Deep neural networks algorithms for stochastic control problems on finite horizon, part I: convergence analysis

Côme HURÉ [*]   Huyên PHAM [†]   Achref BACHOUCH [‡]   Nicolas LANGRENÉ [§]

December 12, 2018

Abstract

This paper develops algorithms for high-dimensional stochastic control problems based on deep learning and dynamic programming (DP). Differently from the classical approximate DP approach, we first approximate the optimal policy by means of neural networks in the spirit of deep reinforcement learning, and then the value function by Monte Carlo regression. This is achieved in the DP recursion by performance or hybrid iteration, and regress now or later/quantization methods from numerical probabilities. We provide a theoretical justification of these algorithms. Consistency and rate of convergence for the control and value function estimates are analyzed and expressed in terms of the universal approximation error of the neural networks. Numerical results on various applications are presented in a companion paper [2] and illustrate the performance of our algorithms.

Keywords: Deep learning, dynamic programming, performance iteration, quantization, convergence analysis.

[*]LPSM, University Paris Diderot hure at lpsm.paris

[†]LPSM, University Paris-Diderot and CREST-ENSAE, pham at lspm.paris The work of this author is supported by the ANR project CAESARS (ANR-15-CE05-0024), and also by FiME and the "Finance and Sustainable Development" EDF - CACIB Chair

[‡]Department of Mathematics, University of Oslo, Norway. The author's research is carried out with support of the Norwegian Research Council, within the research project Challenges in Stochastic Control, Information and Applications (STOCONINF), project number 250768/F20 achrefb at math.uio.no

[§]CSIRO, Data61, RiskLab Australia Nicolas.Langrene at data61.csiro.au

Contents

1 Introduction

Let us consider the following discrete-time stochastic control problem over a finite horizon $N \in \mathbb{N} \setminus \{0\}$. The dynamics of the controlled state process $X^\alpha = (X_n^\alpha)_n$ valued in $\mathcal{X} \subset \mathbb{R}^d$ is given by

$$X_{n+1}^\alpha \;=\; F(X_n^\alpha, \alpha_n, \varepsilon_{n+1}), \quad n = 0, \ldots, N-1, \; X_0^\alpha = x_0 \in \mathbb{R}^d, \qquad (1.1)$$

where $(\varepsilon_n)_n$ is a sequence of i.i.d. random variables valued in some Borel space $(E, \mathcal{B}(E))$, and defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$ equipped with the filtration $\mathbb{F} = (\mathcal{F}_n)_n$ generated by the noise $(\varepsilon_n)_n$ ($\mathcal{F}_0$ is the trivial σ-algebra), the control $\alpha = (\alpha_n)_n$ is an

$\mathbb{F}$-adapted process valued in $\mathbb{A} \subset \mathbb{R}^q$, and F is a measurable function from $\mathbb{R}^d \times \mathbb{R}^q \times E$ into $\mathbb{R}^d$.

Given a running cost function f defined on $\mathbb{R}^d \times \mathbb{R}^q$, a terminal cost function g defined on $\mathbb{R}^d$, the cost functional associated to a control process α is

$$J(\alpha) \;\; = \;\; \mathbb{E}\left[\sum_{n=0}^{N-1} f(X_n^\alpha, \alpha_n) + g(X_N^\alpha)\right].$$

The set $\mathcal{C}$ of admissible control is the set of control processes α satisfying some integrability conditions ensuring that the cost functional $J(\alpha)$ is well-defined and finite. The control problem, also called Markov decision process (MDP), is formulated as

$$V_0(x_0) \;\; := \;\; \inf_{\alpha \in \mathcal{C}} J(\alpha), \tag{1.2}$$

and the goal is to find an optimal control $\alpha^* \in \mathcal{C}$, i.e., attaining the optimal value: $V_0(x_0) = J(\alpha^*)$. Notice that problem (1.1)-(1.2) may also be viewed as the time discretization of a continuous time stochastic control problem, in which case, F is typically the Euler scheme for a controlled diffusion process, and V_0 is the discrete-time approximation of a fully nonlinear Hamilton-Jacobi-Bellman equation.

Problem (1.2) is tackled by the dynamic programming approach, and we introduce the standard notations for MDP: denote by $\{P^a(x, dx'), a \in \mathbb{A}, x \in \mathcal{X}\}$, the family of transition probabilities associated to the controlled (homogenous) Markov chain (1.1), given by

$$P^a(x, dx') \;\; = \;\; \mathbb{P}\left[F(x, a, \varepsilon_1) \in dx'\right]$$

and for any measurable function φ on $\mathcal{X}$:

$$P^a \varphi(x) = \int \varphi(x') P^a(x, dx') \;\; = \;\; \mathbb{E}\left[\varphi\big(F(x, a, \varepsilon_1)\big)\right].$$

With these notations, we have for any measurable function φ on $\mathcal{X}$, for any $\alpha \in \mathcal{C}$,

$$\mathbb{E}[\varphi(X_{n+1}^\alpha)|\mathcal{F}_n] \;\; = \;\; P^{\alpha_n} \varphi(X_n^\alpha), \quad \forall\, n \in \mathbb{N}.$$

The optimal value $V_0(x_0)$ is then determined in backward induction starting from the terminal condition

$$V_N(x) \;\; = \;\; g(x), \quad x \in \mathcal{X},$$

and by the dynamic programming (DP) formula, for $n = N-1, \ldots, 0$:

$$\begin{cases} Q_n(x, a) & = & f(x, a) + P^a V_{n+1}(x), \quad x \in \mathcal{X}, \ a \in \mathbb{A}, \\ V_n(x) & = & \inf_{a \in \mathbb{A}} Q_n(x, a), \end{cases} \tag{1.3}$$

The function Q_n is called optimal state-action value function, and V_n is the (optimal) value function. Moreover, when the infimum is attained in the DP formula at any time n by $a_n^*(x)$, we get an optimal control in feedback form given by: $\alpha^* = (a_n^*(X_n^*))_n$ where $X^* = X^{\alpha^*}$ is the Markov process defined by

$$X_{n+1}^* \;\; = \;\; F(X_n^*, a_n^*(X_n^*), \varepsilon_{n+1}), \quad n = 0, \ldots, N-1, \ \ X_0^* = x_0.$$

The DP has a probabilistic formulation: it says that for any control $\alpha \in \mathcal{A}$, the value function process augmented with the cumulative costs defined by

$$\left\{ S_n^\alpha \ := \ V_n(X_n^\alpha) + \sum_{k=0}^{n-1} f(X_k^\alpha, \alpha_k), \ n = 1, \ldots, N \right\} \tag{1.4}$$

is a submartingale, and a martingale for the optimal control α^*. This martingale property for the optimal control is a key observation for our algorithms described later.

