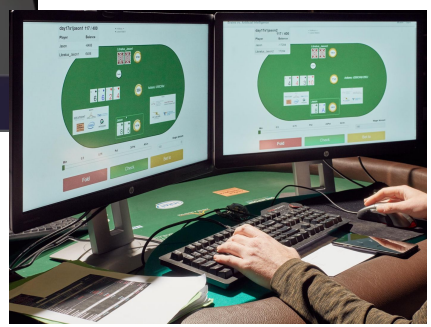
Contributions to deep reinforcement learning and its applications in smartgrids

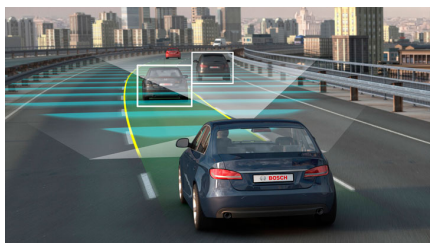
Vincent François-Lavet

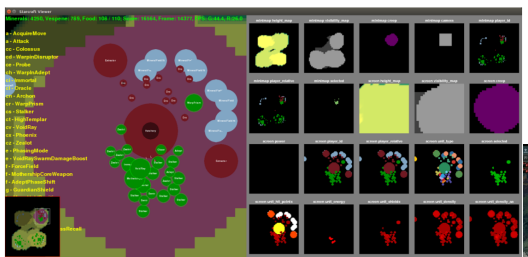
University of Liege, Belgium

September 11, 2017

Motivation

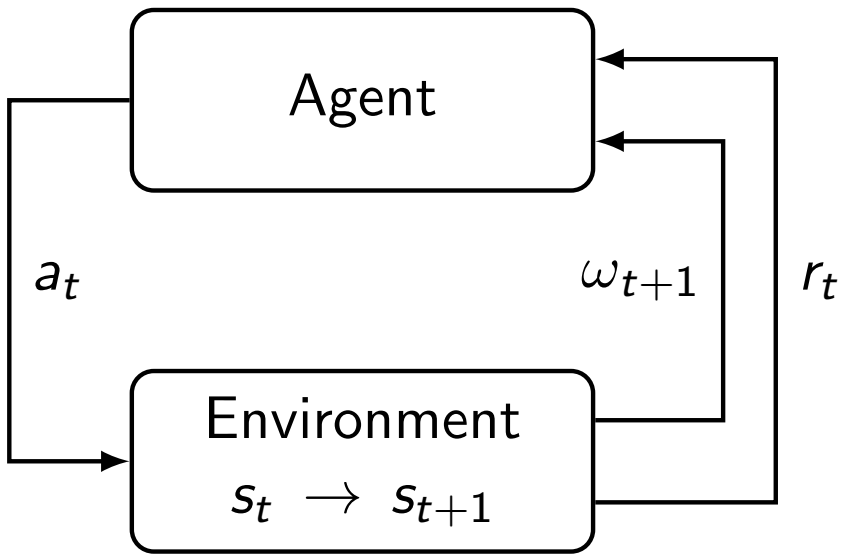






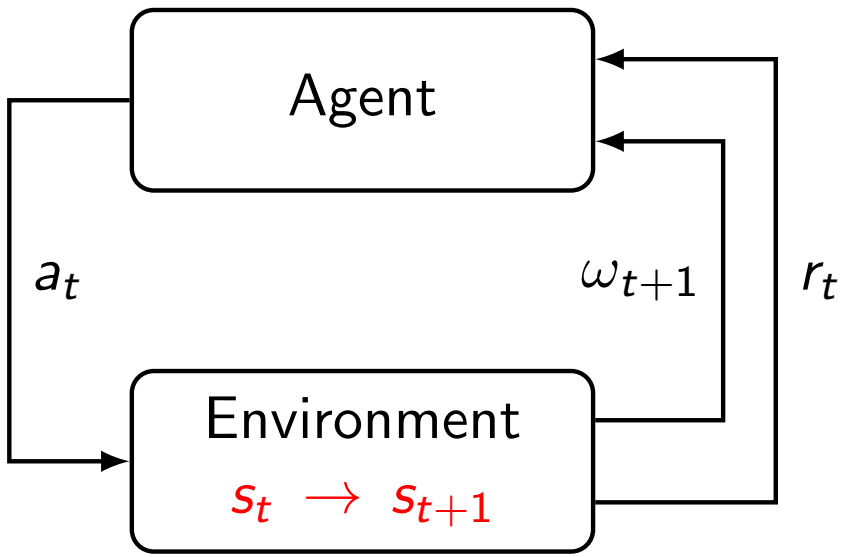
Objective

From experience in an environment,
an artificial agent
should be able to **learn** a sequential decision making task
in order **to achieve goals**.



Objective

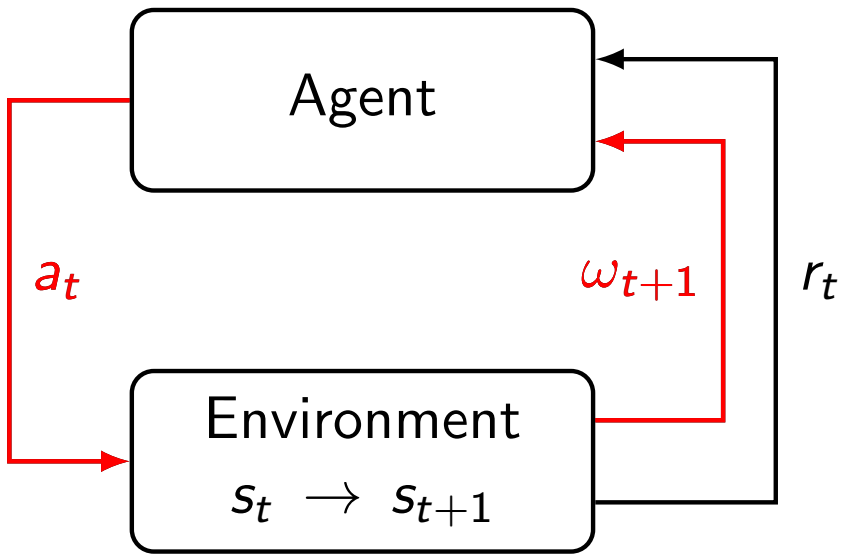
From experience in an environment,
an artificial agent
should be able to **learn** a sequential decision making task
in order **to achieve goals**.



transitions
are usually
stochastic

Objective

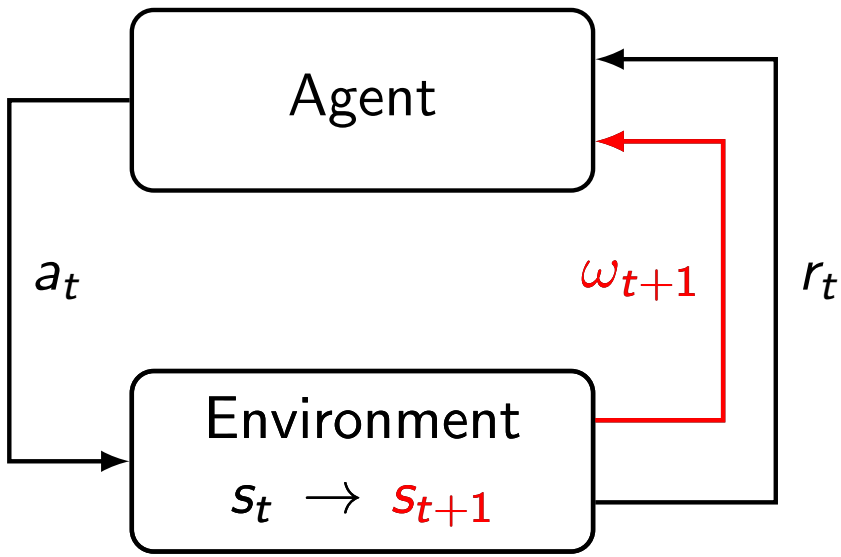
From experience in an environment,
an artificial agent
should be able to **learn** a sequential decision making task
in order **to achieve goals**.



Observations and
actions may be
high dimensional

Objective

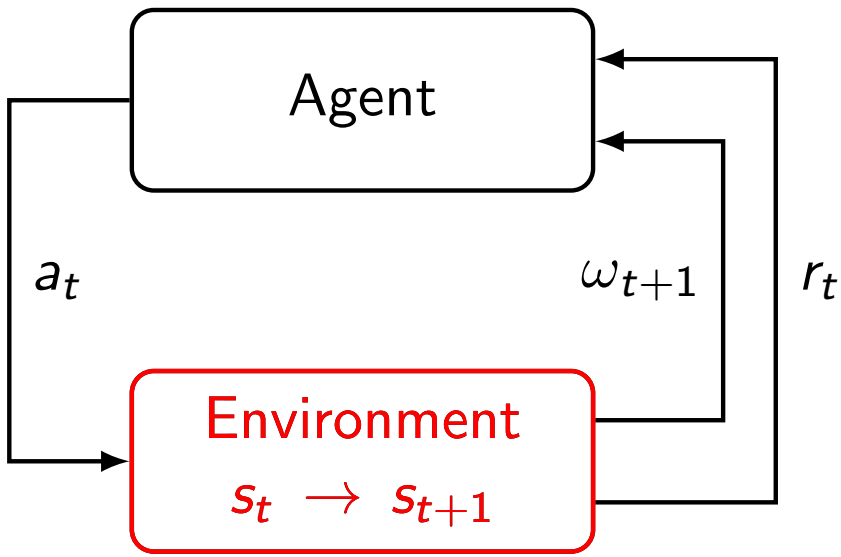
From experience in an environment,
an artificial agent
should be able to **learn** a sequential decision making task
in order **to achieve goals**.



Observations may not
provide full knowledge
of the underlying
state : $\omega_t \neq s_t$

Objective

From experience in an environment,
an artificial agent
should be able to **learn** a sequential decision making task
in order **to achieve goals**.



Experience may be constrained
(e.g., not access to an accurate simulator or limited data)

Outline

Introduction

Introduction

- Experience is gathered in the form of sequences of observations $\omega \in \Omega$, actions $a \in \mathcal{A}$ and rewards $r \in \mathbb{R}$:

$$\omega_0, a_0, r_0, ..., a_{t-1}, r_{t-1}, \omega_t$$

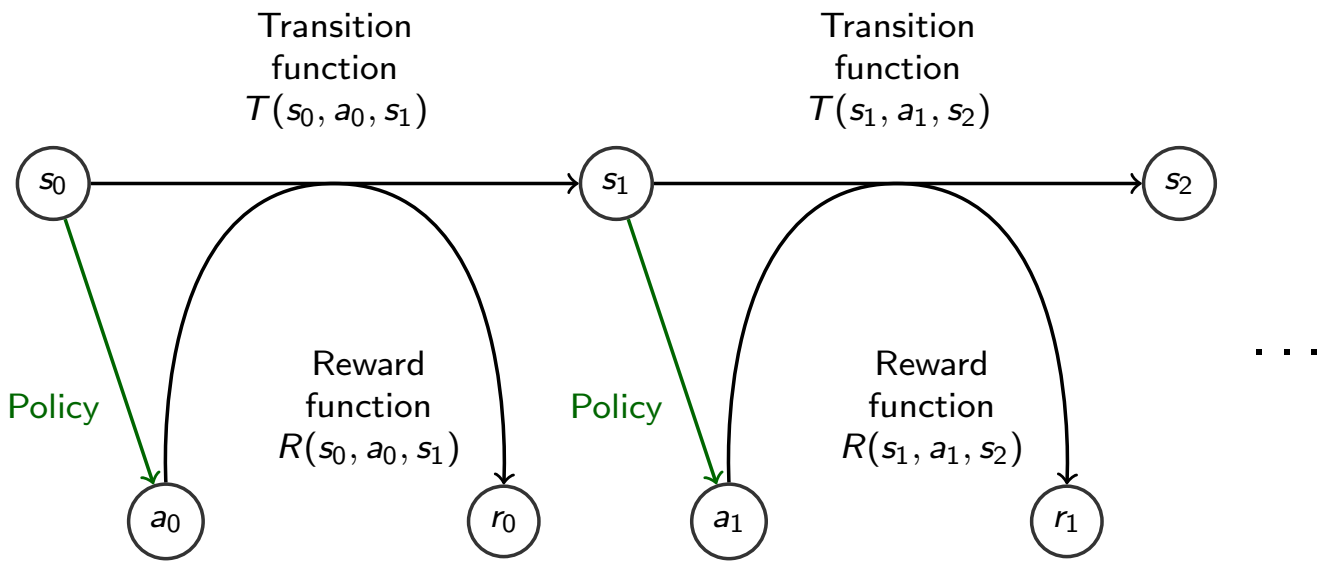
- In a fully observable environment, the state of the system $s_t \in \mathcal{S}$ is available to the agent.

$$s_t = \omega_t$$

Definition of an MDP

An MDP is a 5-tuple $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$ where :

- $\mathcal{S}$ is a finite set of states $\{1, \ldots, N_{\mathcal{S}}\}$,
- $\mathcal{A}$ is a finite set of actions $\{1, \ldots, N_{\mathcal{A}}\}$,
- $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function (set of conditional transition probabilities between states),
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathcal{R}$ is the reward function, where $\mathcal{R}$ is a continuous set of possible rewards in a range $R_{max} \in \mathbb{R}^+$ (e.g., $[0, R_{max}]$),
- $\gamma \in [0, 1)$ is the discount factor.



Performance evaluation

In an MDP $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$, the expected return $V^{\pi}(s) : \mathcal{S} \to \mathbb{R}$ $(\pi \in \Pi)$ is defined such that

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{k=0}^{H-1} \gamma^{k} r_{t+k} \mid s_t = s, \pi\right], \tag{1}$$

with $\gamma \leq 1 (< 1 \text{ if } H \to \infty)$.

