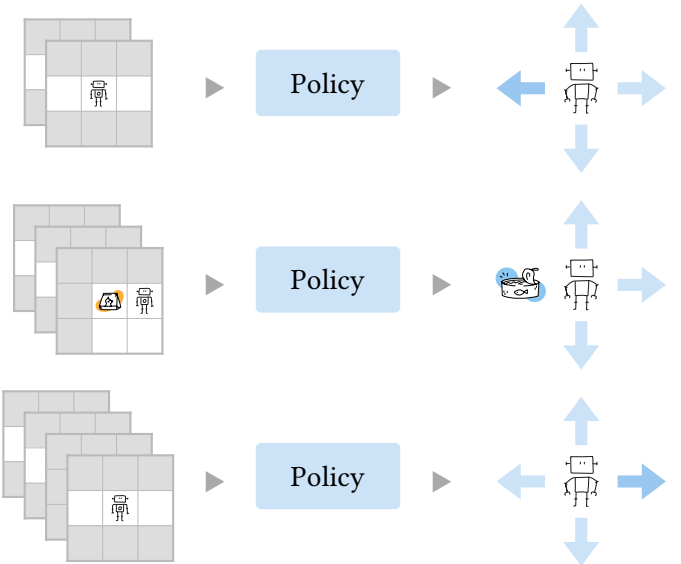
- We need to remember the state of things from the past.



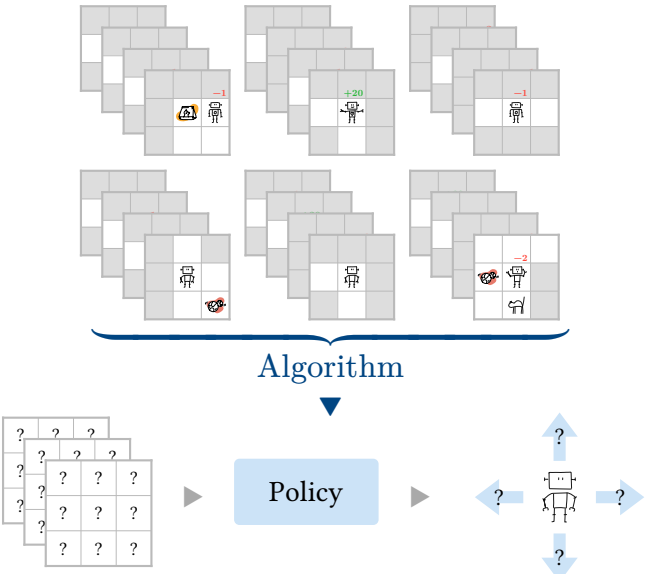
- We need to learn the impact of our actions on the future.



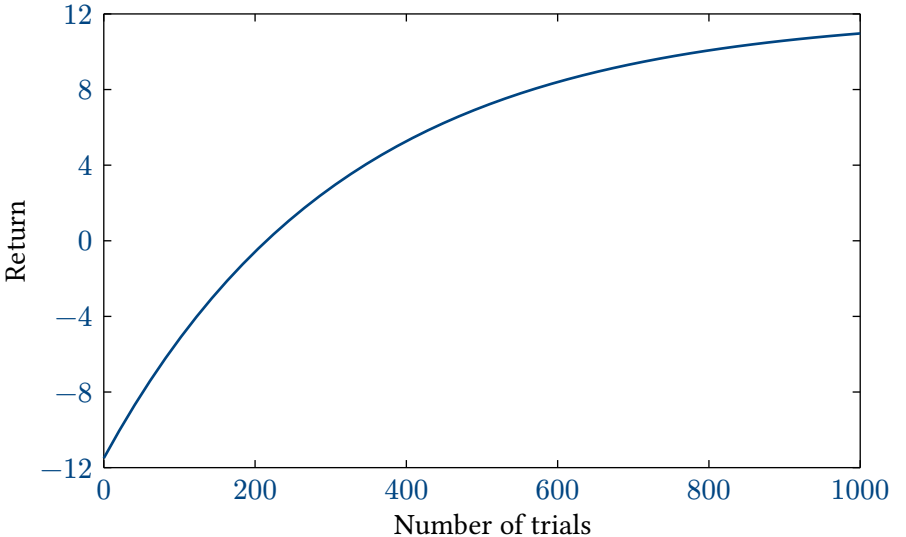
Intelligent behaviors



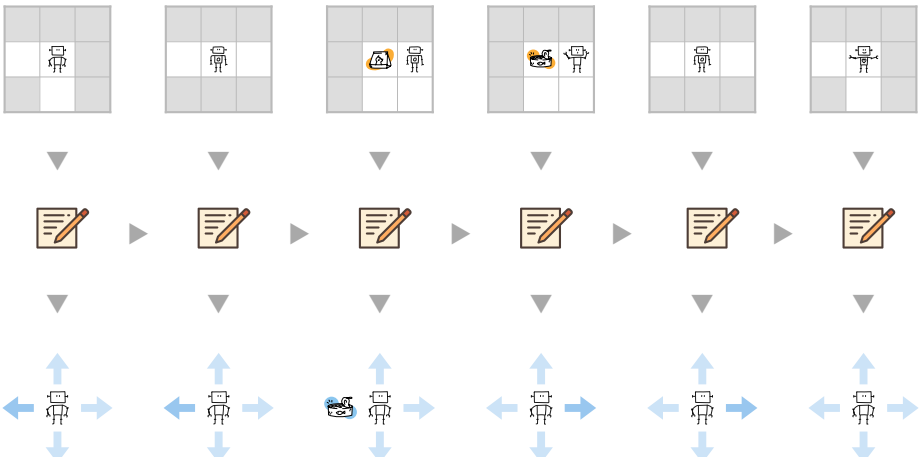
Learning intelligent behaviors



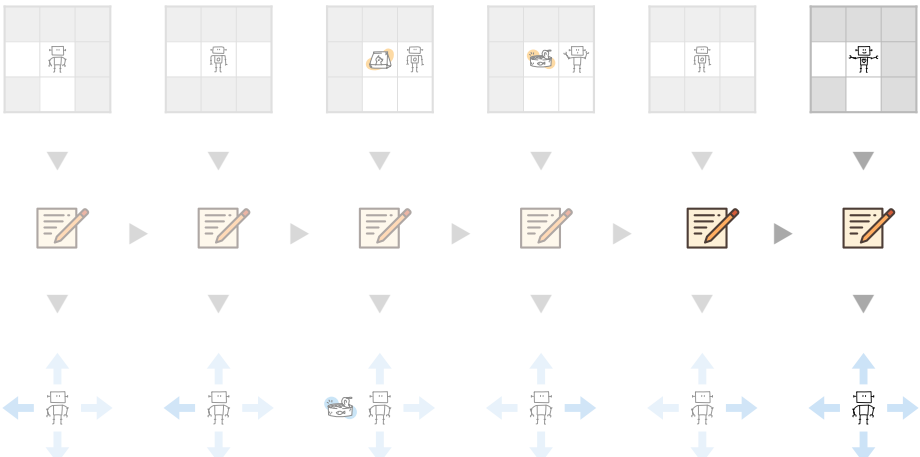
Learning curves



Sufficient memory

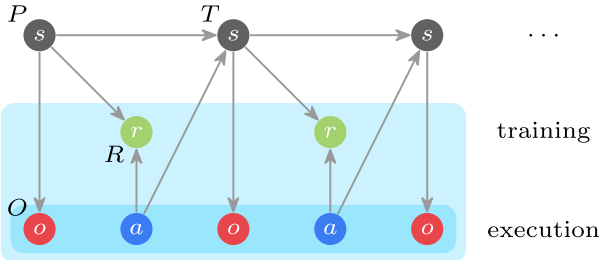


Sufficient memory



Reinforcement Learning under Partial Observability

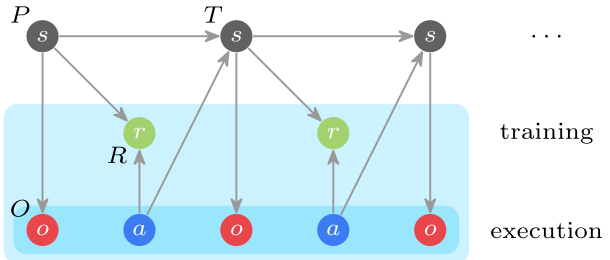
Partially observable Markov decision process



A **POMDP** is described by a model $\mathcal{P} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, T, R, O, P, \gamma)$.