Remark 1.1 We can deal with state/control constraints at any time, which is useful for the applications:

$$(X_n^\alpha, \alpha_n) \ \in \ \mathcal{S} \ \ a.s., \ \ n \in \mathbb{N},$$

where $\mathcal{S}$ is some given subset of $\mathbb{R}^d \times \mathbb{R}^q$. In this case, in order to ensure that the set of admissible controls is not empty, we assume that the sets

$$\mathbb{A}(x) \ := \ \left\{ a \in \mathbb{R}^q : (F(x, a, \varepsilon_1), a) \in \mathcal{S} \ a.s. \right\}$$

are non empty for all $x \in \mathcal{X}$, and the DP formula reads now as

$$V_n(x) \ = \ \inf_{a \in \mathbb{A}(x)} \left[f(x, a) + P^a V_{n+1}(x) \right], \quad x \in \mathcal{X}.$$

From a computational point of view, it may be more convenient to work with unconstrained state/control variable, hence by relaxing the state/control constraint and introducing into the running cost a penalty function $L(x, a)$: $f(x, a) \leftarrow f(x, a) + L(x, a)$, and $g(x) \leftarrow g(x) + L(x, a)$. For example, if the constraint set $\mathcal{S}$ is in the form: $\mathcal{S} = \{(x, a) \in \mathbb{R}^d \times \mathbb{R}^q : h_k(x, a) = 0, k = 1, \ldots, p, \ h_k(x, a) \geq 0, k = p + 1, \ldots, q\}$, for some functions h_k, then one can take as penalty functions:

$$L(x, a) \ = \ \sum_{k=1}^{p} \mu_k |h_k(x, a)|^2 + \sum_{k=p+1}^{q} \mu_k \max(0, -h_k(x, a)).$$

where $\mu_k > 0$ are penalization coefficients (large in practice). $\qquad \square$

The implementation of the DP formula requires the knowledge and explicit computation of the transition probabilities $P^a(x, dx')$. In situations when they are unknown, this leads to the problematic of reinforcement learning for computing the optimal control and value function by relying on simulations of the environment. The challenging tasks from a numerical point of view are then twofold:

1. *Transition probability operator.* Calculations for any $x \in \mathcal{X}$, $a \in \mathbb{A}$ of $P^a V_{n+1}(x)$, for $n = 0, \ldots, N - 1$. This is a computational challenge in high dimension d for the state space with the "curse of dimensionality" due to the explosion of grid points in deterministic methods.

2. *Optimal control.* Computation of the infimum in $a \in \mathbb{A}$ of $\left[f(x, a) + \rho P^a V_{n+1}(x) \right]$ for fixed x and n, and of $\hat{a}_n(x)$ attaining the minimum if it exists. This is also a computational challenge especially in high dimension q for the control space.

4

The classical probabilistic numerical methods based on DP for solving the MDP are sometimes called approximate dynamic programming methods, see e.g. [4], [29], and consist basically of the two following steps:

(i) Approximate at each time step n the Q_n value function defined as a conditional expectation. This can be performed by regression Monte-Carlo (RMC) techniques or quantization. RMC is typically done by least-square linear regression on a set of basis function following the popular approach by Longstaff and Schwarz [24] initiated for Bermudean option problem, where the suitable choice of basis functions might be delicate. Conditional expectation can be also approximated by regression on neural network as in [19] for American option problem, and appears as a promising and efficient alternative in high dimension to the linear regression. The main issue in the controlled case concerns the simulation of the endogenous controlled MDP, and this can be overcome by control randomization as in [17]. Alternatively, quantization method consists in approximating the noise (ε_n) by a discrete random variable on a finite grid, in order to reduce the conditional expectation to a finite sum.

(ii) Control search: Once we get an approximation $(x, a) \mapsto \hat{Q}_n(x, a)$ of the Q_n value function, the optimal control $\hat{a}_n(x)$ which achieves the minimum over $a \in \mathbb{A}$ of $Q_n(x, a)$ can be obtained either by an exhaustive search when $\mathbb{A}$ is discrete (with relatively small cardinality), or by a (deterministic) gradient-based algorithm for continuous control space (with relatively small dimension).

Recently, numerical methods by direct approximation, without DP, have been developed and made implementable thanks to the power of computers: the basic idea is to focus directly on the control approximation by considering feedback control (policy) in a parametric form:

$$a_n(x) \quad = \quad A(x; \theta_n), \quad n = 0, \ldots, N-1,$$

for some given function $A(., \theta_n)$ with parameters $\theta = (\theta_0, \ldots, \theta_{N-1}) \in \mathbb{R}^{q \times N}$, and minimize over θ the parametric functional

$$\tilde{J}(\theta) \quad = \quad \mathbb{E}\left[\sum_{n=0}^{N-1} f(X_n^A, A(x; \theta_n)) + g(X_N^A)\right],$$

where $(X_n^A)_n$ denotes the controlled process with feedback control $(A(., \theta_n))_n$. This approach was first adopted in [21], who used EM algorithm for optimizing over the parameter θ, and further investigated in [13], [6], [15], who considered deep neural networks (DNN) for the parametric feedback control, and stochastic gradient descent methods (SGD) for computing the optimal parameter θ. The theoretical foundation of these DNN algorithms has been recently investigated in [14]. Deep learning has emerged recently in machine learning as a successful technique for dealing with high-dimensional problems in speech recognition, computer vision, etc (see e.g. [22], [9]). Let us mention that DNN approximation in stochastic control has already been explored in the context of reinforcement learning (RL) (see [4] and [30]), and called deep reinforcement learning in the artificial intelligence community

5

[26] (see also [23] for a recent survey) but usually for infinite horizon (stationary) control problems.

In this paper, we combine different ideas from the mathematics (numerical probability) and the computer science (reinforcement learning) communities to propose and compare several algorithms based on dynamic programming (DP), and deep neural networks (DNN) for the approximation/learning of (i) the optimal policy, and then of (ii) the value function. Notice that this differs from the classical approach in DP recalled above, where we first approximate the Q-optimal state/control value function, and then approximate the optimal control. Our learning of the optimal policy is achieved in the spirit of [13] by DNN, but sequentially in time though DP instead of a global learning over the whole period $0, \ldots, N-1$. Once we get an approximation of the optimal policy, and recalling the martingale property (1.4), we approximate the value function by Monte-Carlo (MC) regression based on simulations of the forward process with the approximated optimal control. In particular, we avoid the issue of *a priori* endogenous simulation of the controlled process in the classical Q-approach. The MC regressions for the approximation of the optimal policy and/or value function, are performed according to different features leading to algorithmic variants: Performance iteration (PI) or hybrid iteration (HI), and regress now or regress later/quantization in the spirit of [24] or [8]. Numerical results on several applications are devoted to a companion paper [2]. The theoretical contribution of the current paper is to provide a detailed convergence analysis of our three proposed algorithms: Theorem 4.1 for the *NNContPI* Algo based on control learning by performance iteration with DNN, Theorem 4.2 for the *Hybrid-Now* Algo based on control learning by DNN and then value function learning by regress-now method, and Theorem 4.3 for the *Hybrid-LaterQ* Algo based on on control learning by DNN and then value function learning by regress later method combined with quantization. We rely mainly on arguments from statistical learning and non parametric regression as developed notably in the book [12], for giving estimates of approximated control and value function in terms of the universal approximation error of the neural networks.

The plan of this paper is organized as follows. We recall in Section 2 some basic results about deep neural networks (DNN) and stochastic optimization gradient descent methods used in DNN. Section 3 is devoted to the description of our three algorithms. We analyze in detail in Section 4 the convergence of the three algorithms. Finally the Appendix collect some Lemmas used in the proof of the convergence results.