From the definition of the expected return, the optimal expected return can be defined as

$$V^{*}(s) = \max_{\pi \in \Pi} V^{\pi}(s). \tag{2}$$

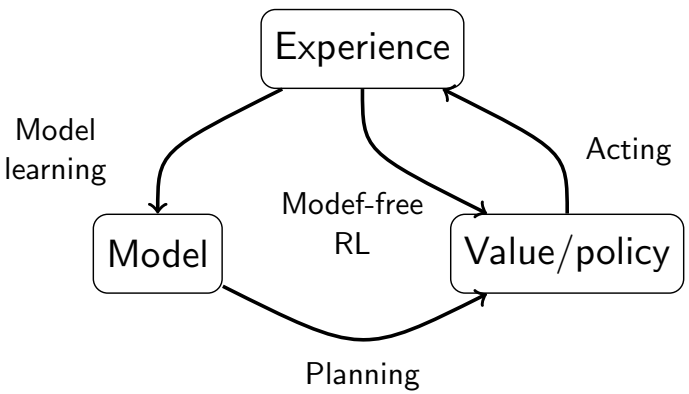
and the optimal policy can be defined as :

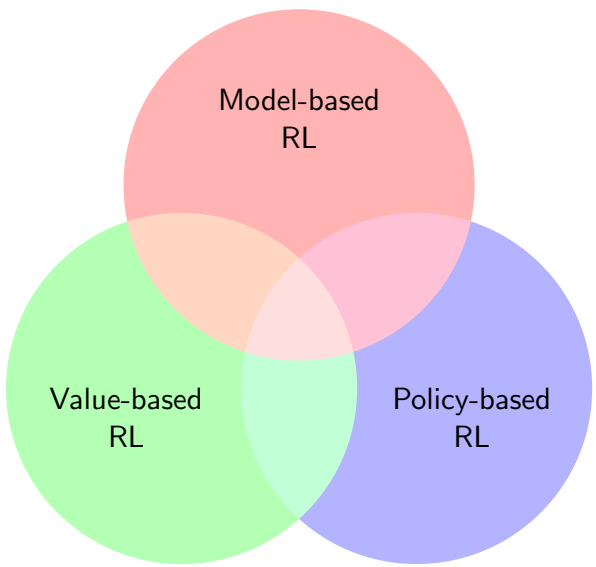
$$\pi^{*}(s) = \operatorname*{argmax}_{\pi \in \Pi} V^{\pi}(s). \tag{3}$$

Overview of deep RL

In general, an RL agent may include one or more of the following components :

- a representation of a value function that provides a prediction of how good is each state or each couple state/action,
- a direct representation of the policy $\pi(s)$ or $\pi(s, a)$, or
- a model of the environment in conjunction with a planning algorithm.





Deep learning has brought its generalization capabilities to RL.

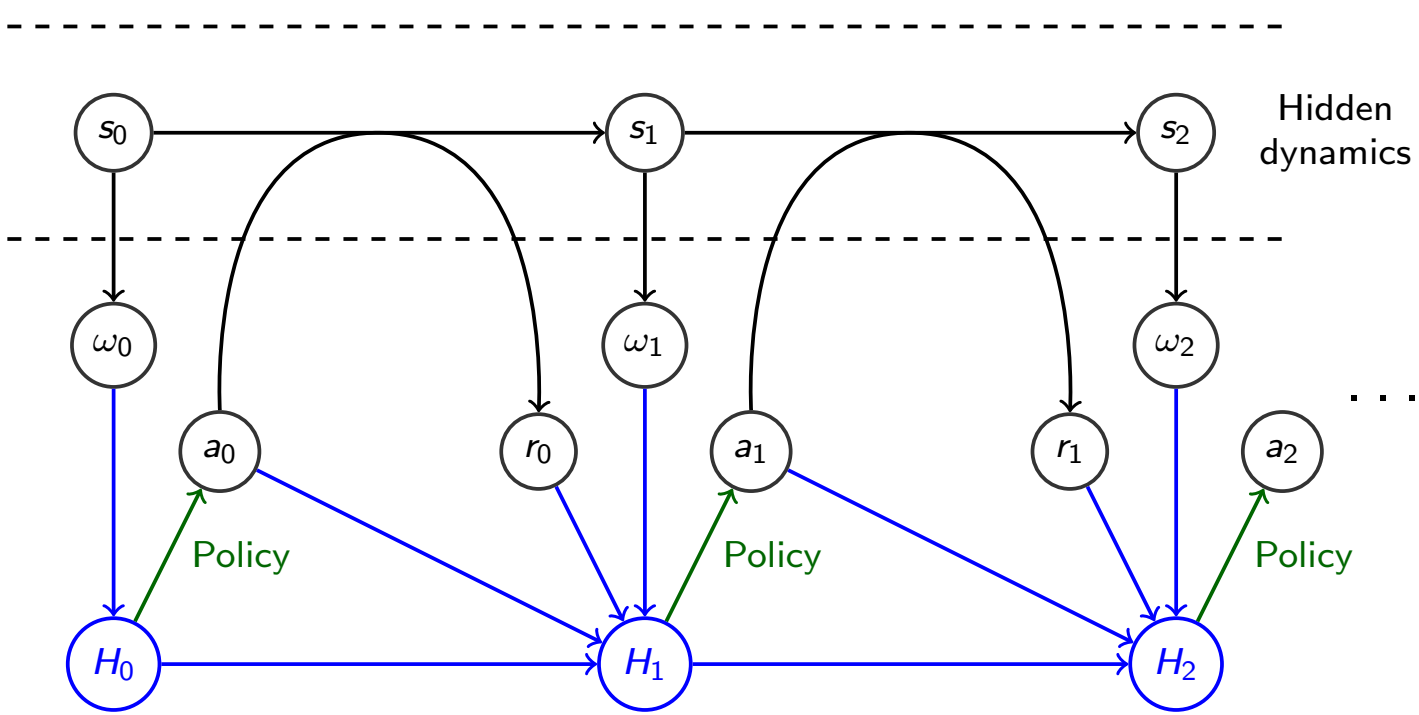
Contributions

Asymptotic bias and overfitting in the general partially observable case

Partial observability

In a partially observable environment, the agent has to rely on features from the history H_t

$$H_t = (\omega_0, r_0, a_0, ..., r_{t-1}, a_{t-1}, \omega_t) \in \mathcal{H}$$



We'll use a mapping $\phi : \mathcal{H} \to \phi(\mathcal{H})$, where $\phi(\mathcal{H}) = \{\phi(H)|H \in \mathcal{H}\}$ is of finite cardinality $|\phi(\mathcal{H})|$.

Partial observability

We consider a discrete-time POMDP model M defined as follows : A POMDP is a 7-tuple $(\mathcal{S}, \mathcal{A}, T, R, \Omega, O, \gamma)$ where :

- $\mathcal{S}$ is a finite set of states $\{1, \ldots, N_{\mathcal{S}}\}$,
- $\mathcal{A}$ is a finite set of actions $\{1, \ldots, N_{\mathcal{A}}\}$,
- $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function (set of conditional transition probabilities between states),
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathcal{R}$ is the reward function, where $\mathcal{R}$ is a continuous set of possible rewards in a range $R_{max} \in \mathbb{R}^+$ (e.g., $[0, R_{max}]$ without loss of generality),
- Ω is a finite set of observations $\{1, \ldots, N_{\Omega}\}$,
- $O : \mathcal{S} \times \Omega \to [0, 1]$ is a set of conditional observation probabilities, and
- $\gamma \in [0, 1)$ is the discount factor.

Importance of the feature space

Definition

The belief state $b(s|H_t)$ (resp. $b_\phi(s|\phi(H_t))$) is defined as the vector of probabilities where the i^{th} component ($i \in \{1, \ldots, N_{\mathcal{S}}\}$) is given by $\mathbb{P}(s_t = i \mid H_t)$ (resp. $\mathbb{P}(s_t = i \mid \phi(H_t))$), for $H_t \in \mathcal{H}$.

Definition

A mapping $\phi_0 : \mathcal{H} \to \phi_0(\mathcal{H})$ is a particular mapping ϕ such that $\phi_0(H)$ is a sufficient statistic for the POMDP M :

$$b(s|H_t) = b_{\phi_0}(s|\phi_0(H_t)), \forall H_t \in \mathcal{H}. \tag{4}$$

Definition

A mapping $\phi_\epsilon : \mathcal{H} \to \phi_\epsilon(\mathcal{H})$ is a particular mapping ϕ such that $\phi_\epsilon(H)$ is an ϵ-sufficient statistic for the POMDP M that satisfies the following condition with $\epsilon \geq 0$ and with the L_1 norm :

$$\|b_{\phi_\epsilon}(\cdot|\phi_\epsilon(H_t)) - b(\cdot|H_t)\|_1 \leq \epsilon, \forall H_t \in \mathcal{H}. \tag{5}$$

Expected return

For stationary and deterministic control policies $\pi \in \Pi : \phi(\mathcal{H}) \to \mathcal{A}$, we introduce $V_M^\pi(\phi(H))$ with $H \in \mathcal{H}$ as the expected return obtained over an infinite time horizon when the system is controlled using policy π in the POMDP M :

$$V_M^\pi(\phi(H)) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | \phi(H_t) = \phi(H), \pi, b(s_0)\right], \tag{6}$$

where $\mathbb{P}\big(s_{t+1}|s_t, \pi(\phi(H_t))\big) = T(s_t, \pi(\phi(H_t)), s_{t+1})$ and $r_t = R\big(s_t, \pi(\phi(H_t)), s_{t+1}\big)$.

Let π^* be an optimal policy in M defined as :

$$\pi^* \in \underset{\pi:\phi_0(\mathcal{H})\to\mathcal{A}}{\operatorname{argmax}} \; V_M^\pi(\phi_0(H_0)), \tag{7}$$

where H_0 is the distribution of initial observations.

Finite dataset

For any POMDP M, we denote by $D \sim \mathcal{D}$ a random dataset generated

- from N_{tr} (unordered) trajectories
- where a trajectory is the observable history $H_{N_l} \in \mathcal{H}_{N_l}$ obtained
- following a stochastic sampling policy π_s that ensures a non-zero probability of taking any action given an observable history $H \in \mathcal{H}$.

For the purpose of the analysis, we also introduce the asymptotic dataset D_∞ when $N_{tr} \to \infty$ and $N_l \to \infty$.

Frequentist approach

Definition

The frequentist-based augmented MDP $(\Sigma, \mathrm{A}, \hat{T}, \hat{R}, \Gamma)$ denoted $\hat{M}_{D,\phi}$ or simply $\hat{M}_D$ is defined with :

- the state space : $\Sigma = \phi(\mathcal{H})$,
- the action space : $\mathrm{A} = \mathcal{A}$,
- the estimated transition function : for $\sigma, \sigma' \in \Sigma$ and $a \in \mathrm{A}$, $\hat{T}(\sigma, a, \sigma')$ is the number of times we observe the transition $(\sigma, a) \times \sigma' \to [0, 1]$ in D divided by the number of times we observe (σ, a) ; if any (σ, a) has never been encountered in a dataset, we set $\hat{T}(\sigma, a, \sigma') = 1/|\Sigma|, \forall \sigma'$,
- the estimated reward function : for $\sigma, \sigma' \in \Sigma$ and $a \in \mathrm{A}$, $\hat{R}(\sigma, a, \sigma')$ is the mean of the rewards observed at (σ, a, σ') ; if any (σ, a, σ') has never been encountered in a dataset, we set $\hat{R}(\sigma, a, \sigma')$ to the average of rewards observed over the whole dataset D, and
- the discount factor $\Gamma \leq \gamma$ (called training discount factor).

Frequentist approach

We introduce $\mathcal{V}^{\pi}_{\hat{M}_D}(\sigma)$ with $\sigma \in \Sigma$ as the expected return obtained over an infinite time horizon when the system is controlled using a policy π s.t. $a_t = \pi(\sigma_t) : \Sigma \to \mathrm{A}, \forall t$ in the augmented decision process $\hat{M}_D$:

$$\mathcal{V}^{\pi}_{\hat{M}_D}(\sigma) = \mathbb{E}\left[\sum_{k=0}^{\infty} \Gamma^k \hat{r}_{t+k} | \sigma_t = \sigma, \pi, b(s_0)\right], \tag{8}$$

where $\hat{r}_t$ is a reward s.t. $\hat{r}_t = \hat{R}(\sigma_t, a_t, \sigma_{t+1})$ and the dynamics is given by $\mathbb{P}(\sigma_{t+1}|\sigma_t, a_t) = \hat{T}(\sigma_t, a_t, \sigma_{t+1})$.

Definition

The frequentist-based policy $\pi_{D,\phi}$ is an optimal policy of the augmented MDP $\hat{M}_D$ defined as : $\pi_{D,\phi} \in \underset{\pi:\Sigma\to\mathrm{A}}{\operatorname{argmax}} \mathcal{V}^{\pi}_{\hat{M}_D}(\sigma_0)$ where $\sigma_0 = \phi(H_0)$.

Bias-overfitting tradeoff

Let us now decompose the error of using a frequentist-based policy $\pi_{D,\phi}$:

$$\mathop{\mathbb{E}}_{D\sim\mathcal{D}}\left[V_M^{\pi^*}(\phi_0(H)) - V_M^{\pi_{D,\phi}}(\phi(H))\right] =$$

$$\underbrace{\left(V_M^{\pi^*}(\phi_0(H)) - V_M^{\pi_{D_\infty,\phi}}(\phi(H))\right)}_{\substack{\text{asymptotic bias function of dataset } D_\infty \text{ (function of } \pi_s\text{)} \\ \text{and frequentist-based policy } \pi_{D_\infty,\phi} \text{ (function of } \phi \text{ and } \Gamma\text{)}}}$$

$$+ \quad \underbrace{\mathop{\mathbb{E}}_{D\sim\mathcal{D}}\left[V_M^{\pi_{D_\infty,\phi}}(\phi(H)) - V_M^{\pi_{D,\phi}}(\phi(H))\right]}_{\substack{\text{overfitting due to finite dataset D} \\ \text{in the context of dataset } D \text{ (function of } \pi_s, N_I, N_{tr}\text{)} \\ \text{and frequentist-based policy } \pi_{D,\phi} \text{ (function of } \phi \text{ and } \Gamma\text{)}}} \quad . \tag{9}$$

Bound on the asymptotic bias

Theorem

The asymptotic bias can be bounded as follows :

$$\max_{H \in \mathcal{H}} \left(V_M^{\pi^*}(\phi_0(H)) - V_M^{\pi_{D_\infty,\phi}}(\phi(H)) \right) \leq \frac{2\epsilon R_{max}}{(1-\gamma)^3}, \qquad (10)$$

where ϵ is such that the mapping $\phi(\mathcal{H})$ is an ϵ-sufficient statistics.

This bound is an original result based on the belief states (which was not considered in other works) via the ϵ-sufficient statistic.