- States $s_t \in \mathcal{S}$,
- Actions $a_t \in \mathcal{A}$,
- **Observations** $o_t \in \mathcal{O}$,
- Discount $\gamma \in [0, 1)$,
- Transition $T(s_{t+1} \mid s_t, a_t)$,
- Reward $r_t \sim R(\cdot \mid s_t, a_t)$,
- **Perception** $O(o_t \mid s_t)$,
- Initialisation $P(s_0)$.

History-dependent policies

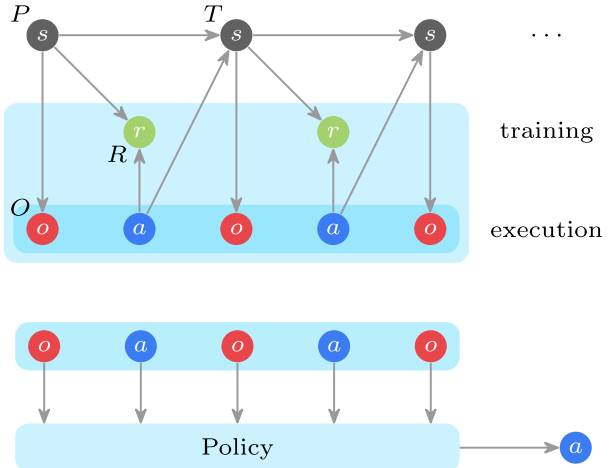


The **history** at time t is $h_t = (o_0, a_0, ..., o_t) \in \mathcal{H}$.

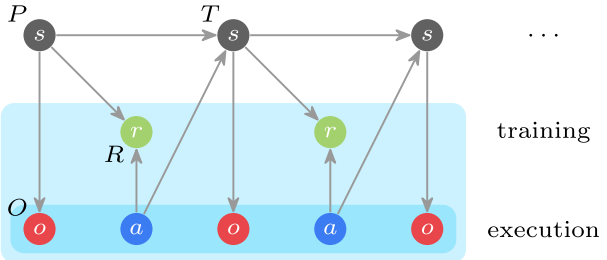
Definition 1: History-dependent policy.

A history-dependent policy $\eta \in \mathrm{H} = \mathcal{H} \to \Delta(\mathcal{A})$ is a mapping from histories to distributions over the actions, with density $\eta(a \mid h)$.

History-dependent policies



Optimal control under partial observability



The problem of **RL in POMDP** is to find an optimal history-dependent policy,

$$\eta^* \in \underset{\eta \in \mathrm{H}}{\operatorname{argmax}} \underbrace{\mathbb{E}^{\eta}\left[\sum_{t=0}^{\infty} \gamma^t R_t\right]}_{J(\eta)},$$

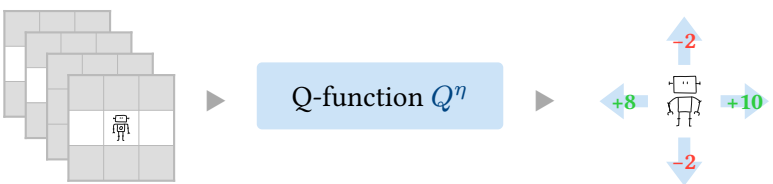
from samples $(o_0, a_0, r_0, ..., o_{t-1}, a_{t-1}, r_{t-1}, o_t)$.

Q-function of a policy

Definition 2: Q-function of a policy.

The **Q-function of a policy** $\eta \in \mathrm{H}$, or **critic**, is defined as,

$$Q^{\eta}(h, a) = \mathbb{E}^{\eta}\left[\sum_{t=0}^{\infty} \gamma^t R_t \,\middle|\, H_0 = h, A_0 = a\right].$$



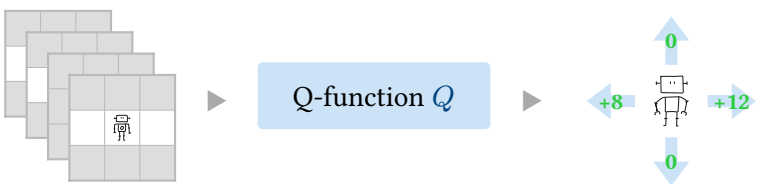
NB: $J(\eta) = \mathbb{E}[Q^{\eta}(H_0, A_0)]$.

Q-function

Definition 3: Q-function.

The **optimal Q-function**, or simply **Q-function**, is defined as,

$$Q(h, a) = \max_{\eta \in \mathrm{H}} Q^{\eta}(h, a) \big(= Q^{\eta^*}(h, a)\big).$$



NB: $\mathrm{Supp}(\eta^*(\cdot \mid h)) \subseteq \mathrm{argmax}_{a \in \mathcal{A}} Q(h, a)$.

History-dependent reinforcement learning

We use **function approximators** for the **policy** or **Q-function** estimation.

- Sliding window, transformer, recurrent networks, etc.

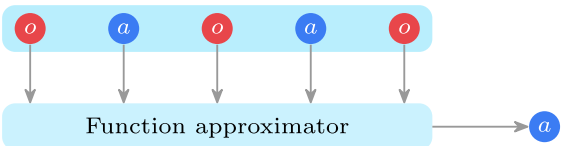


Fig. 1: Policy approximator.

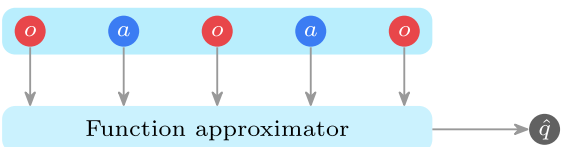


Fig. 2: Q-function approximator.

History-dependent reinforcement learning

We use **function approximators** for the **policy** or **Q-function** estimation.

- Sliding window, transformer, recurrent networks, etc.

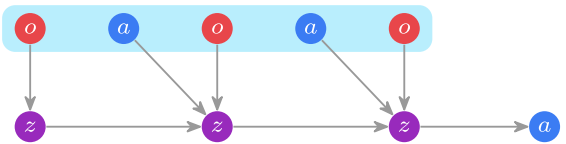


Fig. 1: Policy approximator.

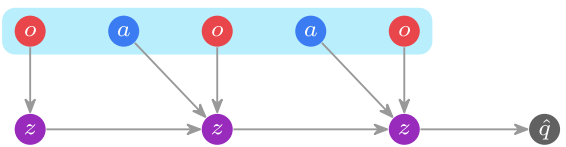


Fig. 2: Q-function approximator.

History-dependent reinforcement learning

We use **function approximators** for the **policy** or **Q-function** estimation.

- Sliding window, transformer, recurrent networks, etc.

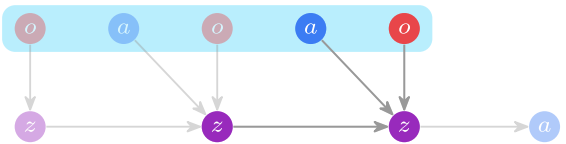


Fig. 1: Policy approximator.