2   Preliminaries on DNN and SGD

2.1   Neural network approximations

Deep Neural networks (DNN) aim to approximate (complex non linear) functions defined on finite-dimensional space, and in contrast with the usual additive approximation theory built via basis functions, like polynomial, they rely on composition of layers of simple functions. The relevance of neural networks comes from the universal approximation theorem and the Kolmogorov-Arnold representation theorem (see [20], [5] or [16]), and this has shown to be successful in numerous practical applications.

We consider here feedforward artificial network (also called multilayer perceptron) for the approximation of the optimal policy (valued in $\mathbb{A} \subset \mathbb{R}^q$) and the value function (valued in $\mathbb{R}$), both defined on the state space $\mathcal{X} \subset \mathbb{R}^d$. The architecture is depicted in Figure 1, and it is mathematically represented by functions

$$x \in \mathcal{X} \quad \longmapsto \quad \Phi(z; \theta) \in \mathbb{R}^o,$$

with $o = q$ or 1 in our context, and where $\theta \in \Theta \subset \mathbb{R}^p$ are the weights (or parameters) of the neural networks. The DNN function $\Phi = \Phi_L$ with input layer $\Phi_0 = (\Phi_0^i)_i = x \in \mathcal{X}$ composed of d units (or neurons), $L-1$ hidden layers (with layer ℓ composed of d_ℓ units), and output layer composed of $d_L = o$ neurons is obtained by successive composition of linear combination and activation function σ_ℓ (that is a nonlinear monotone function like e.g. the sigmoid, the rectified linear unit ReLU, the exponential linear unit ELU, or the softmax):

$$\Phi_\ell \quad = \quad \sigma_\ell(w_\ell \Phi_{\ell-1} + \gamma_\ell) \ \in \ \mathbb{R}^{d_\ell}, \ \ell = 1, \ldots, L,$$

for some matrix weights (w_ℓ) and vector weight (γ_ℓ), aggregating into $\theta = (w_\ell, \gamma_\ell)_{\ell=1,\ldots,L}$. A key feature of neural networks is the computation of the gradient (with respect to the variable x and the weights θ) of the DNN function via a forward-backward propagation algorithm derived from chain rule composition. For example, for the sigmoid activation function $\sigma_\ell(y) = 1/(1 + e^{-y})$, and noting that $\sigma_\ell' = \sigma_\ell(1 - \sigma_\ell)$, we have

$$\left[\frac{\partial \Phi_\ell}{\partial z}\right]_{ij} \quad = \quad \left[w_\ell \frac{\partial \Phi_{\ell-1}}{\partial z}\right]_{ij} \Phi_\ell^i(1 - \Phi_\ell^i), \ \ \ell = 1, \ldots, L, \ i = 1, \ldots, d_\ell, \ j = 1, \ldots, d$$

while the gradient w.r.t. θ of $\mathcal{K}(\theta) = K(\Phi_L(.; \theta))$, for a real-valued differentiable function $y \in \mathbb{R}^{d_L} \mapsto K(y)$, is given in backward induction by

$$\Delta_i^\ell \quad := \quad \left[\frac{\partial \mathcal{K}}{\partial \Phi_\ell}\right]_i \Phi_\ell^i(1 - \Phi_\ell^i), \quad \ell = L, \ldots, 1, \ i = 1, \ldots, d_\ell$$

$$\left[\frac{\partial \mathcal{K}}{\partial w_\ell}\right]_{ij} \quad = \quad \Phi_{\ell-1}^j \Delta_i^\ell, \ \left[\frac{\partial \mathcal{K}}{\partial \gamma_\ell}\right]_i \ = \ \Delta_i^\ell, \ \left[\frac{\partial \mathcal{K}}{\partial \Phi_{\ell-1}}\right]_j \ = \ \sum_{k=1}^{d_\ell} \Delta_k^\ell w_\ell^{kj}, \quad j = 1, \ldots, d_{\ell-1}.$$

We refer to the online book [27] for a gentle introduction to neural networks and deep learning.

2.2 Stochastic optimization in DNN

Approximation by means of DNN requires a stochastic optimization with respect to a set of parameters, which can be written in a generic form as

$$\inf_\theta \mathbb{E}\left[L_n(Z_n; \theta)\right], \tag{2.1}$$

where Z_n is a random variable from which the training samples $Z_n^{(m)}$, $m = 1, \ldots, M$ are drawn, and L_n is a loss function involving DNN with parameters $\theta \in \mathbb{R}^p$, and typically differentiable w.r.t. θ with known gradient $D_\theta L_n$.
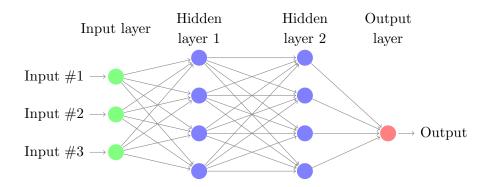
Figure 1: Representation of a neural network with $d = 3$, 2 hidden layers, $d_1 = d_2 = 4$, $d_3 = 1$.

Several basic algorithms are already implemented in TENSORFLOW for the search of infimum in (2.1). Given a training sample of size M, in all the following cases, the sequence $(\theta_n^k)_{k \in \mathbb{N}}$ tends to $\theta_n = \underset{\theta}{\arg\min} \, \mathbb{E}\big[L_n(Z_n; \theta)\big]$ under suitable assumptions on the learning rate sequence $(\gamma_k)_{k=0}^{\infty}$.

- Batch gradient descent: (compute the gradient over the full training set). Fix an integer K, and do

$$\theta_n^{k+1} = \theta_n^k - \gamma_k \frac{1}{M} \sum_{m=1}^{M} D_\theta L_n(Z_n^{(m)}; \theta_n^k), \qquad \text{for } k = 1, \ldots, K.$$

  The main problem with the Batch Gradient Descent is that the convergence is very slow and also the computation of the sum can be painful for very large training sets. Hence it makes it very stable, but too slow in most situations.

- Stochastic gradient descent (SGD): (compute the gradient over one random instance in the training set)

$$\theta_n^{m+1} = \theta_n^m - \gamma_m D_\theta L_n(Z_n^{(m)}; \theta_n^m), \quad m = 1, \ldots, M - 1.$$

  starting from $\theta_n^0 \in \mathbb{R}^p$, with a learning rate γ_m. The Stochastic gradient algorithm computes the gradient based on a single random instance in the training set. It is then a fast but unstable algorithm.

- Mini-batch gradient descent: (compute the gradient over random small subsets of the training set, i.e. mini-batches) let Mb be an integer than divides M. Mb stands for the number of mini-batches and should be taken much smaller than M in the applications.
  For all $k, \ldots, Mb$,

  - Randomly draw a subset $\left(Z_n^{(k,m)}\right)_{m=1}^{M_{k+1}}$ of size $M_{k+1} := \frac{M}{Mb}$ in the training set.
  - iterate: $\theta_n^{k+1} = \theta_n^k - \gamma_k \frac{1}{M_{k+1}} \sum_{m=1}^{M_{k+1}} D_\theta L_n(Z_n^{(m)}; \theta_n^k)$.

The mini-batch gradient descent is often considered to be the best trade-off between speed and stability.

The three gradient descents that we just introduced are the first three historical algorithms that has been designed to learn optimal parameters. Other methods such as the Adaptive optimization methods AdaGrad, RMSProp, and finally Adam are also available. Although not well-understood and even questioned (see e.g. [31]), the latter are often chosen by the practitioners to solve (2.1) and appear to provide the best results in most of the situations.