Sketch of the proof

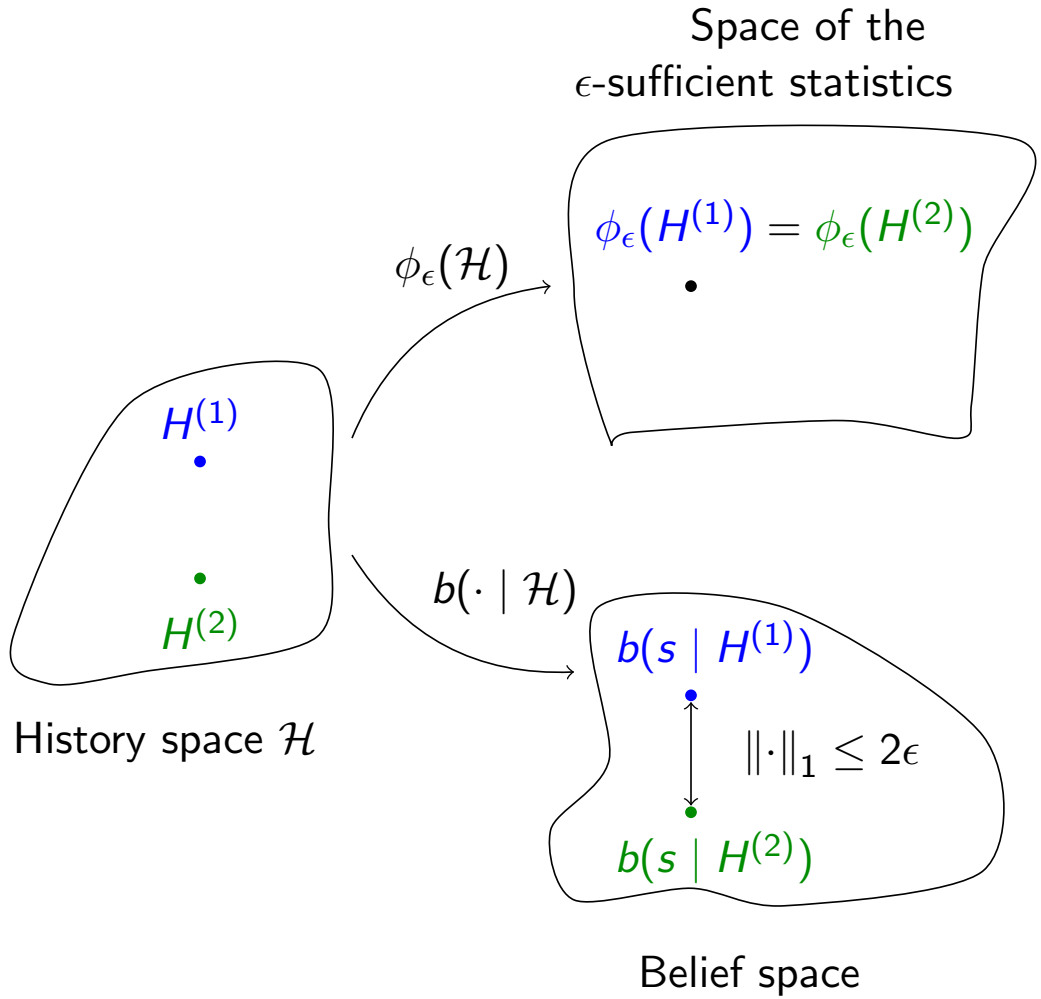


FIGURE: Illustration of the ϕ_ϵ mapping and the belief for $H^{(1)}, H^{(2)} \in \mathcal{H} : \phi_\epsilon(H^{(1)}) = \phi_\epsilon(H^{(2)})$.

Bound on the overfitting

Theorem

With the assumption that D has n transitions from any possible pair $(\phi(H), a) \in (\phi(\mathcal{H}), \mathrm{A})$. Then the overfitting term can be bounded as follows :

$$\max_{H \in \mathcal{H}} \left(V_M^{\pi_{D_\infty, \phi}}(\phi(H)) - V_M^{\pi_{D,\phi}}(\phi(H)) \right) \leq \frac{2R_{max}}{(1-\gamma)^2} \sqrt{\frac{1}{2n} ln \left(\frac{2|\phi(\mathcal{H})||\mathrm{A}|^{1+|\phi(\mathcal{H})|}}{\delta} \right)}, \tag{11}$$

with probability at least $1 - \delta$.

Sketch of the proof :

1. Find a bound between value functions estimated in different environments but following the same policy.
2. A bound in probability using Hoeffding's inequality can be obtained.

Experiments

Sample of N_P POMDPs from a distribution $\mathcal{P}$:

- transition functions $T(\cdot, \cdot, \cdot)$
  - non-zero entry in $[0, 1]$ with proba $1/4$, and it then ensures at least one non-zero for given (s, a) (then normalized),
- reward functions $R(\cdot, \cdot, \cdot)$,
  - i.i.d uniformly in $[-1, 1]$,
- conditional observation probabilities $O(\cdot, \cdot)$.
  - probability to observe $o^{(i)}$ when being in state $s^{(i)}$ is equal to 0.5, while all other values are chosen uniformly randomly so that it is normalized for any s.

Experiments

For each POMDP P, estimate of the average score μ_P :

$$\mu_P = \mathop{\mathbb{E}}_{D \sim \mathcal{D}_P} \mathop{\mathbb{E}}_{\text{rollouts}} \left[\sum_{t=0}^{N_I} \gamma^t r_t | s_0, \pi_{D,\phi} \right]. \tag{12}$$

Estimate of a parametric variance σ_P^2 :

$$\sigma_P^2 = \mathop{var}_{D \sim \mathcal{D}_P} \mathop{\mathbb{E}}_{\text{rollouts}} \left[\sum_{t=0}^{N_I} \gamma^t r_t | s_0, \pi_{D,\phi} \right]. \tag{13}$$

- Trajectories truncated to a length of $N_I = 100$ time steps
- $\gamma = 1$ and $\Gamma = 0.95$
- 20 datasets $D \in \mathcal{D}_P$ where $\mathcal{D}_P$ is a probability distribution over all possible sets of n trajectories ($n \in [2, 5000]$) while taking uniformly random decisions
- 1000 rollouts for the estimators

Experiments with the frequentist-based policies

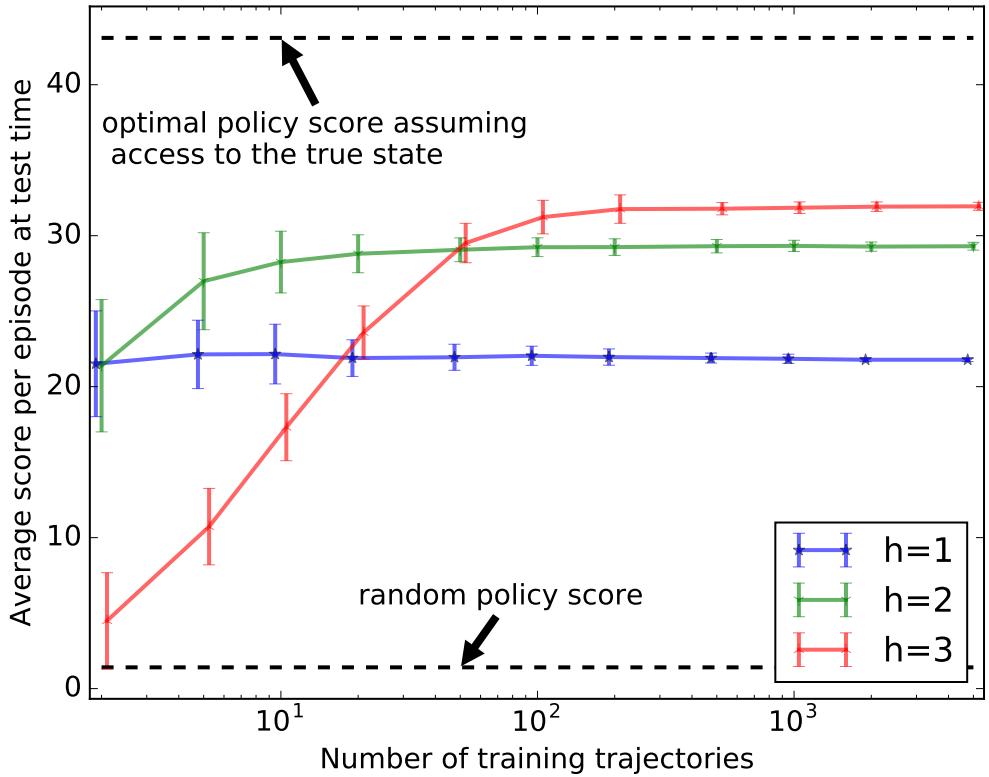


FIGURE: Estimated values of $\mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mu_P \pm \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \sigma_P$ computed from a sample of $N_P = 50$ POMDPs drawn from $\mathcal{P}$. The bars are used to represent the parametric variance. $N_{\mathcal{S}} = 5$, $N_{\mathcal{A}} = 2$, $N_{\Omega} = 5$.

When $h = 1$ (resp. $h = 2$) (resp. $h = 3$), only the current observation (resp. last two observations and last action) (resp. three last observations and two last actions) is (resp. are) used for building the state of the frequentist-based augmented MDP.

Experiments with a function approximator

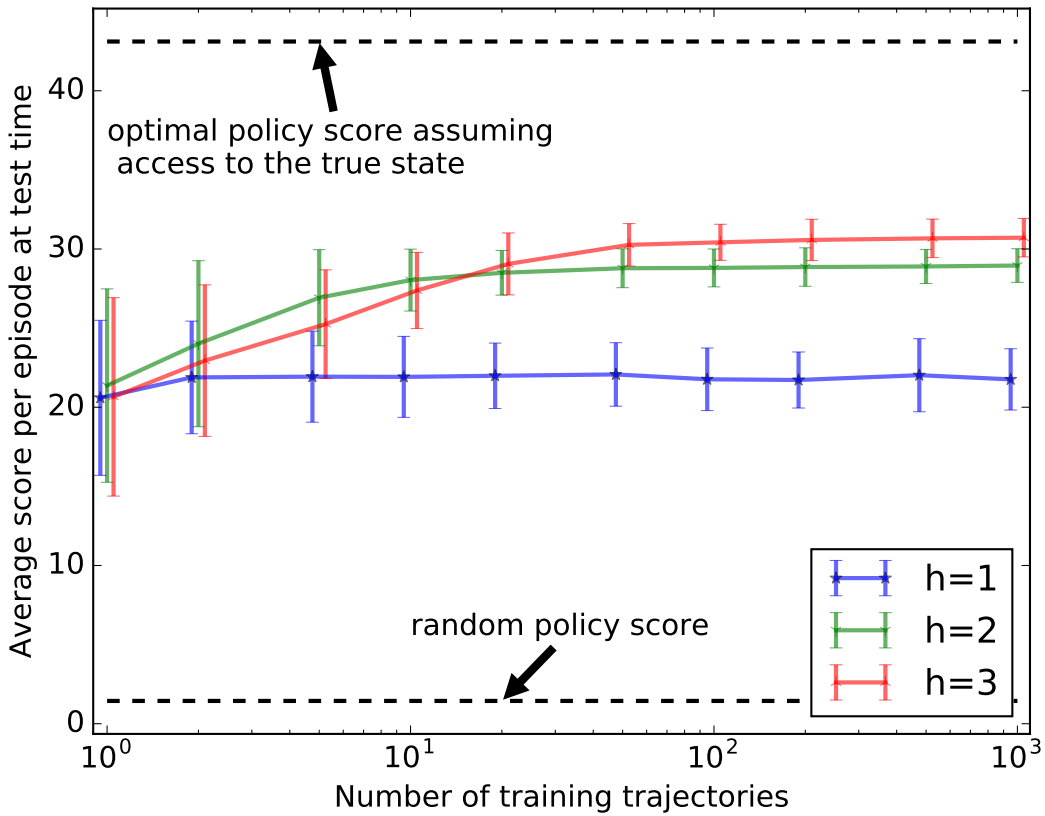


FIGURE: Estimated values of $\mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mu_P \pm \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \sigma_P$ computed from a sample of $N_P = 50$ POMDPs drawn from $\mathcal{P}$ with neural network as a function approximator. The bars are used to represent the parametric variance (when dealing with different datasets drawn from the distribution).

Experiments with the frequentist-based policies : effect of the discount factor

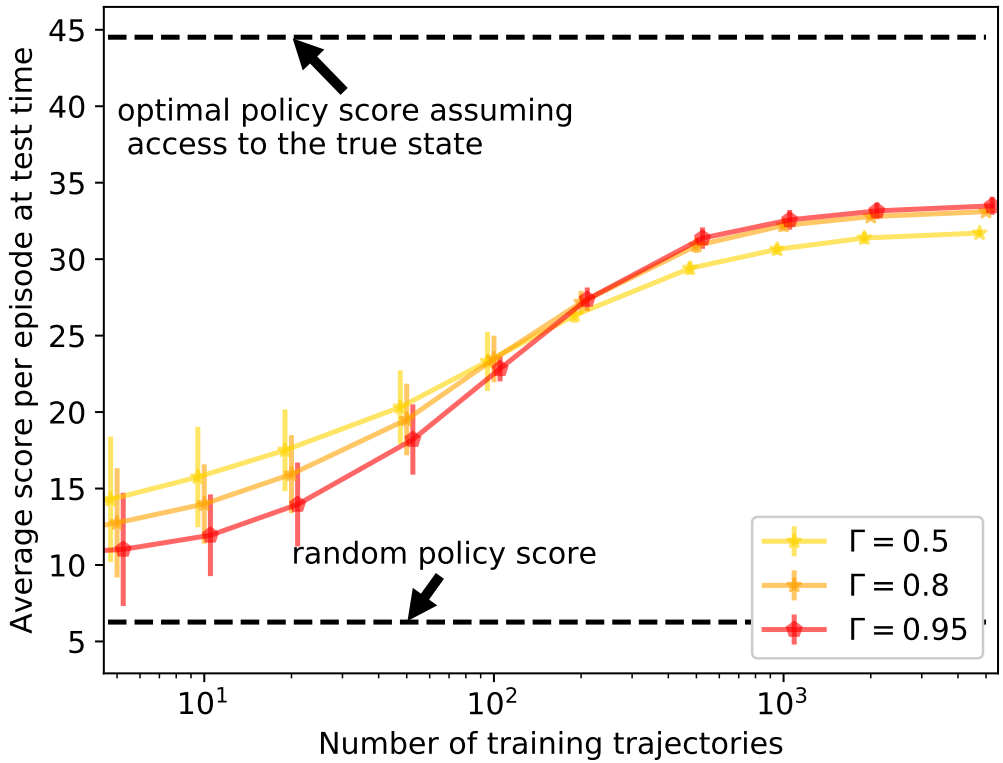


FIGURE: Estimated values of $\mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mu_P \pm \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \sigma_P$ computed from a sample of $N_P = 10$ POMDPs drawn from $\mathcal{P}$ with $N_{\mathcal{S}} = 8$ and $N_\Omega = 8$ ($h = 3$). The bars are used to represent the variance observed when dealing with different datasets drawn from a distribution ; note that this is not a usual error bar.

How to discount deep RL

Motivations

Effect of the discount factor in an online setting (value iteration algorithm).

- *Empirical studies of cognitive mechanisms in delay of gratification* : The capacity to wait longer for the preferred rewards seems to develop markedly only at about ages 3-4 ("marshmallow experiment").
- In addition to the role that the discount factor has on the bias-overfitting error, its study is also interesting because it plays a key role in the instabilities (and overestimations) of the value iteration algorithm.

Main equations of DQN

We investigate the possibility to work with an adaptive discount factor γ, hence targeting a moving optimal Q-value function :

$$Q^*(s,a) = \max_{\pi} \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots | s_t = s, a_t = a, \pi]$$

At every iteration, the current value $Q(s,a;\theta_k)$ is updated towards a target value

$$Y_k^Q = r + \gamma \max_{a' \in A} Q(s', a'; \theta_k^-). \tag{14}$$

where θ_k^- are updated every C iterations with the following assignment : $\theta_k^- = \theta_k$.