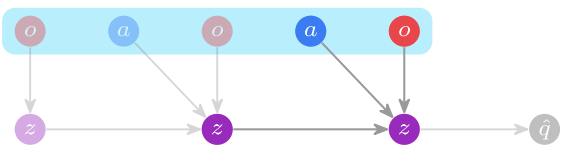


Fig. 2: Q-function approximator.

Reinforcement learning techniques

In this thesis, we considered three main reinforcement learning techniques:

1. **Policy-gradient** (and actor-critic) methods:

$$\nabla_\psi J(\eta_\psi) \approx \mathbb{E}^{d^{\eta_\psi}} \left[Q_\varphi^{\eta_\psi}(H, A) \nabla_\psi \log \eta_\psi(A \mid H) \right]$$

2. **Value-based** methods (Q-learning):

$$Q_\varphi(h, a) \approx \mathbb{E}\left[R + \gamma \max_{a' \in \mathcal{A}} Q_\varphi(H', a') \,\middle|\, H = h, A = a \right]$$

3. **Model-based** methods (supervised learning):

$$q_\theta(r, o' \mid h, a) \approx \Pr(r, o' \mid h, a)$$

Outline

Outline

I. Learning and Remembering
   - Learning for Remembering
   - Remembering for Learning

II. Leveraging Additional Information
   - Sufficiency through Additional Information
   - Learning Faster with Additional Information

III. Entangling Predictions and Decisions
   - Rolling the Dice First
   - Just Looking at the Dice

Outline

I. Learning and Remembering
- **Learning for Remembering**
- Remembering for Learning

II. Leveraging Additional Information
- **Sufficiency through Additional Information**
- **Learning Faster with Additional Information**

III. Entangling Predictions and Decisions
- Rolling the Dice First
- Just Looking at the Dice

I. Learning and Remembering

I.1. Learning for Remembering

Published in Transactions on Machine Learning Research (08/2022)

Recurrent networks, hidden states and beliefs in partially observable environments

Gaspard Lambrechts *gaspard.lambrechts@uliege.be*
Montefiore Institute, University of Liège

Adrien Bolland *adrien.bolland@uliege.be*
Montefiore Institute, University of Liège

Damien Ernst *dernst@uliege.be*
Montefiore Institute, University of Liège
LTCI, Telecom Paris, Institut Polytechnique de Paris

Reviewed on OpenReview: *https://openreview.net/forum?id=dkHfV3uB2l*

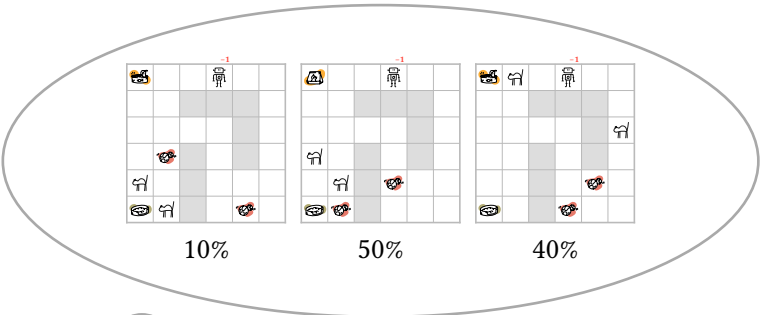
Abstract

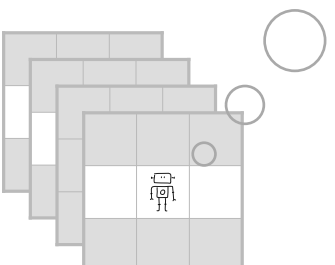
Reinforcement learning aims to learn optimal policies from interaction with environments whose dynamics are unknown. Many methods rely on the approximation of a value function to derive near-optimal policies. In partially observable environments, these functions depend on the complete sequence of observations and past actions, called the history. In this work, we show empirically that recurrent neural networks trained to approximate such value functions internally filter the posterior probability distribution of the current state given the history, called the belief. More precisely, we show that, as a recurrent neural network learns the Q-function, its hidden states become more and more correlated with the beliefs of state variables that are relevant to optimal control. This correlation is measured through their mutual information. In addition, we show that the expected return of an agent increases with the ability of its recurrent architecture to reach a high mutual information between its hidden states and the beliefs. Finally, we show that the mutual information between the hidden states and the beliefs of variables that are irrelevant for optimal control decreases through the learning process. In summary, this work shows that in its hidden states, a recurrent neural network approximating the Q-function of a partially observable environment reproduces a sufficient statistic from the history that is correlated to the relevant part of the belief for taking optimal actions.

1 Introduction

Latest advances in reinforcement learning (RL) rely heavily on the ability to approximate a value function (i.e., state or state-action value function). Modern RL algorithms have been shown to be able to produce

I.1. Learning for Remembering





Belief sufficiency

Definition 4: Belief of a history.

The belief $b = f(h)$ of a history $h \in \mathcal{H}$ is defined as,

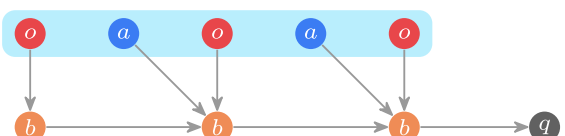
$$b(s) = \Pr(s \mid h).$$

Theorem 1: Belief recurrence.

$$f(h') = u(f(h), a, o').$$

Theorem 2: Belief sufficiency.

$$Q(h, a) = Q'(f(h), a).$$



$\Rightarrow$ If the belief is known, we can discard the history. But it is usually not known.

Recurrent reinforcement learning

In **recurrent Q-learning**, a recurrent approximator learns the Q-function.

- The RNN state $z = f_\theta(h)$ is a statistic of the history (memory).

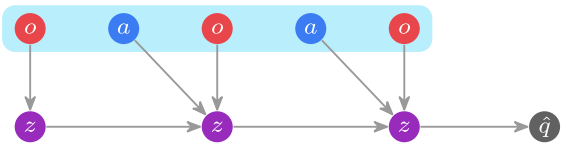


Fig. 3: RNN state and Q-function.

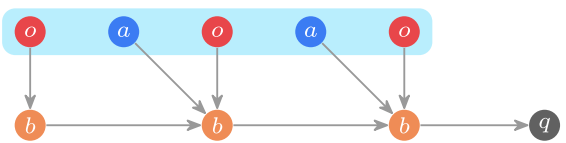


Fig. 4: Belief and Q-function.

$\Rightarrow$ Should RNN states encode the belief?

Recurrent reinforcement learning

We showed that **beliefs emerge** in RNN states during model-free recurrent Q-learning (LSTM, GRU, BRC, NBRC, MGU).

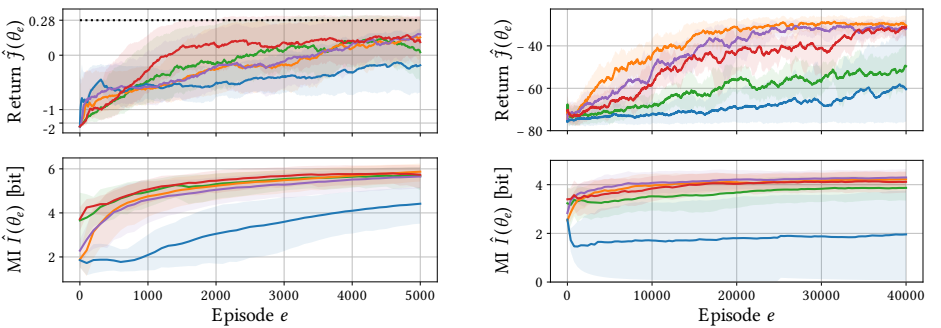


Fig. 5: Return and mutual information throughout training.