For sake of simplicity, we only refer in the sequel to the stochastic gradient descent method, when presenting our algorithms. However, we recommend to test and use different algorithms in order to know which are the ones that provide best and fastest results for a given problem.

3   Description of the algorithms

We propose algorithms relying on a DNN approximation of the optimal policy that we compute sequentially in time through the dynamic programming formula, and using performance or hybrid iteration. The value function is then computed by Monte-Carlo regression either by a regress now method or a regress later joint with quantization approach. These variants lead to three algorithms for MDP that we detail in this section.

Let us introduce a set $\mathcal{A}$ of neural networks for approximating optimal policies, that is a set of parametric functions $x \in \mathcal{X} \mapsto A(x; \beta) \in \mathbb{A}$, with parameters $\beta \in \mathbb{R}^l$, and a set $\mathcal{V}$ of neural networks functions for approximating value functions, that is a set of parametric functions $x \in \mathcal{X} \mapsto \Phi(x; \theta) \in \mathbb{R}$, with parameters $\theta \in \mathbb{R}^p$.

We are also given at each time n a probability measure μ_n on the state space $\mathcal{X}$, which we refer to as a training distribution. Some comments about the choice of the training measure are discussed in Section 3.3.

3.1   Control learning by performance iteration

This algorithm, refereed in short as *NNcontPI* Algo, is designed as follows:

• For $n = N-1, \ldots, 0$, we keep track of the approximated optimal policies $\hat{a}_k$, $k = n + 1, \ldots, N-1$, and approximate the optimal policy at time n by $\hat{a}_n = A(.; \hat{\beta}_n)$ with

$$\hat{\beta}_n \in \arg\min_{\beta \in \mathbb{R}^l} \mathbb{E}\left[f(X_n, A(X_n; \beta)) + \sum_{k=n+1}^{N-1} f(\hat{X}_k^\beta, \hat{a}_k(\hat{X}_k^\beta)) + g(\hat{X}_N^\beta) \right], \qquad (3.1)$$

where $X_n \rightsquigarrow \mu_n$, $\hat{X}_{n+1}^\beta = F(X_n, A(X_n; \beta), \varepsilon_{n+1}) \rightsquigarrow P^{A(X_n; \beta)}(X_n, dx')$, and for $k = n + 1, \ldots, N-1$, $\hat{X}_{k+1}^\beta = F(\hat{X}_k^\beta, \hat{a}_k(\hat{X}_k^\beta), \varepsilon_{k+1}) \rightsquigarrow P^{\hat{a}_k(\hat{X}_k^\beta)}(\hat{X}_k^\beta, dx')$. Given estimate $\hat{a}_k^M$ of $\hat{a}_k$, $k = n + 1, \ldots, N-1$, the approximated policy $\hat{a}_n$ is estimated by using a training sample $\left(X_n^{(m)}, (\varepsilon_{k+1}^{(m)})_{k=n}^{k=N-1} \right)$, $m = 1, \ldots, M$ of $\left(X_n, (\varepsilon_{k+1})_{k=n}^{k=N-1} \right)$ for simulating $\left(X_n, (\hat{X}_{k+1}^\beta)_{k=n}^{k=N-1} \right)$, and optimizing over the parameters $\beta \in \mathbb{R}^l$ of the NN $A(.; \beta) \in \mathcal{A}$, the expectation in (3.1) by stochastic gradient descent method (or its variants) as described in Section (2.2).

▶ We then get an estimate of the optimal policy at any time $n = 0, \ldots, N-1$ by:

$$\hat{a}_n^M \quad = \quad A(.; \hat{\beta}_n^M) \ \in \ \mathcal{A},$$

where $\hat{\beta}_n^M$ is the "optimal" parameter resulting from the SGD in (3.1) with a training sample of size M. This leads to an estimated value function given at any time n by

$$\hat{V}_n^M(x) \quad = \quad \mathbb{E}_M \left[\sum_{k=n}^{N-1} f(\hat{X}_k^{n,x}, \hat{a}_k^M(\hat{X}_k^{n,x})) + g(\hat{X}_N^{n,x}) \right], \qquad (3.2)$$

where $\mathbb{E}_M$ is the expectation conditioned on the training set (used for computing $(\hat{a}_k^M)_k$), and $\left(\hat{X}_k^{n,x} \right)_{k=n,\ldots,N}$, is given by: $\hat{X}_n^{n,x} = x$, $\hat{X}_{k+1}^{n,x} \rightsquigarrow P^{\hat{a}_k^M(\hat{X}_k^{n,x})}(\hat{X}_k^{n,x}, dx')$, $k = n, \ldots, N-1$. The dependence of the estimated value function $\hat{V}_n^M$ upon the training samples $X_k^{(m)}$, for $m = 1, \ldots, M$, used at time $k = n, \ldots, N$, is emphasized through the exponent M in the notations.

Remark 3.1 The *NNcontPI* Algo can be viewed as a combination of the DNN algorithm designed in [13] and dynamic programming. In the algorithm presented in [13], which totally ignores the dynamic programming principle, one learns all the optimal controls $A(.; \beta_n)$, $n = 0, \ldots, N-1$ at the same time, by performing one unique stochastic gradient descent. This is efficient as all the parameters of all the NN are getting trained at the same time, using the same mini-batches. However, when the number of layers of the global neural network gathering all the NN $A(.; \beta_n)$, $n = 0, \ldots, N-1$ is large (say $\sum_{n=0}^{N-1} \ell_n \geq 100$, where ℓ_n is the number of layers in $A(., \beta_n)$), then one is likely to observe vanishing or exploding gradient problems that will affect the training of the weights and biais of the first layers of the global NN (see [7] for more details). Therefore, it may be more reasonable to make use of the dynamic programming structure when N is large, and learn the optimal policy sequentially as proposed in our *NNcontPI* Algo. Notice that a similar idea was already used in [11] in the context of uncertain volatility model where the authors use a specific parametrization for the feedback control instead of a DNN adopted more generally here. □

Remark 3.2 The *NNcontPI* Algo does not require value function iteration, but instead is based on performance iteration by keeping track of the estimated optimal policies computed in backward recursion. The value function is then computed in (3.2) as the gain functional associated to the estimated optimal policies $(\hat{a}_k^M)_k$. Consequently, it provides usually a low bias estimate but induces possibly high variance estimate and large complexity, especially when N is large. □

3.2 Control learning by hybrid iteration

Instead of keeping track of all the approximated optimal policies as in the *NNcontPI* Algo, we use an approximation of the value function at time $n+1$ in order to compute the optimal policy at time n. The approximated value function is then updated at time n by relying

on the martingale property (1.4) under the optimal control. This leads to the following generic algorithm:

Generic Hybrid Algo

1. *Initialization*: $\hat{V}_N = g$

2. For $n = N - 1, \dots, 0$,

   (i) Approximate the optimal policy at time n by $\hat{a}_n = A(.; \hat{\beta}_n)$ with

   $$\hat{\beta}_n \quad \in \quad \arg\min_{\beta \in \mathbb{R}^l} \mathbb{E}\left[f(X_n, A(X_n; \beta)) + \hat{V}_{n+1}(X_{n+1}^{A(.,\beta)}) \right], \tag{3.3}$$

   where $X_n \rightsquigarrow \mu_n$, $\hat{X}_{n+1}^{A(.,\beta)} = F(X_n, A(X_n; \beta), \varepsilon_{n+1}) \rightsquigarrow P^{A(X_n;\beta)}(X_n, dx')$.