Example

FIGURE: Example of an ATARI game : Seaquest

Increasing discount factor

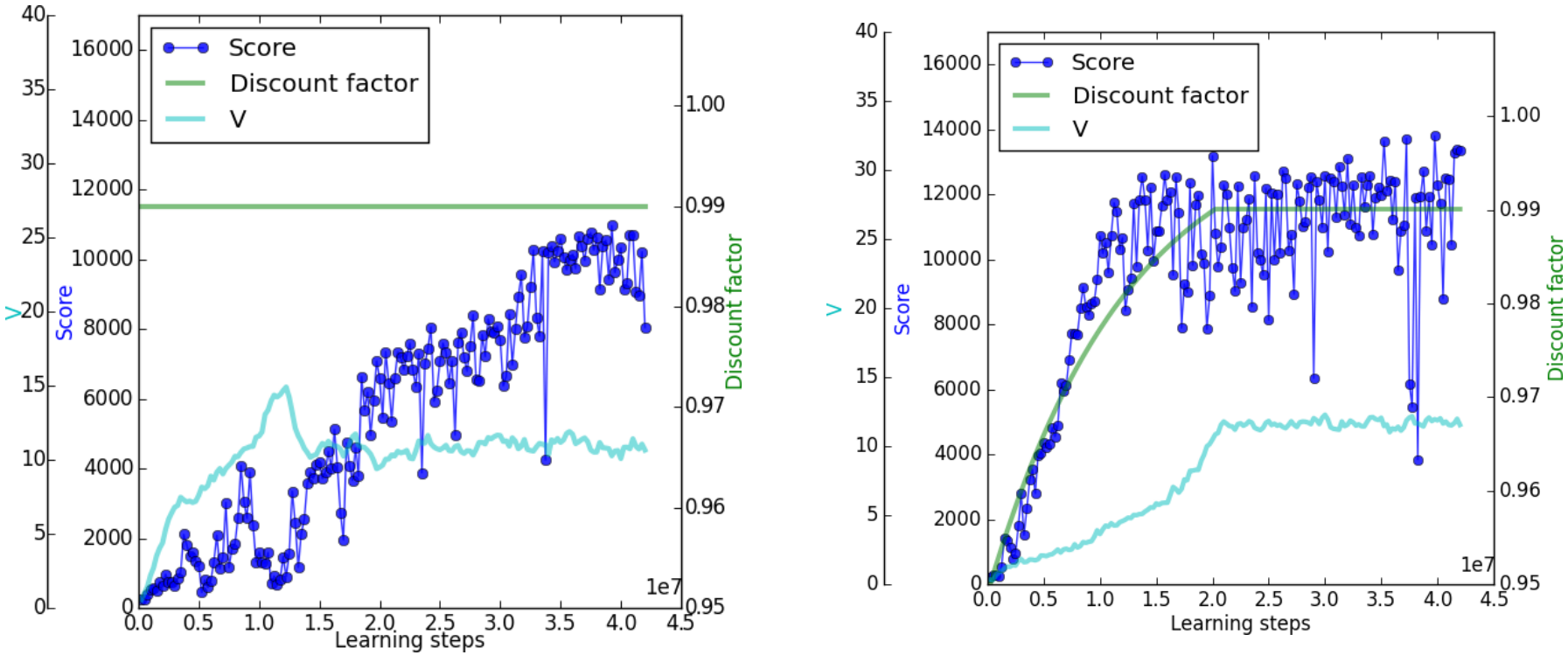


FIGURE: Illustration for the game q-bert of a discount factor γ held fixed on the right and an adaptive discount factor on the right.

Results

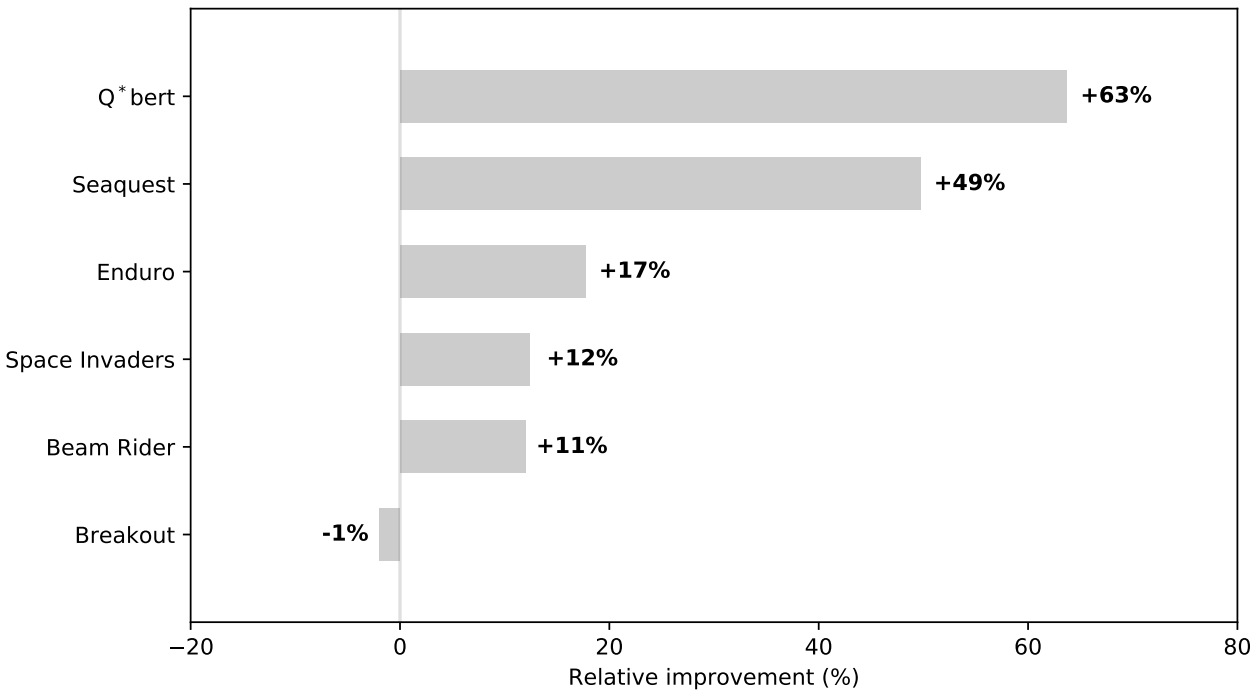


FIGURE: Summary of the results for an increasing discount factor. Reported scores are the relative improvements after 20M steps between an increasing discount factor and a constant discount factor set to its final value $\gamma = 0.99$.

Decreasing learning rate

Possibility to use a more aggressive learning rate in the neural network (when low γ). The learning rate is then reduced to improve stability of the neural Q-learning function.

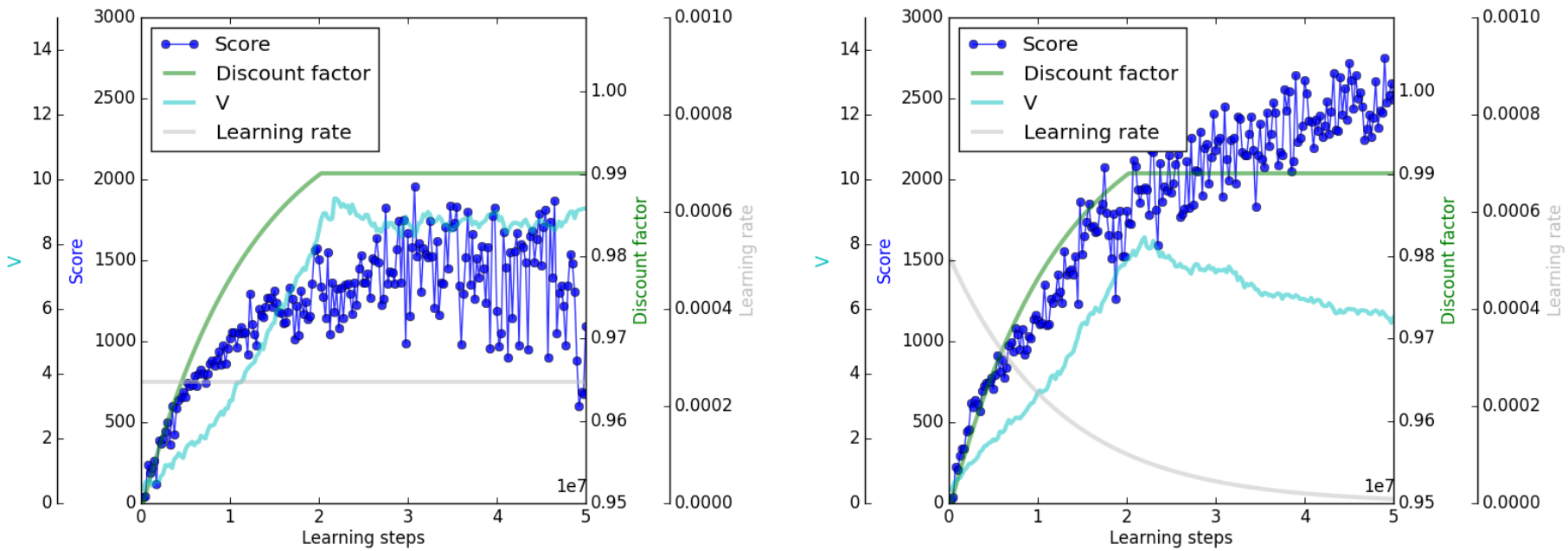


FIGURE: Illustration for the game space invaders. On the left, the deep Q-network with $\alpha = 0.00025$ and on the right with a decreasing learning rate.

Results

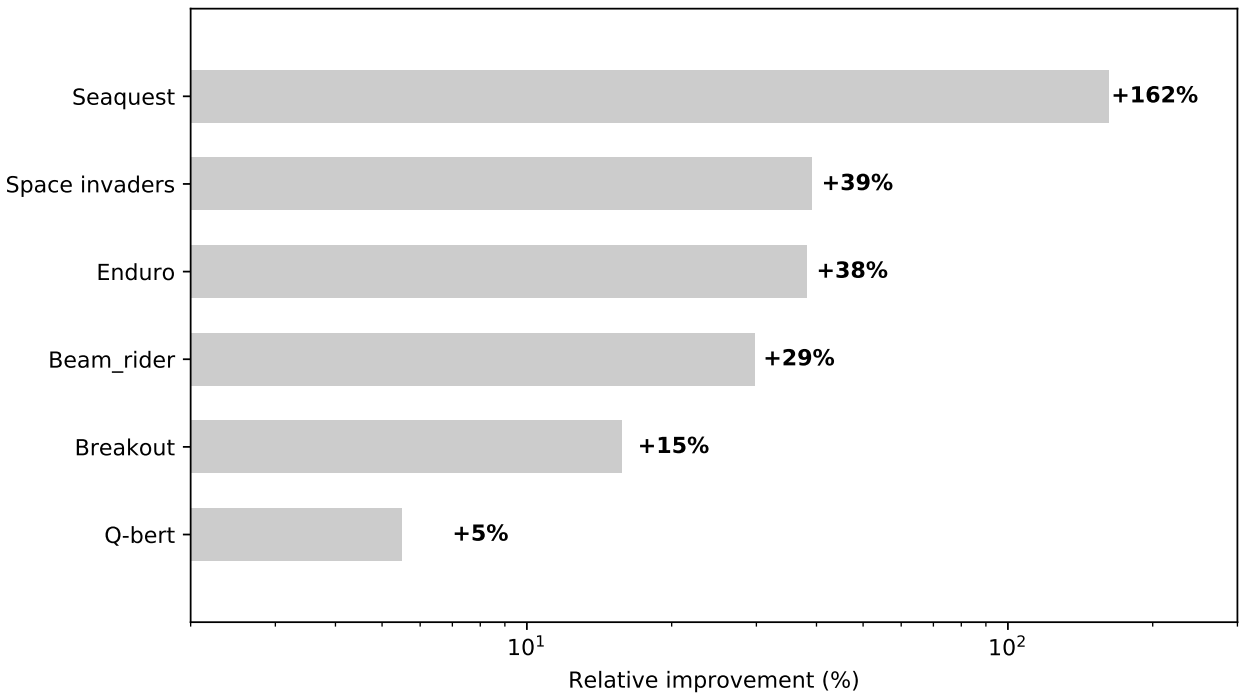


FIGURE: Summary of the results for a decreasing learning rate. Reported scores are the relative improvement after 50M steps between a dynamic discount factor with a dynamic learning rate versus a dynamic discount factor only.