We estimate $\hat{I}(\theta) \approx I(z, b)$ under stationary distribution $p^{\eta_\theta}(h)$.

Recurrent reinforcement learning

The belief of **irrelevant state variable** is not encoded in RNN states (GRU).

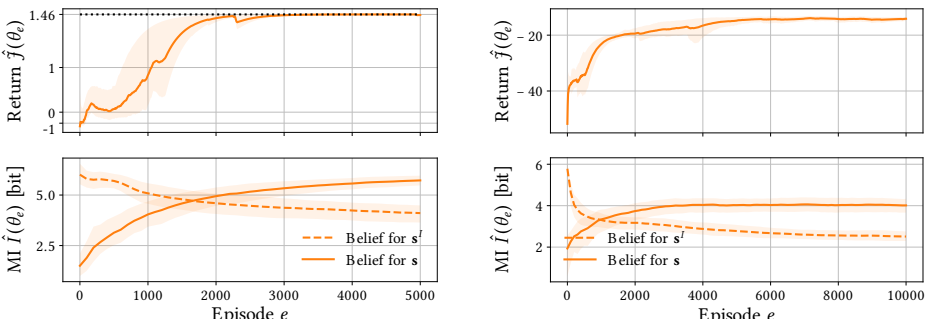


Fig. 6: Return and mutual information (relevant or irrelevant belief).

We estimate $\hat{I}(\theta) \approx I(z, b)$ under stationary distribution $p^{\eta_\theta}(h)$.

The statistic of the history encodes the belief of relevant state variables.

I.2. Remembering for Learning

Neural Networks 166 (2023) 645–669

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Warming up recurrent neural networks to maximise reachable multistability greatly improves learning

Gaspard Lambrechts [a,*,1], Florent De Geeter [a,1], Nicolas Vecoven [a,1], Damien Ernst [a,b], Guillaume Drion [a]

[a] Montefiore Institute, University of Liège, 10 allée de la découverte, Liège, 4000, Belgium
[b] LTCI, Telecom Paris, Institut Polytechnique de Paris, 19 place Marguerite Perey, Palaiseau, 91120, France

ARTICLE INFO

Article history:
Received 27 July 2022
Received in revised form 12 June 2023
Accepted 14 July 2023
Available online 7 August 2023

Keywords:
Recurrent neural network
Multistability
Initialisation procedure
Long-term memory
Warmup
Long time dependencies

ABSTRACT

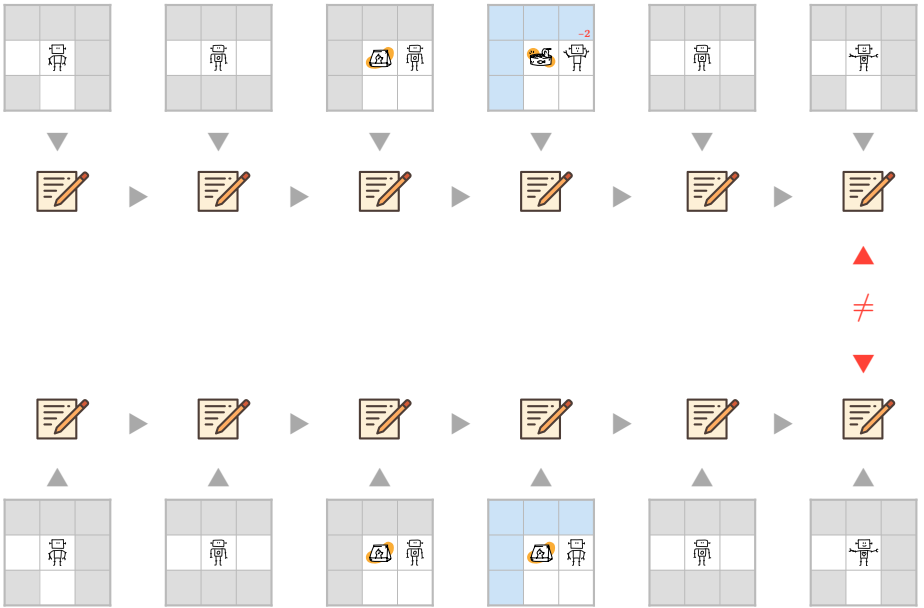
Training recurrent neural networks is known to be difficult when time dependencies become long. In this work, we show that most standard cells only have one stable equilibrium at initialisation, and that learning on tasks with long time dependencies generally occurs once the number of network stable equilibria increases; a property known as multistability. Multistability is often not easily attained by initially monostable networks, making learning of long time dependencies between inputs and outputs difficult. This insight leads to the design of a novel way to initialise any recurrent cell connectivity through a procedure called "warmup" to improve its capability to learn arbitrarily long time dependencies. This initialisation procedure is designed to maximise network reachable multistability, i.e., the number of equilibria within the network that can be reached through relevant input trajectories, in few gradient steps. We show on several information restitution, sequence classification, and reinforcement learning benchmarks that warming up greatly improves learning speed and performance, for multiple recurrent cells, but sometimes impedes precision. We therefore introduce a double-layer architecture initialised with a partial warmup that is shown to greatly improve learning of long time dependencies while maintaining high levels of precision. This approach provides a general framework for improving learning abilities of any recurrent cell when long time dependencies are present. We also show empirically that other initialisation and pretraining procedures from the literature implicitly foster reachable multistability of recurrent cells.

1. Introduction

Despite their performances and widespread use, recurrent neural networks (RNNs) are known to be blackbox models with extremely complex internal dynamics. A growing body of work has focused on understanding the internal dynamics of trained RNNs (Ceni, Ashwin, & Livi, 2020; Maheswaranathan, Williams, Golub, Ganguli, & Sussillo, 2019; Sussillo & Barak, 2013), providing invaluable intuition into the RNN prediction process. This viewpoint has already been used to understand the difficulties for RNNs to capture longer time dependencies (Bengio, Frasconi, 2013). This line of work has argued that locating such fixed points efficiently could provide insights into RNN dynamics and input–output properties. Here, we build upon this line of work by studying the impact of the number of reachable fixed points in an RNN on the ability to learn long time dependencies. Moreover, we highlight how maximising the number of reachable fixed points at initialisation can improve RNN learning, in particular in the presence of arbitrarily long dependencies.

More precisely, we introduce a fast-to-compute measure of the multistability of a network called variability amongst attractors (VAA). This measure gives the number of reachable attractors

I.2. Remembering for Learning



II. Leveraging Additional Information

II.1. Sufficiency through Additional Information

Informed POMDP: Leveraging Additional Information in Model-Based RL

Gaspard Lambrechts
gaspard.lambrechts@uliege.be
Montefiore Institute, University of Liège

Adrien Bolland
adrien.bolland@uliege.be
Montefiore Institute, University of Liège

Damien Ernst
dernst@uliege.be
Montefiore Institute, University of Liège
LTCI, Télécom Paris, Institut Polytechnique de Paris