   (ii) *Updating*: approximate the value function by

   $$\hat{V}_n(x) \quad = \quad \mathbb{E}\left[f(X_n, \hat{a}_n(X_n)) + \hat{V}_{n+1}(X_{n+1}^{\hat{a}_n}) | X_n = x \right]. \tag{3.4}$$

The approximated policy $\hat{a}_n$ is estimated by using a training sample $\left(X_n^{(m)}, \varepsilon_{n+1}^{(m)} \right)$, $m = 1, \dots, M$ of (X_n, ε_{n+1}) to simulate $\left(X_n, X_{n+1}^{A(.;\beta)} \right)$, and optimizing over the parameters $\beta \in \mathbb{R}^l$ of the NN $A(.; \beta) \in \mathcal{A}$, the expectation in (3.3) by stochastic gradient descent method (or its variants) as described in Section (2.2). We then get an estimate $\hat{a}_n^M = A\left(.; \hat{\beta}_n^M \right)$. The approximated value function written as a conditional expectation in (3.4) is estimated according to a Monte Carlo regression, either by a regress now method (in the spirit of [19]) or a regress later (in the spirit of [8] and [3]) joint with quantization approach, and this leads to the following algorithmic variants detailed in the two next paragraphs.

3.2.1 Hybrid-Now Algo

Given an estimate $\hat{a}_n^M$ of the optimal policy at time n, and an estimate $\hat{V}_{n+1}^M$ of $\hat{V}_{n+1}$, we estimate $\hat{V}_n$ by neural networks regression, i.e.,

$$\hat{V}_n^M \quad \in \quad \arg\min_{\Phi(.;\theta) \in \mathcal{V}} \mathbb{E}\left| f(X_n, \hat{a}_n^M(X_n)) + \hat{V}_{n+1}^M(X_{n+1}^{\hat{a}_n^M}) - \Phi(X_n; \theta) \right|^2 \tag{3.5}$$

using samples $X_n^{(m)}$, $X_{n+1}^{\hat{a}_n^M,(m)}$, $m = 1, \dots, M$ of $X_n \rightsquigarrow \mu_n$, and $X_{n+1}^{\hat{a}_n^M,(m)}$ of $X_{n+1}^{\hat{a}_n^M}$. In other words, we have

$$\hat{V}_n^M \quad = \quad \Phi\left(.; \hat{\theta}_n^M \right),$$

where $\hat{\theta}_n^M$ is the "optimal" parameter resulting from the SGD in (3.5) with a training sample of size M.

3.2.2 Hybrid-LaterQ Algo

Given an estimate $\hat{a}_n^M$ of the optimal policy at time n, and an estimate $\hat{V}_{n+1}^M$ of $\hat{V}_{n+1}$, the regress-later approach for estimating $\hat{V}_n$ is achieved in two stages: (a) we first regress/interpolate the estimated value $\hat{V}_{n+1}^M\left(X_{n+1}^{\hat{a}_n^M}\right)$ at time $n+1$ by a NN (or alternatively a Gaussian process) $\Phi(X_{n+1}^{\hat{a}_n^M})$, (b) Analytical formulae are applied to the conditional expectation of this NN of future values $X_{n+1}^{\hat{a}_n^M}$ with respect to the present value X_n, and this is obtained by quantization of the noise (ε_n) driving the dynamics (1.1) of the state process.

The ingredients of the quantization approximation are described as follows:

- We denote by $\hat{\varepsilon}$ a K-quantizer of the E-valued random variable $\varepsilon_{n+1} \rightsquigarrow \varepsilon_1$ (typically a Gaussian random variable), that is a discrete random variable on a grid $\Gamma = \{e_1, \ldots, e_K\} \subset E^K$ defined by

$$\hat{\varepsilon} \;=\; \mathrm{Proj}_\Gamma(\varepsilon_1) \;:=\; \sum_{\ell=1}^{K} e_\ell 1_{\varepsilon_1 \in C_\ell(\Gamma)},$$

  where $C_1(\Gamma)$, ..., $C_K(\Gamma)$ are Voronoi tesselations of Γ, i.e., Borel partitions of the Euclidian space $(E, |.|)$ satisfying

$$C_\ell(\Gamma) \;\subset\; \left\{ e \in E : |e - e_\ell| \;=\; \min_{j=1,\ldots,K} |e - e_j| \right\}.$$

  The discrete law of $\hat{\varepsilon}$ is then characterized by

$$\hat{p}_\ell \;:=\; \mathbb{P}[\hat{\varepsilon} = e_\ell] \;=\; \mathbb{P}[\varepsilon_1 \in C_\ell(\Gamma)], \quad \ell = 1, \ldots, K.$$

  The grid points (e_ℓ) which minimize the L^2-quantization error $\|\varepsilon_1 - \hat{\varepsilon}\|_2$ lead to the so-called optimal L-quantizer, and can be obtained by a stochastic gradient descent method, known as Kohonen algorithm or competitive learning vector quantization (CLVQ) algorithm, which also provides as a byproduct an estimation of the associated weights $(\hat{p}_\ell)$. We refer to [28] for a description of the algorithm, and mention that for the normal distribution, the optimal grids and the weights of the Voronoi tesselations are precomputed on the website http://www.quantize.maths-fi.com

- Recalling the dynamics (1.1), the conditional expectation operator is equal to

$$P^{\hat{a}_n^M(x)} W(x) \;=\; \mathbb{E}\big[W(X_{n+1}^{\hat{a}_n^M}) | X_n = x\big] \;=\; \mathbb{E}\big[W(F(x, \hat{a}_n^M(x), \varepsilon_1))\big], \;\; x \in \mathcal{X},$$

  that we shall approximate analytically by quantization via:

$$\widehat{P}^{\hat{a}_n^M(x)} W(x) \;:=\; \mathbb{E}\big[W(F(x, \hat{a}_n^M(x), \hat{\varepsilon}))\big] \;=\; \sum_{\ell=1}^{K} \hat{p}_\ell W\big(F(x, \hat{a}_n^M(x), e_\ell)\big). \quad (3.6)$$

The two stages of the regress-later are then detailed as follows:

(a) *(Later) interpolation of the value function*: Given a DNN $\Phi\left(.;\theta\right)$ on $\mathbb{R}^d$ with para-meters $\theta \in \mathbb{R}^p$, we interpolate $\hat{V}_{n+1}^M$ by

$$\widetilde{V}_{n+1}^M(x) \quad := \quad \Phi\left(x;\theta_{n+1}^M\right),$$

where θ_{n+1}^M is obtained via SGD (as described in paragraph 2.2) from the regression of $\hat{V}_{n+1}^M(X_{n+1}^{\hat{a}_n^M})$ against $\Phi\left(X_{n+1}^{\hat{a}_n^M};\theta\right)$, using training samples $X_n^{(m)}$, $X_{n+1}^{\hat{a}_n^M,(m)}$, $m = 1,\ldots,M$ of $X_n \rightsquigarrow \mu_n$, and $X_{n+1}^{\hat{a}_n^M,(m)}$ of $X_{n+1}^{\hat{a}_n^M}$.