Risk of local optima

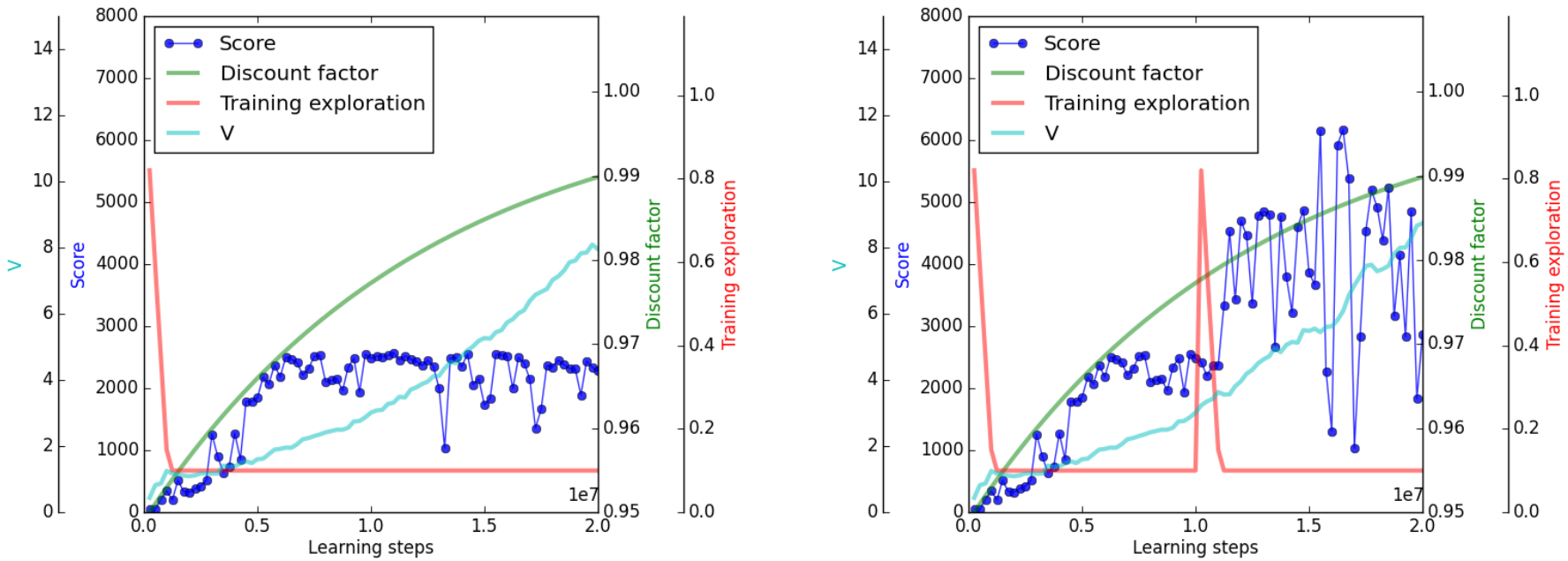


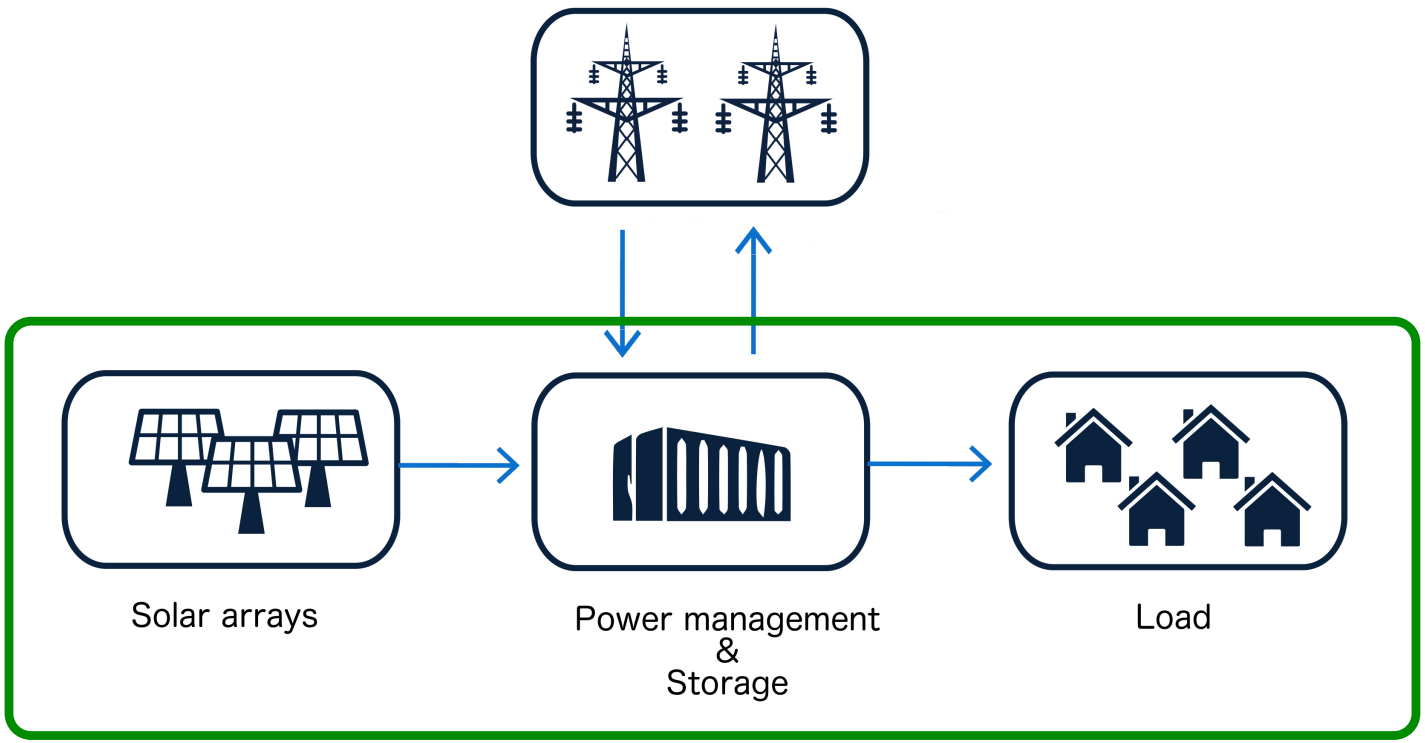
FIGURE: Illustration for the game Seaquest. On the left, the flat exploration rate fails in some cases to get the agent out of a local optimum. On the right, illustration that a simple rule that increases exploration may allow the agent to get out of the local optimum.

Application to smartgrids

The microgrid benchmark

Microgrid

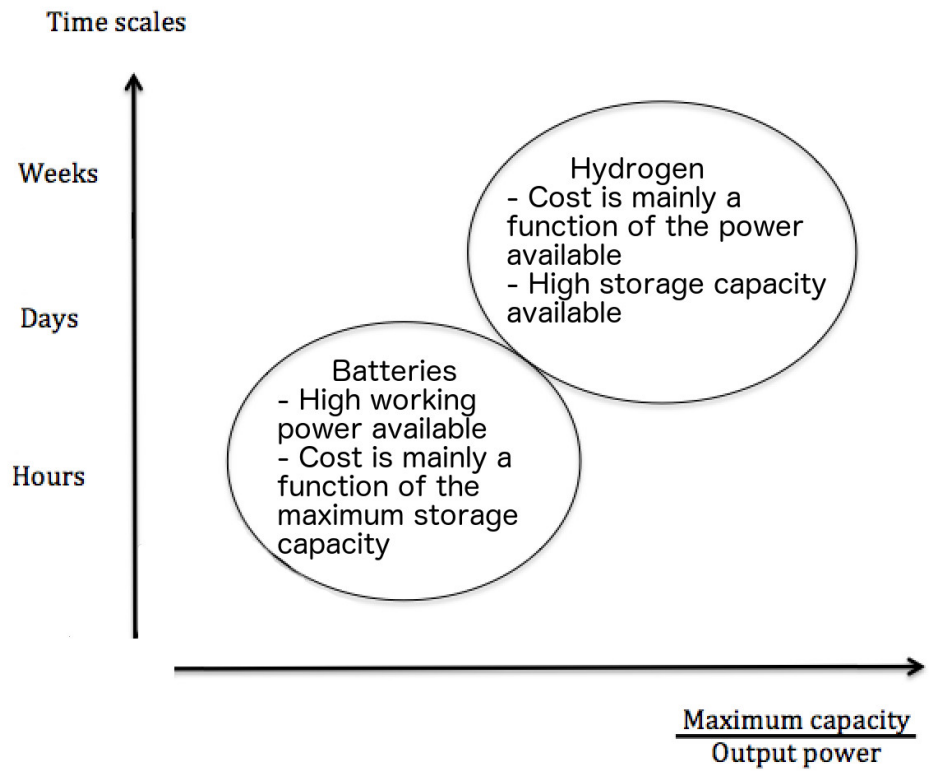
A microgrid is an electrical system that includes multiple loads and distributed energy resources that can be operated in parallel with the broader utility grid or as an electrical island.



Microgrids and storage

There exist opportunities with microgrids featuring :

- A short term storage capacity (typically batteries),
- A long term storage capacity (e.g., hydrogen).



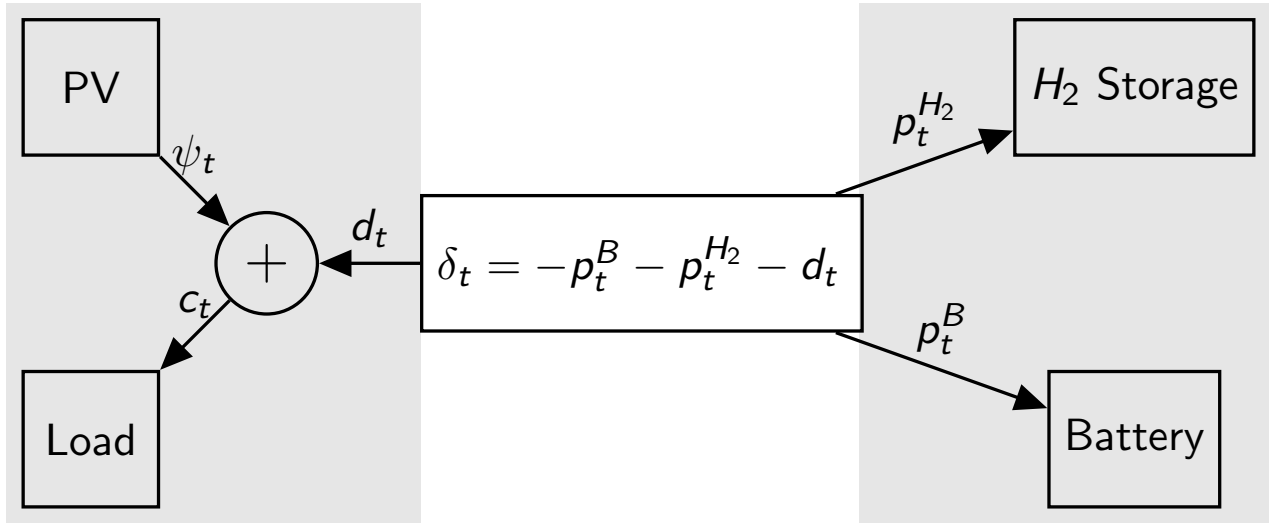


FIGURE: Schema of the microgrid featuring PV panels associated with a battery and a hydrogen storage device.

Formalisation and problem statement : exogenous variables

$$\mathbf{E}_t = (c_t, i_t, \boldsymbol{\mu}_t, \mathbf{e}^{PV}_{1,t}, ..., \mathbf{e}^{PV}_{K,t}, \mathbf{e}^{B}_{1,t}, ..., \mathbf{e}^{B}_{L,t}, \mathbf{e}^{H_2}_{1,t}, ..., \mathbf{e}^{H_2}_{M,t}) \in \mathcal{E}, \ \forall t \in \mathcal{T}$$

$$\text{and with } \mathcal{E} = \mathbb{R}^{+2} \times \mathcal{I} \times \prod_{k=1}^{K} \mathcal{E}^{PV}_k \times \prod_{l=1}^{L} \mathcal{E}^{B}_l \times \prod_{m=1}^{M} \mathcal{E}^{H_2}_m \ ,$$

where :

- c_t $[W] \in \mathbb{R}^+$ is the electricity demand within the microgrid ;
- i_t $[W/m$ or $W/W_p] \in \mathbb{R}^+$ denotes the solar irradiance incident to the PV panels ;
- $\boldsymbol{\mu}_t \in \mathcal{I}$ represents the model of interaction (buying price $k[€/kWh]$, selling price β $[€/kWh]$) ;
- $\mathbf{e}^{PV}_{k,t} \in \mathcal{E}^{PV}_k$, $\forall k \in \{1, ..., K\}$, models a photovoltaic technology ;
- $\mathbf{e}^{B}_{l,t} \in \mathcal{E}^{B}_l$, $\forall l \in \{1, ..., L\}$, represents a battery technology ;
- $\mathbf{e}^{H_2}_{m,t} \in \mathcal{E}^{H_2}_m$, $\forall m \in \{1, ..., M\}$, denotes a hydrogen storage technology ;

Working hypotheses

As a first step, we make the hypothesis that the future consumption and production are known so as to obtain lower bounds on the operational cost.

Main hypotheses on the model :

- we consider one type of battery with a cost proportional to its capacity and one type of hydrogen storage with a cost proportional to the maximum power flows,
- for the storage devices, we considered a charging and a discharging dynamics with constant efficiencies (independent of the power) and without aging (only limited lifetime), and
- for the PV panels, we considered a production to be proportional to the solar irradiance, also without any aging effect on the dynamics (only limited lifetime).

Linear programming

The overall optimization of the operation can be written as :

$$\mathcal{M}_{\text{op}}\left(T,\Delta t,n,\tau_1,\ldots,\tau_n,r,\mathbf{E}_1,\ldots,\mathbf{E}_T,\mathbf{s}^{(s)}\right) = \min \frac{\sum_{y=1}^{n} \frac{M_y}{(1+\rho)^y}}{\sum_{y=1}^{n} \frac{\sum_{t\in\tau_y} c_t \Delta t}{(1+\rho)^y}} \tag{15a}$$

$$\text{s.t. } \forall y \in \{1,\ldots,n\} : \tag{15b}$$

$$M_y = \sum_{t\in\tau_y} \left(k\,\delta_t^- - \beta\,\delta_t^+\right)\Delta t\,, \tag{15c}$$

$$\forall t \in \{1,\ldots,T\} : \tag{15d}$$

$$0 \le s_t^B \le x^B\,, \tag{15e}$$

$$0 \le s_t^{H_2} \le R^{H_2}\,, \tag{15f}$$

$$-P^B \le p_t^B \le P^B\,, \tag{15g}$$

$$-x^{H_2} \le p_t^{H_2} \le x^{H_2}\,, \tag{15h}$$

$$\delta_t = -p_t^B - p_t^{H_2} - c_t + \eta^{PV} x^{PV} i_t\,, \tag{15i}$$

$$p_t^B = p_t^{B,+} - p_t^{B,-}\,, \tag{15j}$$

$$p_t^{H_2} = p_t^{H_2,+} - p_t^{H_2,-}\,, \tag{15k}$$

$$\delta_t = \delta_t^+ - \delta_t^-\,, \tag{15l}$$

$$p_t^{B,+}, p_t^{B,-}, p_t^{H_2,+}, p_t^{H_2,-}, \delta_t^+, \delta_t^- \ge 0\,, \tag{15m}$$

$$s_1^B = 0, s_1^{H_2} = 0, \tag{15n}$$

$$\forall t \in \{2,\ldots,T\} : \tag{15o}$$

$$s_t^B = r^B s_{t-1}^B + \eta^B p_{t-1}^{B,+} - \frac{p_{t-1}^{B,-}}{\zeta^B}\,, \tag{15p}$$

$$s_t^{H_2} = r^{H_2} s_{t-1}^{H_2} + \eta^{H_2} p_{t-1}^{H_2,+} - \frac{p_{t-1}^{H_2,-}}{\zeta^{H_2}}\,, \tag{15q}$$

$$-\zeta^B s_T^B \le p_T^B \le \frac{x^B - s_T^B}{\eta^B}\,, \tag{15r}$$

$$-\zeta^{H_2} s_T^{H_2} \le p_T^{H_2} \le \frac{R^{H_2} - s_T^{H_2}}{\eta^{H_2}}\,. \tag{15s}$$

The overall optimization of the operation and sizing :

$$\mathcal{M}_{\text{size}}(T,\Delta t,n,\tau_1,\ldots,\tau_n,r,\mathbf{E}_0,\mathbf{E}_1,\ldots,\mathbf{E}_T) = \min \frac{I_0 + \sum_{y=1}^{n} \frac{M_y}{(1+\rho)^y}}{\sum_{y=1}^{n} \frac{\sum_{t\in\tau_y} c_t \Delta t}{(1+\rho)^y}} \tag{16a}$$

$$\text{s.t. } I_0 = a_0^{PV} c_0^{PV} + a_0^{B} c_0^{B} + a_0^{H_2} c_0^{H_2}\,, \tag{16b}$$

$$(x^{B}, x^{H_2}, x^{PV}) = (a_0^{B}, a_0^{H_2}, a_0^{PV})\,, \tag{16c}$$

$$15b - 15s\,. \tag{16d}$$

A robust optimization model that integrates a set $\mathbf{E} = \{(E_t^1)_{t=1\ldots T}, \ldots, (E_t^N)_{t=1\ldots T}\}$ of candidate trajectories of the environment vectors can be obtained with two additional levels of optimization :

$$\mathcal{M}_{\text{rob}}(T,\Delta t,n,\tau_1,\ldots,\tau_n,r,\mathbf{E}_0,\mathbf{E}) = \tag{17a}$$

$$\min_{a_0^{B},a_0^{H_2},a_0^{PV}} \max_{i\in 1,\ldots,N} \mathcal{M}_{\text{size}}(T,\Delta t,n,\tau_1,\ldots,\tau_n,r,\mathbf{E}_0,\mathbf{E}_1^{(i)},\ldots,\mathbf{E}_T^{(i)})\,. \tag{17b}$$

Results : characteristics of the different components

Parameter	Value
c^{PV}	1€/W$_p$
η^{PV}	18%
L^{PV}	20 years

TABLE: Characteristics used for the PV panels.