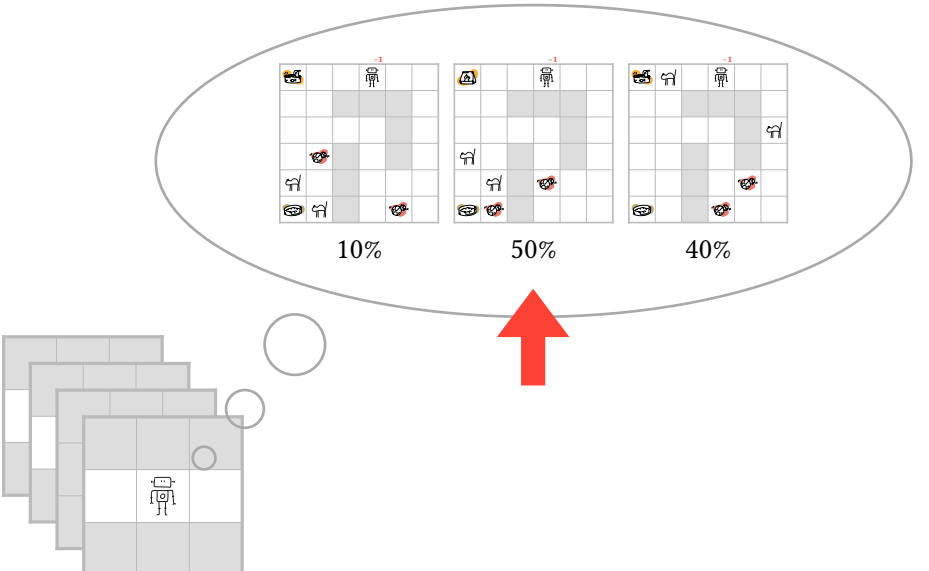
Abstract

In this work, we generalize the problem of learning through interaction in a POMDP by accounting for eventual additional information available at training time. First, we introduce the informed POMDP, a new learning paradigm offering a clear distinction between the information at training and the observation at execution. Next, we propose an objective that leverages this information for learning a sufficient statistic of the history for the optimal control. We then adapt this informed objective to learn a world model able to sample latent trajectories. Finally, we empirically show a learning speed improvement in several environments using this informed world model in the Dreamer algorithm. These results and the simplicity of the proposed adaptation advocate for a systematic consideration of eventual additional information when learning in a POMDP using model-based RL.

1 Introduction

Reinforcement learning (RL) aims to learn to act optimally through interaction with environments whose dynamics are unknown. A major challenge in this field is partial observability, where only a partial observation o of the Markovian state of the environment s is available for taking action a. Such an environment can be formalized as a partially observable Markov decision process (POMDP). In this context, an optimal policy $\eta(a|h)$ generally depends on the history h of all observations and previous actions, which grows linearly with time. Fortunately, it is theoretically possible to find a statistic $f(h)$ of the history h that is updated recurrently and that summarizes all relevant information to act optimally. Such a statistic is said to be recurrent and sufficient for the optimal control. Formally, a statistic $f(h)$ is recurrent when it is updated according to $f(h') = w(f(h), a, o')$ each time an action a is taken and a new observation o' is received, with $h' = (h, a, o')$. And a statistic $f(h)$ is sufficient for the optimal control when there exists an optimal policy $\eta(a|h) = g(a|f(h))$.

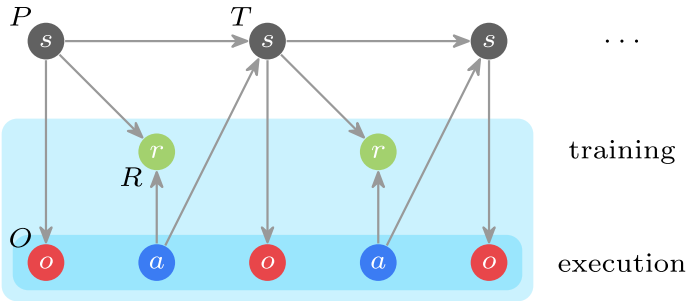
II.1. Sufficiency through Additional Information



A story of partial observability

Decision process	Execution	Training	Generality
MDP	s	s	**Too optimistic.**
POMDP	o	o	**Too pessimistic.**
Privileged POMDP	o	s	**Too optimistic.**
Informed POMDP	o	i	**Just right?**

Classical POMDP

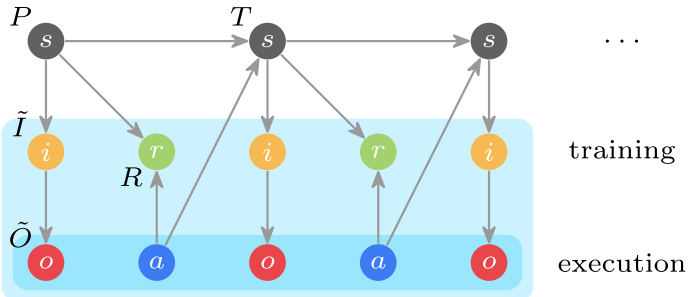


The problem of **RL in POMDP** is to find an optimal history-dependent policy

$$\eta^* \in \underset{\eta \in \mathrm{H}}{\operatorname{argmax}}\, \mathbb{E}^{\eta}\left[\sum_{t=0}^{\infty} \gamma^t R_t\right]$$

from samples $(o_0, a_0, r_0, ..., o_t)$.

Informed POMDP



The problem of **RL in POMDP** is to find an optimal history-dependent policy

$$\eta^* \in \underset{\eta \in \mathrm{H}}{\operatorname{argmax}}\, \mathbb{E}^{\eta}\left[\sum_{t=0}^{\infty} \gamma^t R_t\right]$$

from samples $(i_0, o_0, a_0, r_0, ..., i_t, o_t)$.

Sufficiency for optimal control

Fig. 7: Statistic $z = f(h)$ of the history h.

The history h is compressed into a statistic z by a function f.
$\Rightarrow$ It should summarize all relevant information to act optimally.

Definition 5: Sufficiency for optimal control.

A statistic $f : \mathcal{H} \to \mathcal{Z}$ is **sufficient for optimal control** if, and only if,

$$\max_g J(g \circ f) = \max_\eta J(\eta).$$

For example, the **belief** is a sufficient statistic.

Sufficiency in an informed POMDP

Theorem 3: Sufficiency of recurrent predictive statistics.

A statistic $f : \mathcal{H} \to \mathcal{Z}$ is **sufficient for optimal control** if it is,

(i) **recurrent**: $f(h') = u(f(h), a, o')$,

(ii) **predictive**: $p(r, i' \mid h, a) = p(r, i' \mid f(h), a)$.

It motivates the **informed world model** $q(r, i' \mid f(h), a)$ objective:

$$\max_{f,q} \mathop{\mathbb{E}}_{p(r,i' \mid h,a)} q(r, i' \mid f(h), a).$$

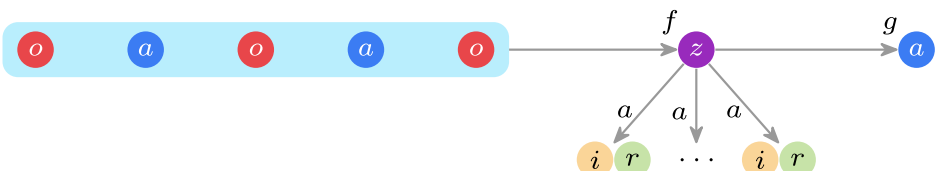
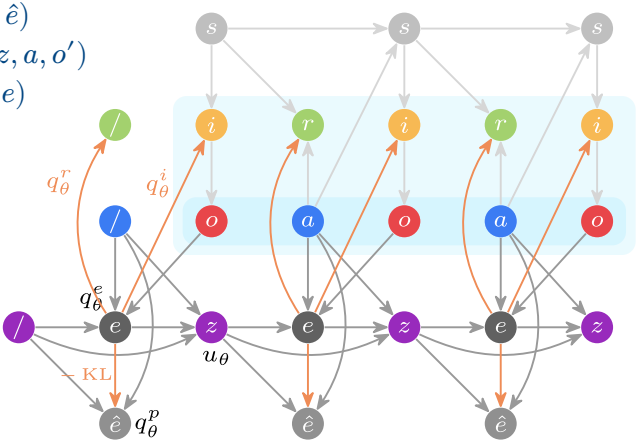


Fig. 8: Statistic $z = f(h)$ of the history h encoding the transition distribution.

Informed world model

The Informed Dreamer uses a variational recurrent neural network.