(b) *Updating/approximation of the value function*: by using the hat operator in (3.6) for the approximation of the conditional expectation by quantization, we calculate analytically

$$\hat{V}_n^M(x) \quad := \quad f(x,a) + \widehat{P}^{\hat{a}_n^M}\widetilde{V}_{n+1}^M(x) \; = \; f(x,a) + \sum_{\ell=1}^K \hat{p}_\ell \Phi\left(F(x,\hat{a}_n^M(x),e_\ell);\theta_{n+1}^M\right).$$

Remark 3.3 Let us discuss and compare the Algos Hybrid-Now and Hybrid-LaterQ. When regressing later, one just has to learn a deterministic function through the interpolation step (a), as the noise is then approximated by quantization for getting analytical formula. Therefore, compared to Hybrid-Now, the Hybrid-LaterQ Algo reduces the variance of the estimate $\hat{V}_n^M$. Moreover, one has a wide choice of loss functions when regressing later, e.g., MSE loss function, $L1$-loss, relative error loss, etc, while the $L2$-loss function is required to approximate of condition expectation using regress-now method. However, although quantization is quite easy and fast to implement in small dimension for the noise, it might be not efficient in high-dimension compared to Hybrid-Now. □

Remark 3.4 Again, we point out that the estimated value function $\hat{V}_n^M$ in Hybrid-Now or Hybrid-LaterQ depend on training samples $X_k^{(m)}$, $m = 1,\ldots,M$, used at times $k = n,\ldots,N$, for computing the estimated optimal policies $\hat{a}_k^M$, and this is emphasized through the exponent M in the notations. □

3.3 Training sets design

We discuss here the choice of the training measure μ_n used to generate the training sets on which will be computed the estimations. Two cases are considered in this section. The first one is a knowledge-based selection, relevant when the controller knows with a certain degree of confidence where the process has to be driven in order to optimize her cost functional. The second case, on the other hand, is when the controller has no idea where or how to drive the process to optimize the cost functional.

Exploitation only strategy

In the knowledge-based setting, there is no need for exhaustive and expensive (in time mainly) exploration of the state space, and the controller can directly choose training sets Γ_n constructed from distributions μ_n that assign more points to the parts of the state space where the optimal process is likely to be driven.

In practice, at time n, assuming we know that the optimal process is likely to stay in the ball centered around the point m_n and with radius r_n, we choose a training measure μ_n centered around m_n as, for example $\mathcal{N}(m_n, r_n^2)$, and build the training set as sample of the latter.

Explore first, exploit later

- *Explore first:* If the agent has no idea of where to drive the process to receive large rewards, she can always proceed to an exploration step to discover favorable subsets of the state space. To do so, Γ_n, the training sets at time n, for $n = 0, \ldots, N-1$, can be built as uniform grids that cover a large part of the state space, or μ can be chosen uniform on such domain. It is essential to explore far enough to have a well understanding of where to drive and where not to drive the process.

- *Exploit later:* The estimates for the optimal controls at time t_n, $n = 0, \ldots, N-1$, that come up from the *Explore first* step, are relatively good in the way that they manage to avoid the wrong areas of state space when driving the process. However, the training sets that have been used to compute the estimated optimal control are too sparse to ensure accuracy on the estimation. In order to improve the accuracy, the natural idea is to build new training sets by simulating M times the process using the estimates on the optimal strategy computed from the *Explore first* step, and then proceed to another estimation of the optimal strategies using the new training sets. This trick can be seen as a two steps algorithm that improves the estimate of the optimal control.

3.4 Some remarks

We end this section with some comments about our proposed algorithms.

3.4.1 Case of finite control space: classification

In the case where the control space $\mathbb{A}$ is finite, i.e., $\mathrm{Card}(\mathbb{A}) = L < \infty$ with $\mathbb{A} = \{a_1, \ldots, a_L\}$, one can think of the optimal control searching task as a problem of classification. This means that we randomize the control in the sense that given a state value x, the controller chooses a_ℓ with a probability $p_\ell(x)$. We can then consider a neural network that takes state x as an input, and returns at each time n a probability vector $p = (p_\ell)_\ell$ with softmax output layer:

$$z \quad \longmapsto \quad S_\ell(z; \beta) \;=\; \frac{\exp(\beta_\ell . z)}{\sum_{\ell=1}^{L} \exp(\beta_\ell . z)}, \quad \ell = 1, \ldots, L,$$

after some hidden layers. Finally, in practice, we use pure strategies given a state value x, choose $a_{\ell^*(x)}$ with

$$\ell^*(x) \quad \in \quad \arg\max_{\ell = 1, \ldots, L} p_\ell(x).$$

For example, the *NNcontPI* Algo with classification reads as follows:

- For $n = N - 1, \ldots, 0$, keep track of the approximated optimal policies $\hat{a}_k$, $k = n + 1, \ldots, N - 1$, and compute

$$
\hat{\beta}_n \; \in \; \arg \min_\beta \mathbb{E}\Big[\sum_{\ell=1}^{L} p_\ell(X_n; \beta)\Big(f(X_n, a_\ell) + \sum_{k=n+1}^{N-1} f\big(\hat{X}_k^\ell, \hat{a}_k(\hat{X}_k^\ell)\big) \; + \; g(\hat{X}_N^\ell)\Big)\Big],
$$

where $X_n \rightsquigarrow \mu_n$, $\hat{X}_{n+1}^\ell = F(X_n, a_\ell, \varepsilon_{n+1})$, $\hat{X}_{k+1}^\ell = F(\hat{X}_k^\ell, \hat{a}_k(\hat{X}_k^\ell), \varepsilon_{n+1})$, for $k = n + 1, \ldots, N - 1$, $\ell = 1, \ldots, L$.

- Update the approximate optimal policy at time n by

$$
\hat{a}_n(x) \; = \; a_{\hat{\ell}_n(x)} \quad \text{with} \quad \hat{\ell}_n(x) \; \in \; \arg \max_{\ell=1,\ldots,L} p_\ell(x; \hat{\beta}_n).
$$

3.4.2 Comparison of the algorithms

We emphasize the pros (+) and cons (-) of the three proposed algorithms in terms of bias estimate for the value function, variance, complexity and dimension for the state space.

Algo	Bias estimate	Variance	Complexity	Dimension	Number of time steps N
NNContPI	+	-	-	+	--
Hybrid-Now	-	+	+	+	+
Hybrid-LaterQ	-	++	+	-	+

This table is the result of observations made when numerically solving various control problems, combined to a close look at the rates of convergence derived for the three algorithms in Theorems 4.1, 4.2 and 4.3. Note that the sensibility of the NNContPI and the Hybrid-LaterQ algorithms w.r.t. the number of time steps N is clearly described in the studies of their rate of convergence achieved in Theorems 4.1 and 4.3. However, we could only provide a weak result on the rate of convergence of the Hybrid algorithm (see Theorem 4.3), which in particular does not explain why the latter does not suffer from large value of N, unless stronger assumptions are made on the loss of the neural network estimating the optimal controls.

4 Convergence analysis

This section is devoted to the convergence of the estimator $\hat{V}_n^M$ of the value function V_n obtained from a training sample of size M and using DNN algorithms listed in Section 3.