Parameter	Value
c^{B}	500 €/kWh
η_0^{B}	90%
ζ_0^{B}	90%
P^{B}	>10kW
r^{B}	99%/month
L^{B}	20 years

TABLE: Data used for the $LiFePO_4$battery.

Parameter	Value
c^{H_2}	14 €/W$_p$
$\eta_0^{H_2}$	65%
$\zeta_0^{H_2}$	65%
r^{H_2}	99%/month
L^{H_2}	20 years
R^{H_2}	∞

TABLE: Data used for the Hydrogen storage device.

Results : consumption and production profiles

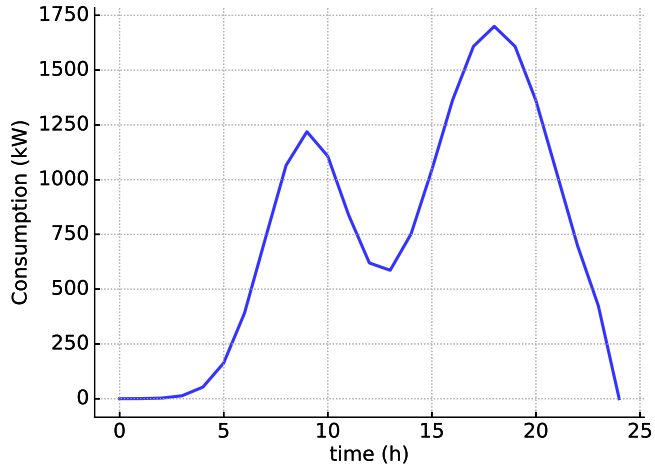
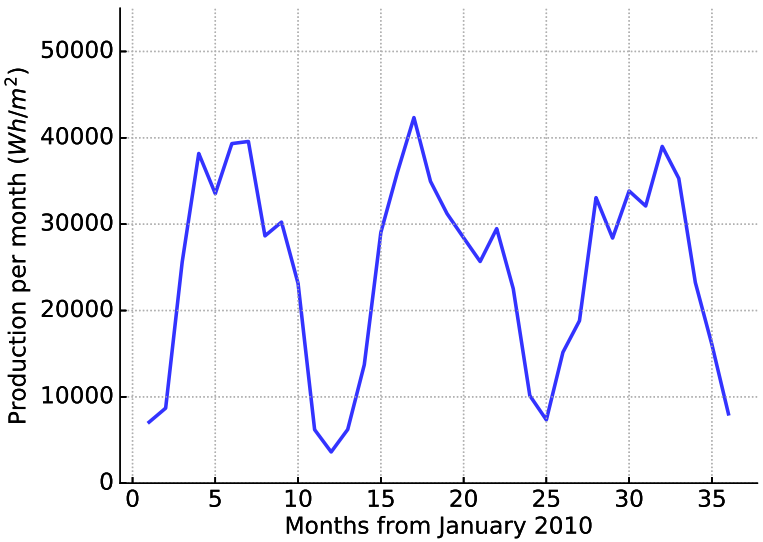


FIGURE: Representative residential consumption profile.



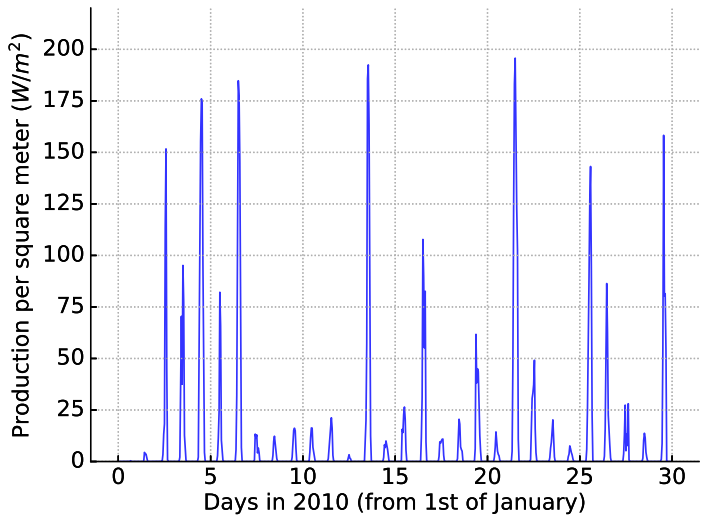


FIGURE: Production for the PV panels in Belgium

Results - Belgium

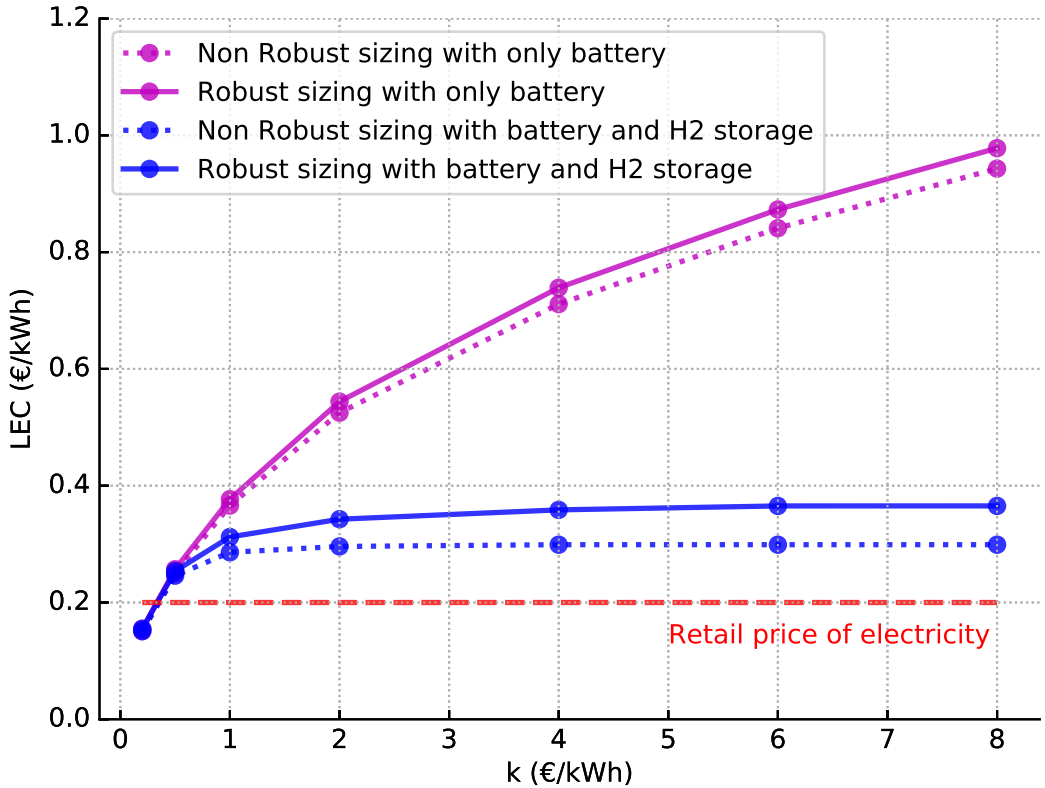


FIGURE: LEC ($\rho = 2\%$) in Belgium over 20 years for different investment strategies as a function of the cost endured per kWh not supplied within the microgrid(NB :$\beta = 0$ €/kWh).

Results - Spain

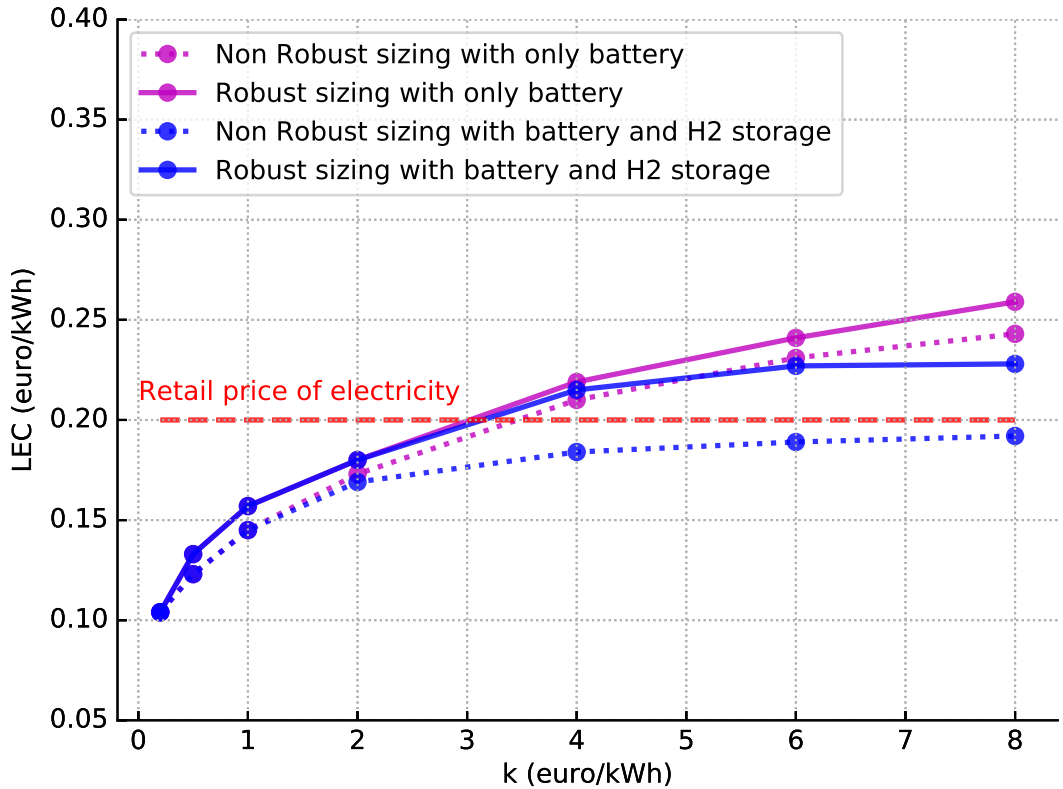


FIGURE: LEC ($\rho = 2\%$) in Spain over 20 years for different investment strategies as a function of the cost endured per kWh not supplied within the microgrid ($\beta = 0$ €/kWh).

Deep RL solution

We remove the hypothesis that the future consumption and production are known (realistic setting).

The goal of this is two-fold :

- obtaining an operation that can actually be used in practice.
- determine the additional costs as compared to the lower bounds (when known future production and consumption)

Setting considered

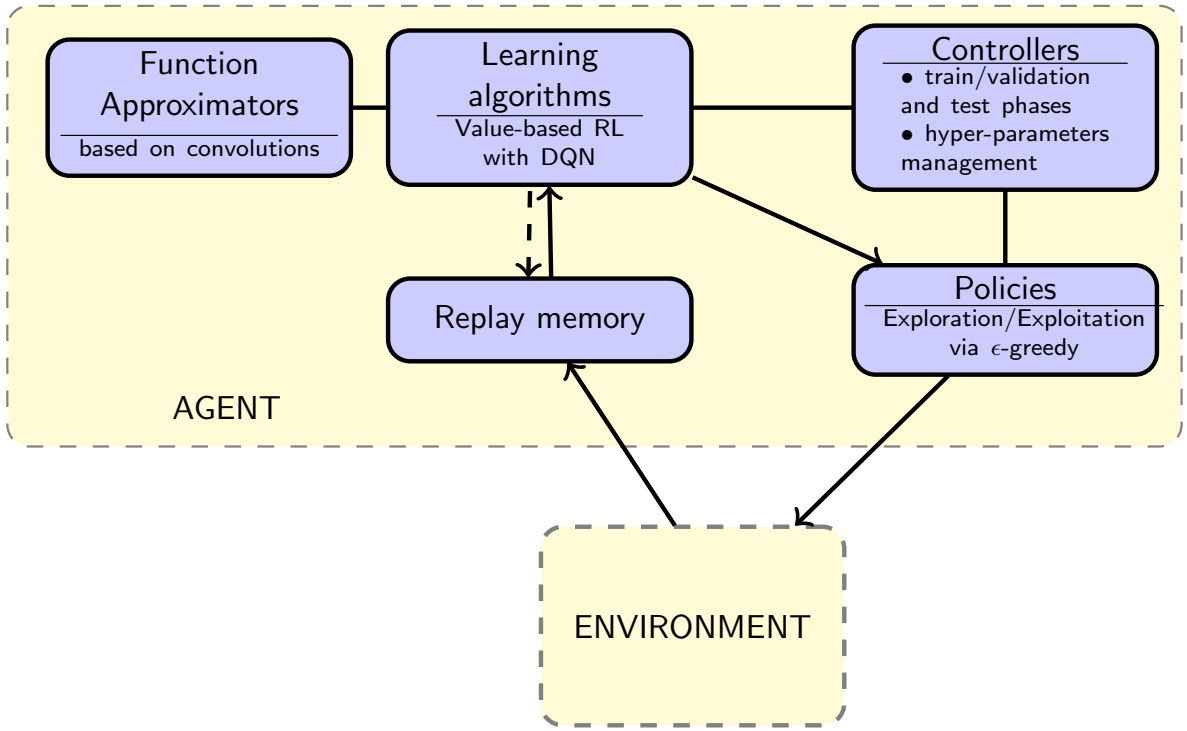
The only required assumptions, rather realistic, are that

- the dynamics of the different constituting elements of the microgrid are known and
- past time series providing the weather dependent PV production and consumption within the microgrid are available (one year of data is used for training, one year for validation, and one year is used for the test environment).