- Prior $\hat{e} \sim q^e(\cdot \mid z, a)$
- **Information** $\hat{\imath} \sim q^i(\cdot \mid z, \hat{e})$
  - Instead of observation $\hat{o} \sim q^o(\cdot \mid z, \hat{e})$
- Reward $\hat{r} \sim q^r(\cdot \mid z, \hat{e})$
- **Encoder** $e \sim q^e(\cdot \mid z, a, o')$
- Update $z' = u(z, a, e)$



Learning in imagination

From the learned world model, we can learn a latent policy $g_\varphi : \mathcal{Z} \to \Delta(\mathcal{A})$.

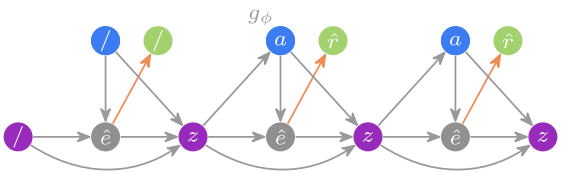


Fig. 9: Imagining trajectories with the Informed Dreamer and latent policy g_φ.

Informed Dreamer

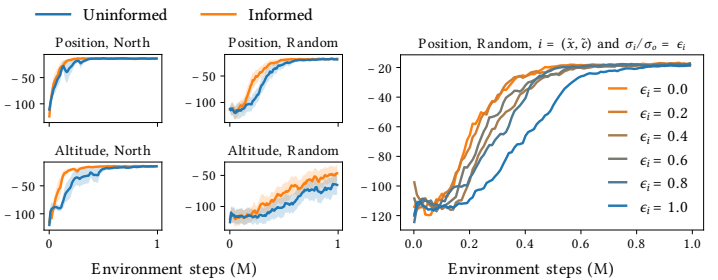


Fig. 10: Varying Mountain Hike

Informed Dreamer

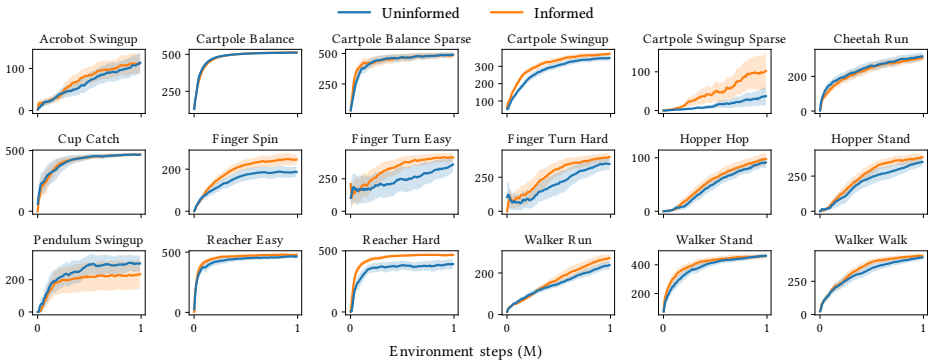


Fig. 11: Velocity DeepMind Control

Informed Dreamer

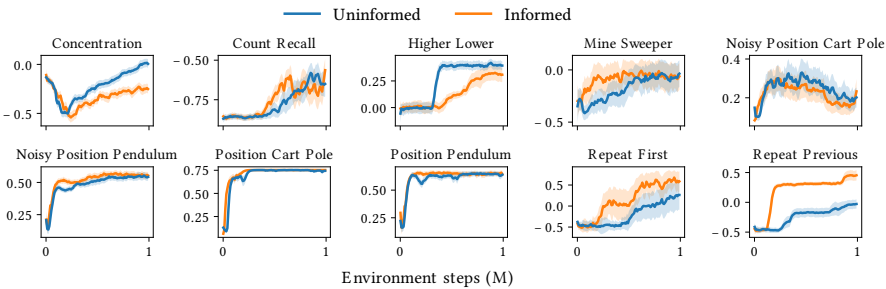


Fig. 12: Pop Gym

Informed world models provide faster policy learning than observational ones.

II.2. Learning Faster with Additional Information

A Theoretical Justification for Asymmetric Actor-Critic Algorithms

Gaspard Lambrechts [1 *] Damien Ernst [1] Aditya Mahajan [2]

Abstract

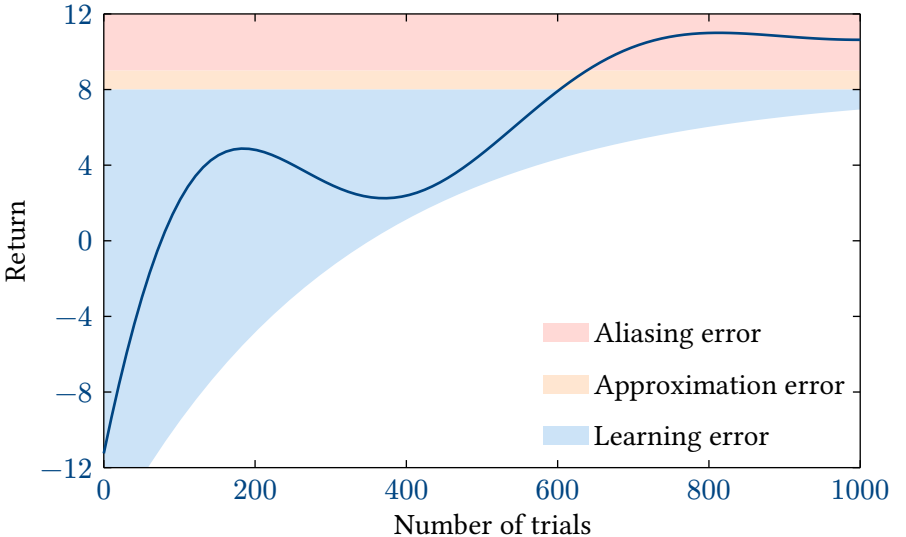
In reinforcement learning for partially observable environments, many successful algorithms were developed within the asymmetric learning paradigm. This paradigm leverages additional state information available at training time for faster learning. Although the proposed learning objectives are usually theoretically sound, these methods still lack a theoretical justification for their potential benefits. We propose such a justification for asymmetric actor-critic algorithms with linear function approximators by adapting a finite-time convergence analysis to this setting. The resulting finite-time bound reveals that the asymmetric critic eliminates an error term arising from aliasing in the agent state.