Training samples rely on a given family of probability distributions μ_n on $\mathcal{X}$, for $n = 0, \ldots, N$, refereed to as training distribution (see Section 3.3 for a discussion on the choice of μ). For sake of simplicity, we consider that μ_n does not depend on n, and denote then by μ the training distribution. We shall assume that the family of controlled transition probabilities has a density w.r.t. μ, i.e.,

$$
P^a(x, dx') \; = \; r(x, a; x')\mu(dx').
$$

15

We shall assume that r is uniformly bounded in $(x, x', a) \in \mathcal{X}^2 \times \mathbb{A}$, and uniformly Lipschitz w.r.t. (x, a), i.e.,

(Hd) There exists some positive constants $\|r\|_\infty$ and $[r]_L$ s.t.

$$|r(x, a; x')| \leq \|r\|_\infty, \quad \forall x, x' \in \mathcal{X}, \ a \in \mathbb{A},$$
$$|r(x_1, a_1; x') - r(x_2, a_2; x')| \leq [r]_L (|x_1 - x_2| + |a_1 - a_2|), \quad \forall x_1, x_2 \in \mathcal{X}, \ a_1, a_2 \in \mathbb{A}.$$

Remark 4.1 Assumption **(Hd)** is usually satisfied when the state and control space are compacts. While the compactness on the control space $\mathbb{A}$ is not explicitly assumed, the compactness condition on the state space $\mathcal{X}$ turns out to be more crucial for deriving estimates on the estimation error (see Lemma 4.1), and will be assumed to hold true for simplicity. Actually, this compactness condition on $\mathcal{X}$ can be relaxed by truncation and localization arguments (see proposition A.1 in the appendix) by considering a training distribution μ such that **(Hd)** is true and which admits a moment of order 1, i.e. $\int |y| d\mu(y) < +\infty$. □

We shall also assume some boundedness and Lipschitz condition on the reward functions:

(HR) There exists some positive constants $\|f\|_\infty$, $\|g\|_\infty$, $[f]_L$, and $[f]_L$ s.t.

$$|f(x, a)| \leq \|f\|_\infty, \quad |g(x)| \leq \|g\|_\infty, \quad \forall x \in \mathcal{X}, \ a \in \mathbb{A},$$
$$|f(x_1, a_1) - f(x_2, a_2)| \leq [f]_L (|x_1 - x_2| + |a_1 - a_2|),$$
$$|g(x_1) - g(x_2)| \leq [g]_L |x_1 - x_2|, \quad \forall x_1, x_2 \in \mathcal{X}, \ a_1, a_2 \in \mathbb{A}.$$

Under this boundedness condition, it is clear that the value function V_n is also bounded:

$$\|V_n\|_\infty \leq (N - n)\|f\|_\infty + \|g\|_\infty, \quad \forall n \in \{0, ..., N\}.$$

We shall finally assume a Lipschitz condition on the dynamics of the MDP.

(HF) For any $e \in E$, there exists $C(e)$ such that for all couples (x, a) and (x', a') in $\mathcal{X} \times \mathbb{A}$:

$$\left| F(x, a, e) - F(x', a', e) \right| \leq C(e) \left(|x - x'| + |a - a'| \right).$$

In the sequel, we define for any $M \in \mathbb{N}^*$:

$$\rho_M = \mathbb{E}\left[\sup_{1 \leq m \leq M} C(\varepsilon^m) \right],$$

where the $(\varepsilon^m)_m$ is a i.i.d. sample of the noise ε. The rate of convergence of ρ_M toward infinity will play a crucial role to show the convergence of the algorithms.

Remark 4.2 A typical example when **(HF)** holds is the case where F is defined through the time discretization of an Euler scheme, i.e., as

$$F(x, a, \varepsilon) := b(x, a) + \sigma(x, a)\varepsilon,$$

with b and σ Lipschitz-continuous w.r.t. the couple (x,a), and $\varepsilon \sim \mathcal{N}(0, I_d)$, where I_d is the identity matrix of size $d \times d$. Indeed, in this case, it is straightforward to see that $C(\varepsilon) = [b]_L + [\sigma]_L \|\varepsilon\|_d$, where $[b]_L$ and $[\sigma]_L$ stand for the Lipschitz coefficients of b and σ, and $\|.\|_d$ stands for the Euclidean norm in $\mathbb{R}^d$. Moreover, one can show that:

$$\rho_M \leq [b]_L + d[\sigma]_L \sqrt{2\log(2dM)}, \tag{4.1}$$

which implies in particular that

$$\rho_M \underset{M \to +\infty}{=} \mathcal{O}\left(\sqrt{\log(M)}\right).$$

Let us indeed check the inequality (4.1). For this, let us fix some integer $M' > 0$ and let $Z := \sup_{1 \leq m \leq M'} |\epsilon_1^m|$ where ϵ_1^m are i.i.d. such that $\epsilon_1^1 \sim \mathcal{N}(0,1)$. From Jentzen inequality to the r.v. Z and the convex function $z \mapsto \exp(tz)$, where $t > 0$ will be fixed later, we get

$$\exp\left(t\mathbb{E}[Z]\right) \leq \mathbb{E}\left[\exp\left(tZ\right)\right] \leq \mathbb{E}\left[\sup_{1 \leq m \leq M'} \exp\left(t|\epsilon_1^m|\right)\right] \leq \sum_{m=1}^{M'} \mathbb{E}\left[\exp\left(t|\epsilon_1^m|\right)\right] \leq 2M' \exp\left(\frac{t^2}{2}\right),$$

where we used the closed-form expression of the moment generating function of the folded normal distribution[a] to write the last inequality. Hence, we have for all $t > 0$:

$$\mathbb{E}[Z] \leq \frac{\log(2M')}{t} + \frac{t}{2}.$$

We get, after taking $t = \sqrt{2\log(2M')}$:

$$\mathbb{E}[Z] \leq \sqrt{2\log(2M')}. \tag{4.2}$$

Since inequality $\|x\|_d \leq d\|x\|_\infty$ holds for all $x \in \mathbb{R}^d$, we derive

$$\mathbb{E}\left[\sup_{1 \leq m \leq M} C(\varepsilon^m)\right] \leq [b]_L + d[\sigma]_L \mathbb{E}\left[\sup_{1 \leq m \leq dM} C(\epsilon_1^m)\right],$$

and apply (4.2) with $M' = dM$, to complete the proof of (4.1). $\qquad\square$

Remark 4.3 Under **(Hd)**, **(HR)** and **(HF)**, it is straightforward to see from the dynamic programming formula 1.3 that V_n is Lipschitz for all $n = 0, \ldots, N$, with a Lipschitz coefficient $[V_n]_L$, which -can be bounded by the minimum of the two following bounds:

$$\begin{cases} [V_N]_L & = & [g]_L \\ [V_n]_L & \leq & [f]_L + \|V_n\|_\infty [r]_L, \quad \text{for } n = 0, \ldots, N-1. \end{cases}$$

and

$$\begin{cases} [V_N]_L & = & [g]_L \\ [V_n]_L & \leq & \rho_1 \frac{1-\rho_1^{N-n}}{1-\rho_1} + \rho_1^{N-n}[g]_L, \quad \text{for } n = 0, \ldots, N-1, \end{cases}$$