The setting considered is as follows :

- Fixed sizing of the microgrid (corresponding to the robust sizing : $x^{PV} = 12kW_p$, $x^{B} = 15kWh$, $x^{H_2} = 1.1kW$)
- Cost k incurred per kWh not supplied within the microgrid set to 2 €/kWh (corresponding to a value of loss load).
- Revenue (resp. cost) per Wh of hydrogen produced (resp. used) set to 0.1 €/kWh.

Overview



Related to the methodological/theoretical contributions :

- validation phase to obtain the best bias-overfitting tradeoff (and to select the Q-network when instabilities are not too harmful),
- increasing discount factor to improve training time and stability

Implementation : https ://github.com/VinF/deer

Structure of the Q-network

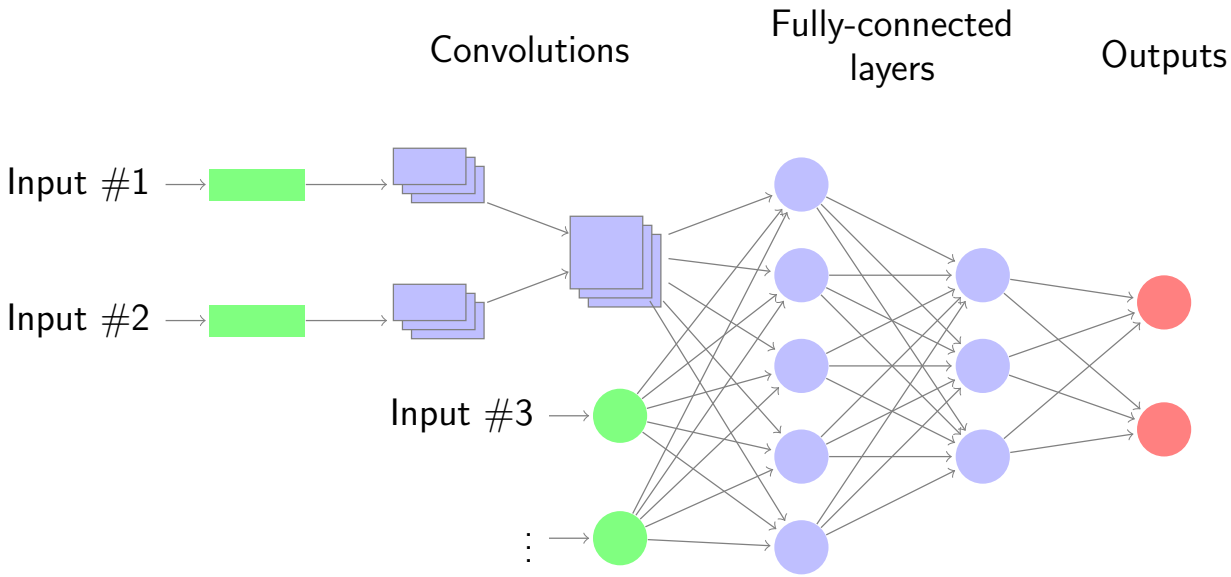
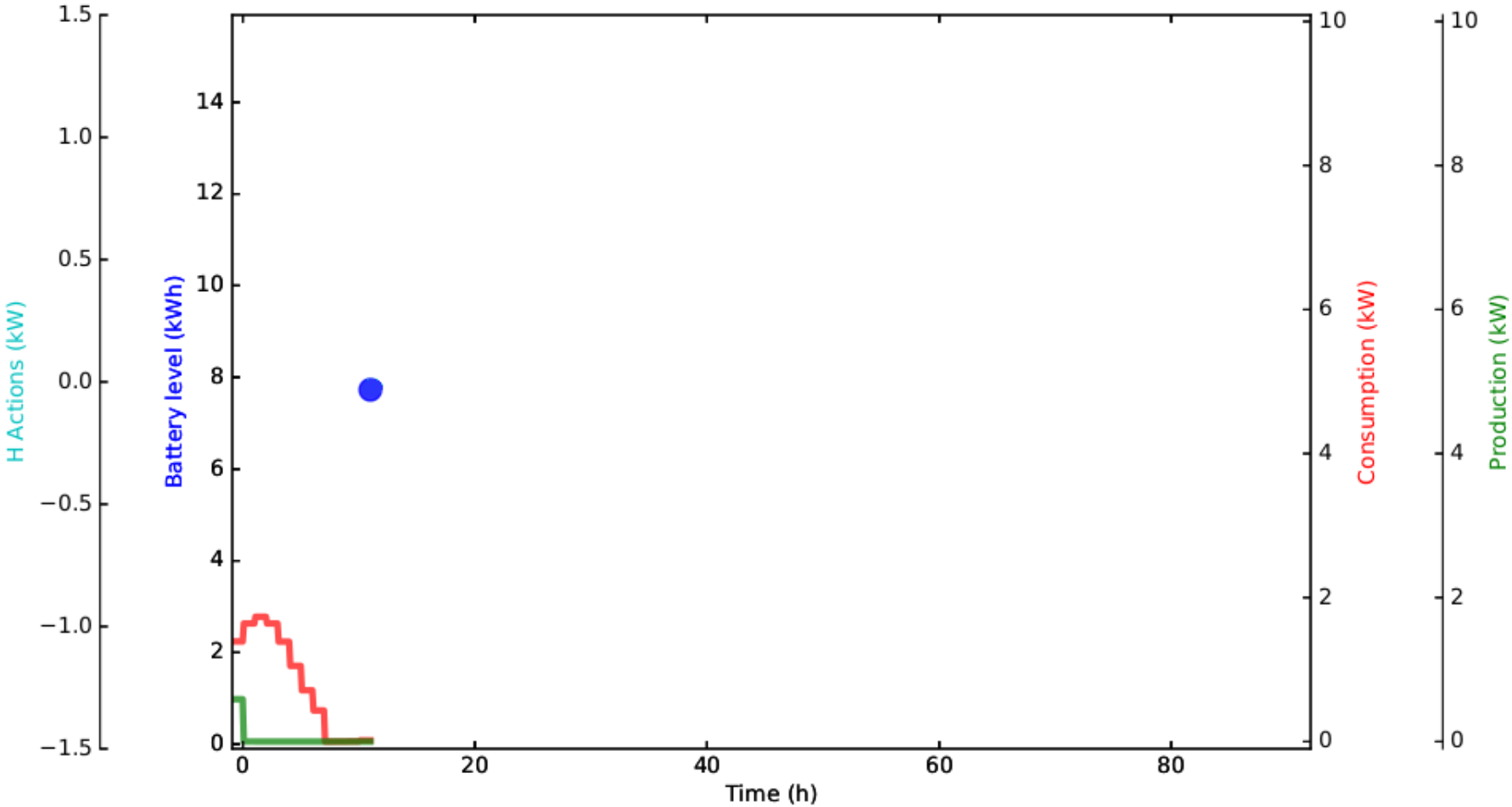


FIGURE: Sketch of the structure of the neural network architecture. The neural network processes the time series using a set of convolutional layers. The output of the convolutions and the other inputs are followed by fully-connected layers and the ouput layer. Architectures based on LSTMs instead of convolutions obtain similar results.

Example

Illustration of the policy on the test data with the following features :

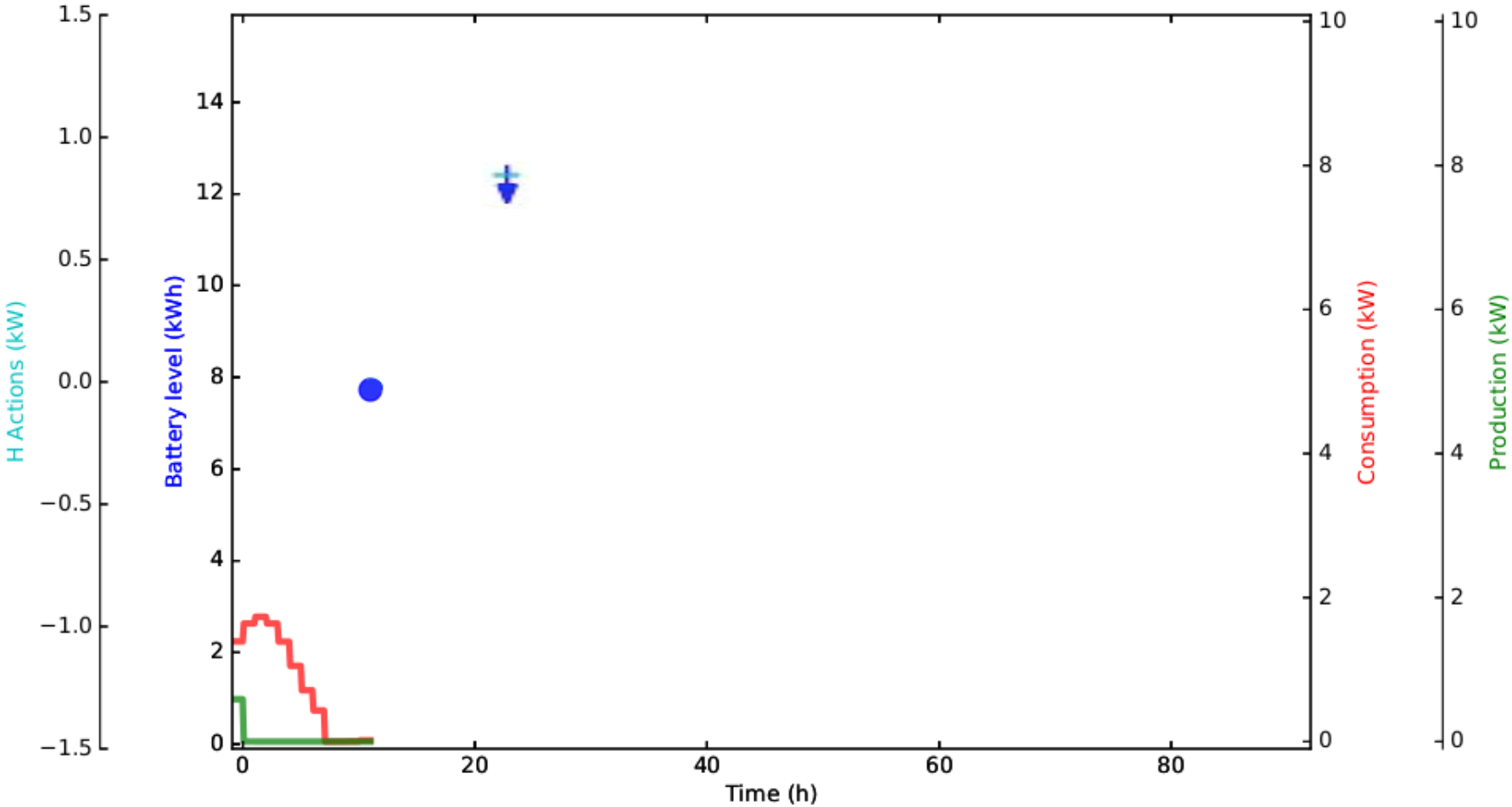
- past 12 hours for the production and consumption, and
- charge level of the battery.



Example

Illustration of the policy on the test data with the following features :

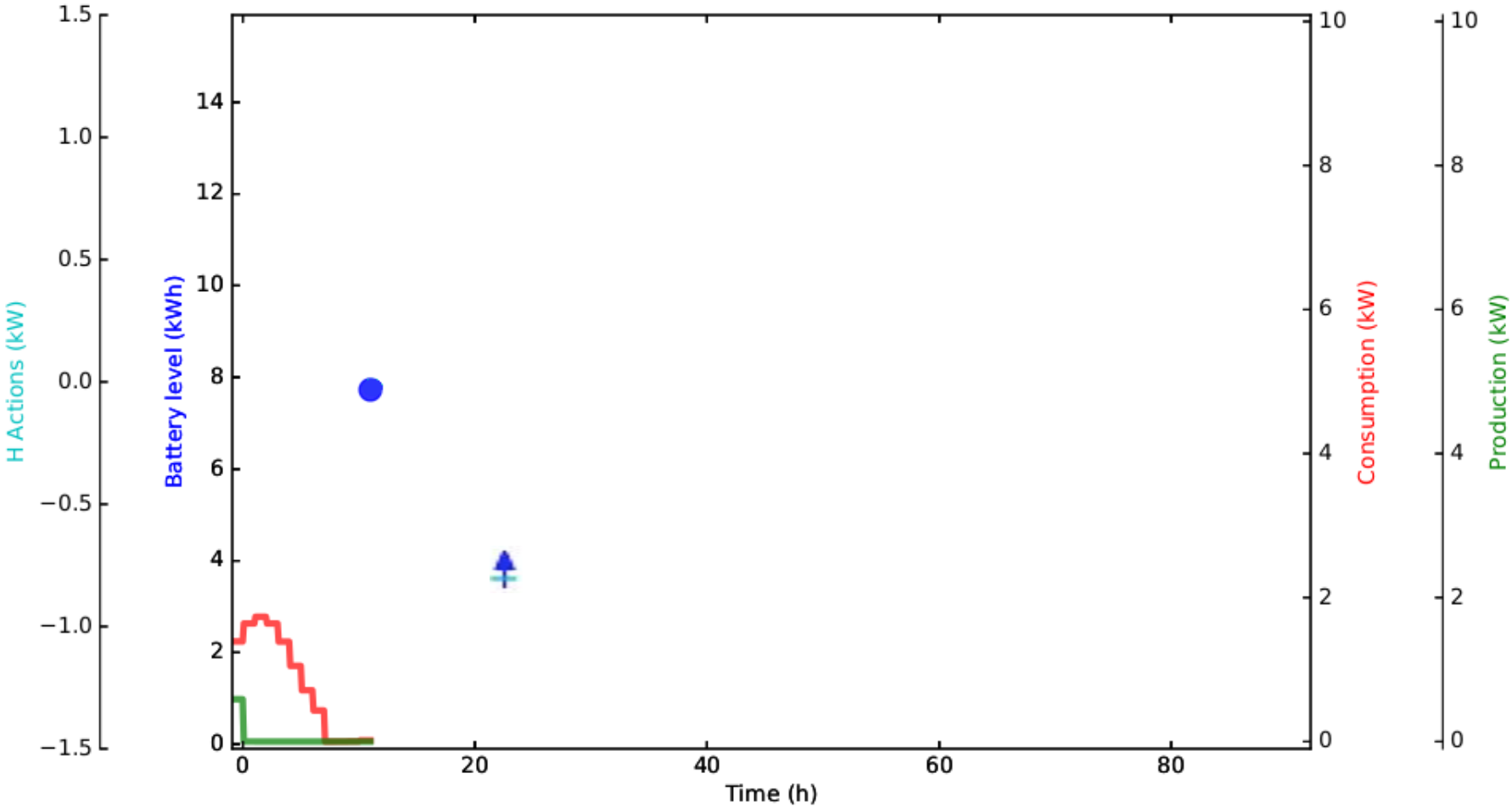
- past 12 hours for the production and consumption, and
- charge level of the battery.