1. Introduction

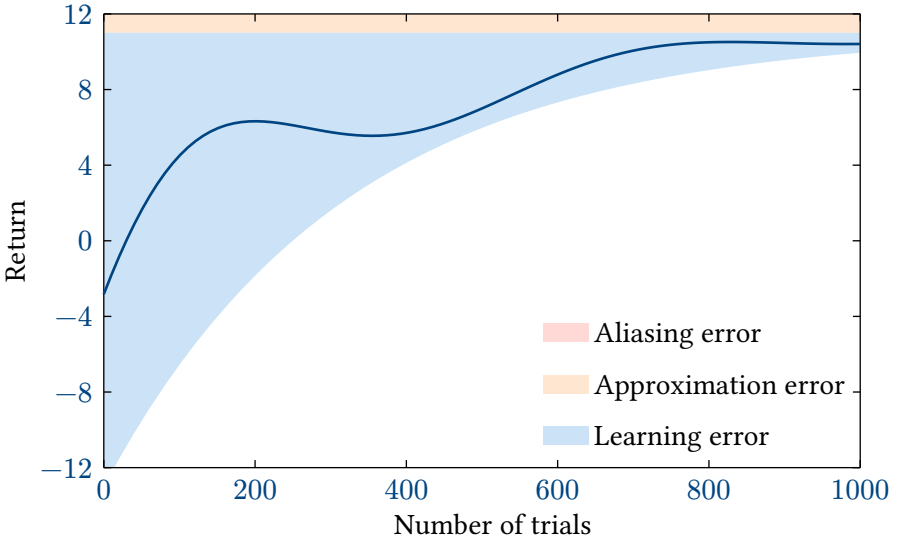
Reinforcement learning (RL) is an appealing framework for solving decision making problems, notably because it makes very few assumptions about the problem at hand. In its purest form, the promise of an RL algorithm is to learn an optimal behavior from interaction with an environment whose dynamics are unknown. More formally, an RL algorithm aims at learning a policy (i.e., a mapping from observations to actions) in order to maximize a reward signal from samples obtained by interacting with an environment. While RL has offered empirical successes for a plethora of challenging problems ranging from games to robotics (Mnih et al., 2015; Schrittwieser et al., 2020; Levine et al., 2015; Akkaya et al., 2019), most of these achievements have assumed full observability. A more realistic assumption is partial observability, where only a partial observation of the state of the environment is available for history-dependent policies, usually using a recurrent neural network to process histories (Bakker, 2001; Wierstra et al., 2010; Hausknecht & Stone, 2015; Heess et al., 2015; Zhang et al., 2016; Zhu et al., 2017). Given the difficulty of learning effective history-dependent policies, various auxiliary representation learning objectives have been proposed to compress the history into useful representations (Igl et al., 2018; Buesing et al., 2018; Guo et al., 2018; Gregor et al., 2019; Han et al., 2019; Hafner et al., 2019; Guo et al., 2020; Lee et al., 2020; Subramanian et al., 2022; Ni et al., 2024).

While these methods are theoretically able to learn optimal history-dependent policies, they learn solely the partial state observations, which can be too restrictive. Indeed, assuming the same partial observability at training time and execution time can be too pessimistic for many environments, notably for those that are simulated. This motivated the asymmetric learning paradigm, where additional state information available at training time is leveraged during the learning process of the history-dependent policy. Although the optimal policies obtained by asymmetric learning are theoretically equivalent to those learned by symmetric learning, the promise of asymmetric learning is to improve the convergence speed towards a near-optimal policy. Early approaches notably proposed to imitate a privileged policy conditioned on the state (Choudhury et al., 2018), or to use an asymmetric critic conditioned on the state (Pinto et al., 2018). These heuristic methods initially lacked a theoretical framework, and a recent line of work has focused on proposing theoretically grounded asymmetric learning objectives. First, imitation learning of a privileged policy was known to be suboptimal, and it was addressed by constraining the privileged policy so that its imitation results in an optimal policy for the partially observable environment (Warrington et al., 2021). Similarly, asymmetric actor-critic approaches were proven to provide biased gradients, and an unbiased actor-critic approach was proposed by introducing

II.2. Learning Faster with Additional Information



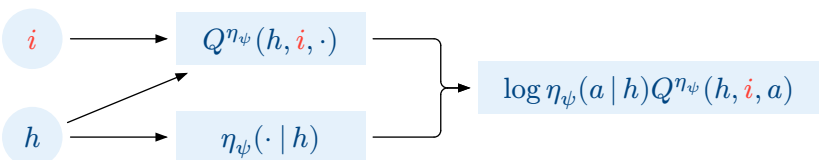
II.2. Learning Faster with Additional Information



Asymmetric actor-critic algorithm

Actor-critic algorithms are policy-gradient methods with a critic $Q_\varphi^{\eta_\psi} \approx Q^{\eta_\psi}$.

- The critic is **only used** for estimating the policy-gradient.
- It can be **informed** with additional information: $Q(h, a) \to Q(h, i, a)$.



$\Rightarrow$ Very effective, but **no theoretical justification for its benefits**.

Asymmetric actor-critic algorithm

- **State-informed**:
  - We study the case where $i = s$.
- **Fixed statistic:**
  - Fixed update $z' \sim U(\cdot \mid z, a, o')$, and policy $a \sim \pi(\cdot \mid z)$.
- **Finite state Q-functions:**
  - Asymmetric $\mathcal{Q}^\pi(s, z, a)$ and symmetric $Q^\pi(z, a)$.
- **Linear approximations:**
  - $\hat{\mathcal{Q}}^\pi_\beta(s, z, a) = \langle \beta, \varphi(s, z, a) \rangle$ and $\hat{Q}^\pi_\beta(z, a) = \langle \beta, \chi(z, a) \rangle$.
  - $\pi_\theta(a \mid z) \propto \exp(\langle \theta, \psi(z, a) \rangle)$.

Actor-critic algorithm

Algorithm 1: Asymmetric and symmetric actor-critic.

1. Initialize policy parameters ψ_0.
2. For $t = 1 \ldots T$
   1. Estimate $\hat{Q}^{\pi}_{\varphi} \approx Q^{\pi_\psi}$ or $\hat{Q}^{\pi}_{\chi} \approx Q^{\pi_\psi}$ (**TD learning**).
   2. Estimate $g_{t-1} \approx \nabla_\psi J\left(\pi_{\psi_{t-1}}\right)$ using Q_φ or Q_χ (**NPG estimation**).
   3. Update policy $\psi_t = \psi_{t-1} + \eta g_{t-1}$.
3. Return π_{ψ_T}

From the belief $b(s \mid h) = \Pr(s \mid h)$ and approximate belief $\hat{b}(s \mid z) = \Pr(s \mid z)$, we introduce a **measure of the aliasing** of the agent state z.

Aliasing measure.

$$\varepsilon_{\text{alias}} \propto \mathbb{E}\left[\left\| b(\cdot \mid h) - \hat{b}(\cdot \mid z) \right\|\right].$$

Finite-time bound for the critics

Theorem 4: Finite-time bound for asymmetric and symmetric Q-functions.

For any $\pi \in \Pi_{\mathcal{M}}$, and any $m \in \mathbb{N}$, we have for TD learning with $\alpha = \frac{1}{K}$,

$$\sqrt{\mathbb{E}\left[\left\|Q^\pi - \overline{Q}^\pi\right\|_{d^\pi}^2\right]} \leq \varepsilon_{\text{td}} + \varepsilon_{\text{app}} + \varepsilon_{\text{shift}}$$

$$\sqrt{\mathbb{E}\left[\left\|Q^\pi - \overline{Q}^\pi\right\|_{d^\pi}^2\right]} \leq \varepsilon_{\text{td}} + \varepsilon_{\text{app}} + \varepsilon_{\text{shift}} + \varepsilon_{\text{alias}}.$$

$$\varepsilon_{\text{td}} = \sqrt{\frac{4B^2 + \left(\frac{1}{1-\gamma} + 2B\right)^2}{2\sqrt{K}(1-\gamma^m)}}$$

$$\varepsilon_{\text{app}} = \frac{1+\gamma^m}{1-\gamma^m} \min_{f \in \mathcal{F}_\varphi^B} \|f - Q^\pi\|_{d^\pi}$$

$$\varepsilon_{\text{shift}} = \left(B + \frac{1}{1-\gamma}\right)\sqrt{\frac{2\gamma^m}{1-\gamma^m}\sqrt{\|d_m^\pi \otimes \pi - d^\pi \otimes \pi\|_{\text{TV}}}}$$

$$\varepsilon_{\text{alias}} = \frac{2}{1-\gamma}\left\|\mathbb{E}^\pi\left[\sum_{k=0}^{\infty}\gamma^{km}\left\|\hat{b}_{km} - b_{km}\right\|_{\text{TV}}\middle| Z_0 = \cdot, A_0 = \cdot\right]\right\|_{d^\pi}.$$

Finite-time bound for the actors

Theorem 5: Finite-time bound for asymmetric and symmetric NAC.