Example

Illustration of the policy on the test data with the following features :

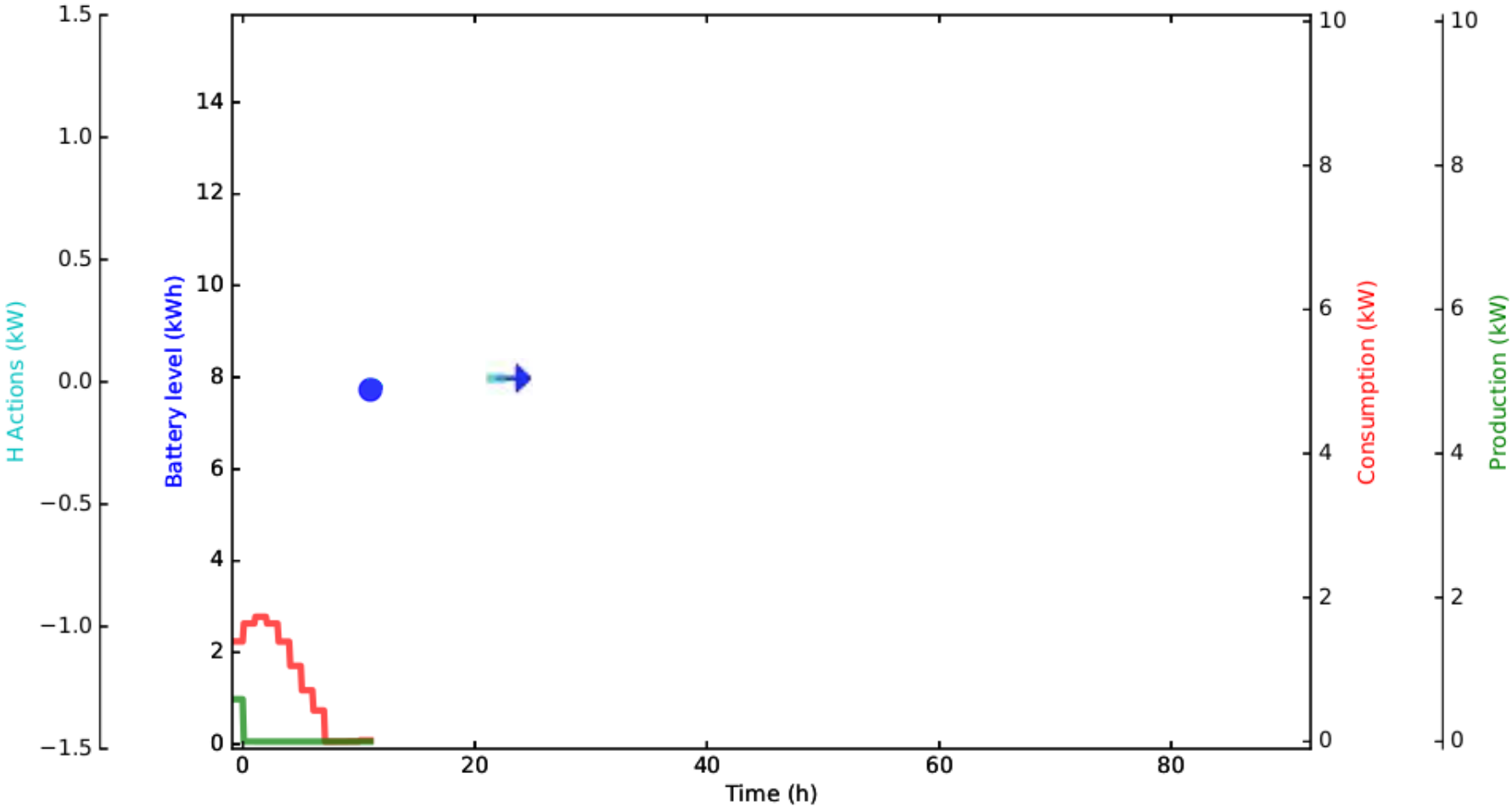
- past 12 hours for the production and consumption, and
- charge level of the battery.



Example

Illustration of the policy on the test data with the following features :

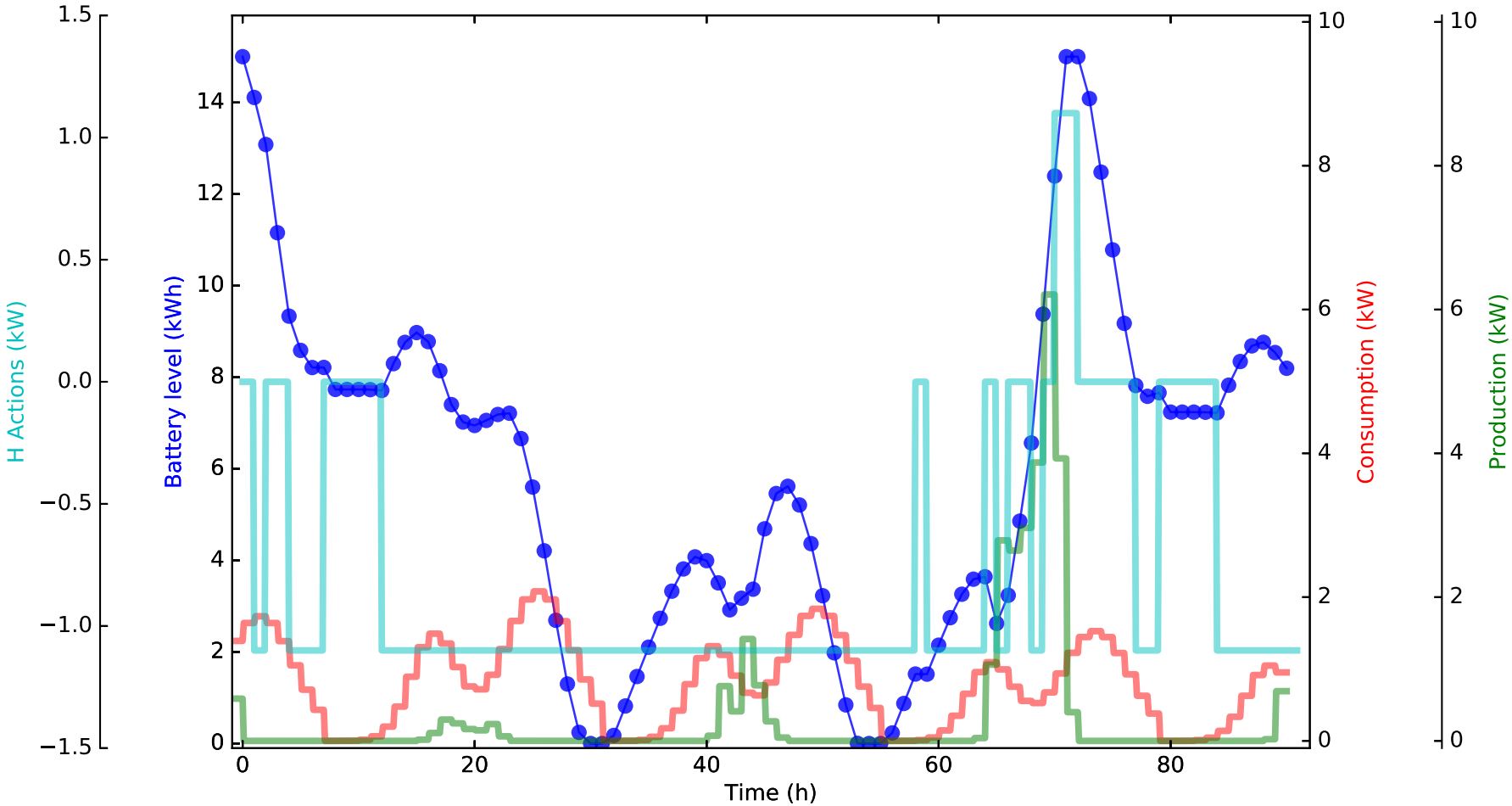
- past 12 hours for the production and consumption, and
- charge level of the battery.



Example

Illustration of the policy on the test data with the following features :

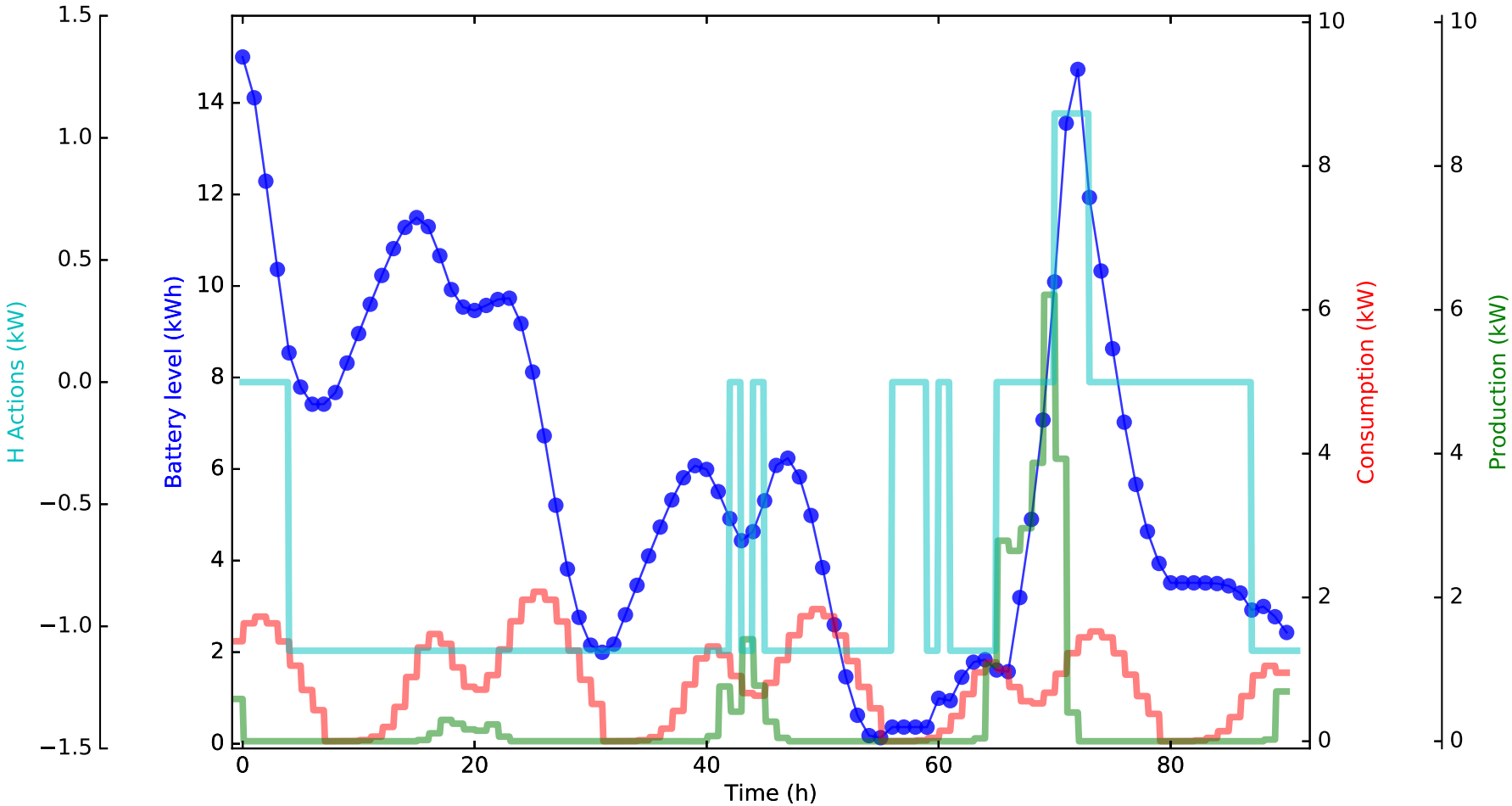
- past 12 hours for the production and consumption, and
- charge level of the battery.



Example

Illustration of the policy on the test data with the following features :

- past 12 hours for the production and consumption,
- level of the battery, and
- Accurate forecast for the mean production of the next 24h and 48h.



Results

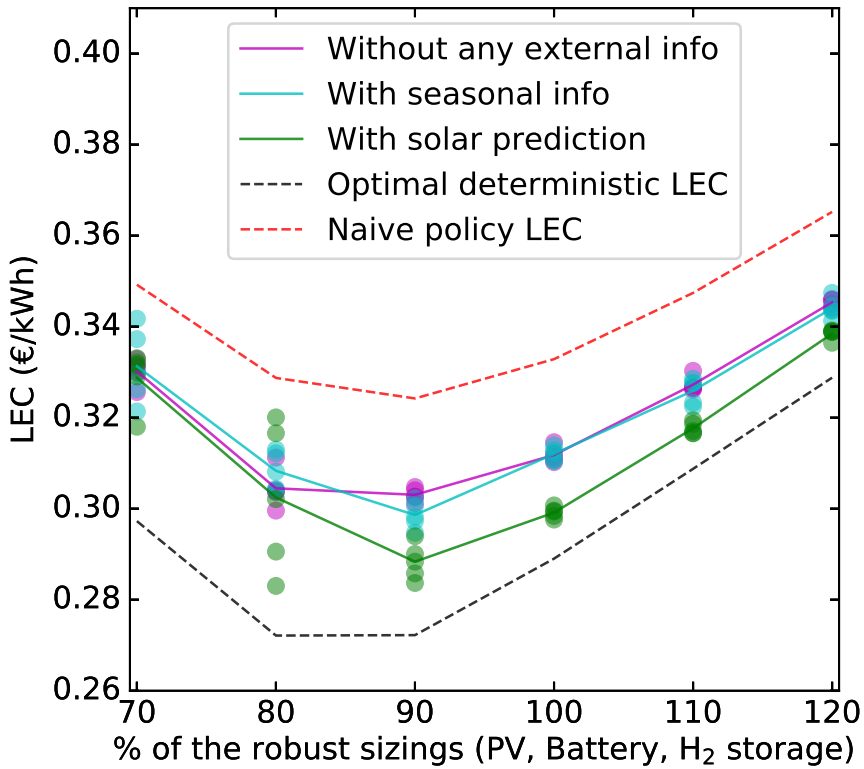


FIGURE: LEC on the test data function of the sizings of the microgrid.

Conclusions

Contributions of the thesis

- Review of the domain of reinforcement learning with a focus on deep RL.
- Theoretical and experimental contributions to the partially observable context (POMDP setting) where only limited data is available (batch RL).
- Case of the discount factor in a value iteration algorithm with deep learning (online RL).
- Linear optimization techniques to solve both the optimal operation and the optimal sizing of a microgrid with PV, long-term and short-term storage (deterministic hypothesis).
- Deep RL techniques can be used to obtain a performant real-time operation (realistic setting).
- Open source library DeeR (`http://deer.readthedocs.io`)

Thank you