For any $(\mathcal{Z}, U)$, we have for NAC with $\alpha = \frac{1}{K}$, $\zeta = \frac{B\sqrt{1-\gamma}}{\sqrt{2N}}$, $\eta = \frac{1}{\sqrt{T}}$,

$$(1-\gamma) \min_{0 \leq t < T} \mathbb{E}[J(\pi^*) - J(\pi_t)] \leq \varepsilon_{\text{nac}} + \varepsilon_{\text{actor}} + \varepsilon_{\text{inf}} + \varepsilon_{\text{grad}} + \frac{1}{T} \sum_{t=0}^{T-1} \varepsilon_{\text{critic}}^{\pi_t},$$

$$\varepsilon_{\text{nac}} = \frac{B^2 + 2\log|A|}{2\sqrt{T}}$$

$$\varepsilon_{\text{actor}} = \overline{C}_\infty \sqrt{\frac{(2-\gamma)B}{(1-\gamma)\sqrt{N}}}$$

$$\varepsilon_{\text{inf,asym}} = 0 \quad \varepsilon_{\text{inf,sym}} = 2\mathbb{E}^{\pi^*}\left[\sum_{k=0}^{\infty} \gamma^k \left\|\hat{b}_k - b_k\right\|_{\text{TV}}\right]$$

$$\varepsilon_{\text{grad,asym}} = 2\overline{C}_\infty \sup_{0 \leq t < T} \sqrt{\min_w \mathcal{L}_t(w)} \quad \varepsilon_{\text{grad,sym}} = 2\overline{C}_\infty \sup_{0 \leq t < T} \sqrt{\min_w L_t(w)}$$

$$\varepsilon_{\text{critic,asym}}^{\pi_t} = 2\overline{C}_\infty \sqrt{6}(\varepsilon_{\text{td}} + \varepsilon_{\text{app}} + \varepsilon_{\text{shift}}) \quad \varepsilon_{\text{critic,sym}}^{\pi_t} = 2\overline{C}_\infty \sqrt{6}(\varepsilon_{\text{td}} + \varepsilon_{\text{app}} + \varepsilon_{\text{shift}} + \varepsilon_{\text{alias}})$$

Asymmetric learning is insensitive to aliasing in the statistic of the history.

III. Entangling Predictions and Decisions

III.1. Rolling the Dice First

ICML 2024 Next Generation of Sequence Modeling Architectures Workshop

Parallelizing Autoregressive Generation with Variational State Space Models

Gaspard Lambrechts * GASPARD.LAMBRECHTS@ULIEGE.BE
Yann Claes * Y.CLAES@ULIEGE.BE
Pierre Geurts P.GEURTS@ULIEGE.BE
Damien Ernst DERNST@ULIEGE.BE
Montefiore Institute, University of Liège

Abstract

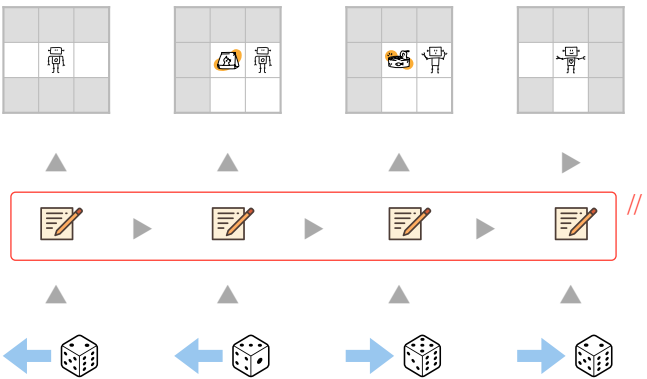
Attention-based models such as Transformers and recurrent models like state space models (SSMs) have emerged as successful methods for autoregressive sequence modeling. Although both enable parallel training, none enable parallel generation due to their autoregressiveness. We propose the variational SSM (VSSM), a variational autoencoder (VAE) where both the encoder and decoder are SSMs. Since sampling the latent variables and decoding them with the SSM can be parallelized, both training and generation can be conducted in parallel. Moreover, the decoder recurrence allows generation to be resumed without reprocessing the whole sequence. Finally, we propose the autoregressive VSSM that can be conditioned on a partial realization of the sequence, as is common in language generation tasks. Interestingly, the autoregressive VSSM still enables parallel generation. We highlight on toy problems (MNIST, CIFAR) the empirical gains in speed-up and show that it competes with traditional models in terms of generation quality (Transformer, Mamba SSM).

Keywords: Parallel, Autoregressive, Generation, VAE, SSM, VSSM

1. Introduction

Sequence modeling tasks, namely time-series forecasting and text generation, have gained in popularity and various types of architectures were designed to tackle such problems. Transformers were proven effective [17, 19], yet they nonetheless reprocess the complete sequence at each timestep, making generation less efficient. Recurrent neural networks (RNNs) [3, 8] update a hidden state based on new inputs at each timestep, enabling efficient generation. SSMs [9–11, 18], a recently introduced class of RNNs, enable parallel training thanks to their linear recurrence. Alternatively, several works adapt VAEs for sequential modeling. Some architectures integrate Transformers [13, 14] and enable parallel training, although little work [5] proposes models that can be conditioned on partial realizations (e.g., prompts). Conversely, variational RNNs (VRNNs) [4] loose parallelizabil-

III.1. Rolling the Dice First



III.2. Just Looking at the Dice

Efficient World Models through Parallel Imagination.

Gaspard Lambrechts
gaspard.lambrechts@uliege.be

Damien Ernst
dernst@uliege.be

April 7th, 2025

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

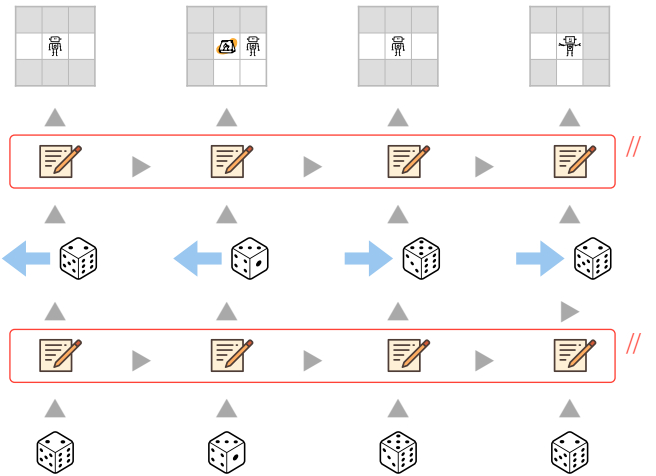
1 Introduction

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio. [Lambrechts et al., 2024].

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

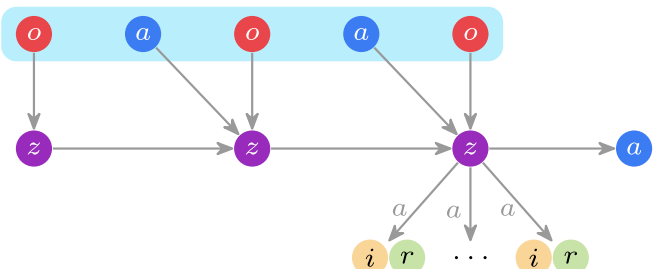
III.2. Just Looking at the Dice



A Matter of Abstractions

Conclusion

Instead of the history MDP, we should consider the **structure of the solution**, focusing on **recurrent approximators** that are **predictive of the future**.



I. **Good initial memory** learning to focus on the **relevant belief**,
II. **Predictive of the state**, or can be **augmented with state** at training,
III. Learned as **predictive latent** of the future, sampled and decoded in parallel.

Future perspectives

We developed techniques to **learn an optimal behavior** for any **given task**.
→ Could we instead **learn a behavior** that optimally **generalize to all tasks?**

Generalization is equivalent to optimal control under partial observability!