---

[a]The folded normal distribution is defined as the distribution of $|Z|$ where $Z \sim \mathcal{N}_1(\mu, \sigma)$. Its moment generating function is given by $t \mapsto \exp\left(\frac{\sigma^2 t^2}{2} + \mu t\right)\left[1 - \Phi\left(-\frac{\mu}{\sigma} - \sigma t\right)\right] + \exp\left(\frac{\sigma^2 t^2}{2} - \mu t\right)\left[1 - \Phi\left(\frac{\mu}{\sigma} - \sigma t\right)\right]$, where Φ is the c.d.f. of $\mathcal{N}_1(0,1)$.

which holds since we have by standard arguments:

$$\begin{cases} [V_N]_L & = & [g]_L \\ [V_n]_L & \leq & [f]_L + \rho_1 [V_{n+1}]_L \quad \text{for } n = 0, \dots, N-1. \end{cases}$$

Note that we use the usual convention $\frac{1-x^p}{1-x} = p$ for $p \in \mathbb{N}^*$ and $x = 0$. The Lipschitz continuity of V_n plays a significant role to prove the convergence of the Hybrid and the LaterQ algorithms described and studied in sections 4.2 and 4.3. □

4.1 Control learning by performance iteration (NNcontPI)

In this paragraph, we analyze the convergence of the NN control learning by performance iteration as described in Section 3.1. Actually, we shall consider neural networks for the optimal policy with one hidden layer, K neurons with total variation[b] smaller than γ, kernel bounded by η, Relu activation function for the hidden layer, and activation function $\sigma_{\mathbb{A}}$ for the output layer (in order to ensure that the NN is valued in $\mathbb{A}$): this is represented by the parametric set of functions

$$
{}^\eta\mathcal{A}_K^\gamma \;\; := \;\; \Big\{ x \in \mathcal{X} \mapsto A(x; \beta) \;=\; (A_1(x; \beta), \dots, A_q(x; \beta)) \in \mathbb{A},
$$

$$
A_i(x; \beta) = \sigma_{\mathbb{A}} \Big(\sum_{j=1}^K c_{ij}(a_{ij}.x + b_{ij})_+ + c_{0j} \Big), \quad i = 1, \dots, q,
$$

$$
\beta = (a_{ij}, b_{ij}, c_{ij})_{i,j}, \; a_{ij} \in \mathbb{R}^d, \|a_{ij}\| \leq \eta, b_{ij}, c_{ij} \in \mathbb{R}, \sum_{i=0}^K |c_{ij}| \leq \gamma \Big\},
$$

where $\|.\|$ is the Euclidean norm in $\mathbb{R}^d$.

Let K_M, η_M and γ_M be sequences of integers such that

$$
K_M \xrightarrow[M \to \infty]{} \infty, \quad \gamma_M \xrightarrow[M \to \infty]{} \infty, \quad \eta_M \xrightarrow[M \to \infty]{} \infty,
$$

$$
\rho_M^{N-1} \gamma_M^{N-1} \eta_M^{N-2} \sqrt{\frac{\log(M)}{M}} \xrightarrow[M \to \infty]{} 0. \tag{4.3}
$$

We denote by $\mathcal{A}_M := {}^{\eta_M}\mathcal{A}_{k_M}^{\gamma_M}$ the class of neural network for policy with norm η_M on the kernel $a = (a_{ij})$, K_M neurons and norm γ_M that satisfy conditions (4.3).

Remark 4.4 In the case where F is defined in dimension d as: $F(x, a, \varepsilon) = b(x, a) + \sigma(x, a)\varepsilon$, we can use (4.1) to bound ρ_M^{N-n} and get:

$$
\rho_M^{N-n} \underset{M \to +\infty}{=} \mathcal{O}\left(\sqrt{\log(M)}^{N-n} \right).
$$

□

---

[b]The total variation for the class of NN $\mathcal{A}_K^\gamma$ is equal to $\sum_{i=0}^K |c_{ij}|$ with the notations above. See e.g. [1] for a general definition.

Recall that the approximation of the optimal policy in the NNcontPI algorithm is computed in backward induction as follows: For $n = N-1, \ldots, 0$, generate a training sample for the state $X_n^{(m)}$, $m = 1, \ldots, M$ from the training distribution μ, and samples of the exogenous noise $\left(\varepsilon_k^m\right)_{m=1, k=n+1}^{M,N}$.

- Compute the approximated policy at time n

$$\hat{a}_n^M \quad \in \quad \operatorname*{argmin}_{A \in \mathcal{A}_M} \frac{1}{M} \sum_{m=1}^{M} \left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] \tag{4.4}$$

where

$$\hat{Y}_{n+1}^{(m),A} = \sum_{k=n+1}^{N-1} f\left(X_k^{(m),A}, \hat{a}_k^M\left(X_k^{(m),A} \right) \right) + g\left(X_N^{(m),A} \right), \tag{4.5}$$

with $\left(X_k^{(m),A} \right)_{k=n+1}^{N}$ defined by induction as follows, for $m=1, \ldots, M$:

$$\begin{cases} X_{n+1}^{(m),A} &= F\left(X_n^m, A\left(X_n^m \right), \varepsilon_{n+1}^m \right) \\ X_k^{(m),A} &= F\left(X_{k-1}^{(m),A}, A\left(X_{k-1}^{(m),A} \right), \varepsilon_k^m \right), \quad \text{for } k = n+2, \ldots, N. \end{cases}$$

- Compute the estimated value function $\hat{V}_n^M$ as in (3.2).

Remark 4.5 In order to simplify the theoretical analysis, we assume that the argmin in (4.4) is exactly reached by running batch, mini-batch or stochastic gradient descent, which are the methods that we used to code the algorithm in our companion paper. □

Remark 4.6 The minimization problem in (4.4) is actually a problem of minimization over the parameter β (of the neural network A) of the expectation of a function of noises $\left(X_n^{(m)} \right)_{m=1}^{M}, \left(\varepsilon_k^m \right)_{m=1, k=n+1}^{M,N}$ and β, where F is iterated many times. Stochastic-gradient-based methods are chosen for such a task, although the gradient becomes more and more difficult to compute when we are going backward in time, since there are more and more iterations of F involved in the derivatives of the gradients.

The integrand is differentiable if assumption **(HF)** holds, but it is always possible to apply the stochastic-gradient-based algoritm for certain classes of non-differentiable functions F (see e.g. the gradient-descent implementation in TENSORFLOW which works with the non-differentiable at 0 ReLu activation functions.). □

We now state our main result about the convergence of the NNcontPI algorithm.

Theorem 4.1 *Assume that there exists an optimal feedback control* $(a_k^{\mathrm{opt}})_{k=n,\ldots,N-1}$ *for the control problem with value function* V_n, $n = 0, \ldots, N$, *and let* $X_n \rightsquigarrow \mu$. *Then, as* $M \to \infty^{\mathrm{c}}$

$$\mathbb{E}\left[\hat{V}_n^M(X_n) - V_n(X_n) \right] \quad = \quad \mathcal{O}\Bigg(\frac{\rho_M^{N-n-1} \gamma_M^{N-n-1} \eta_M^{N-n-2}}{\sqrt{M}} \tag{4.6}$$

$$+ \sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E}\left[\left| A(X_k) - a_k^{\mathrm{opt}}(X_k) \right| \right] \Bigg),$$

---

^cThe notation $x_M = \mathcal{O}(y_M)$ as $M \to \infty$, means that the ratio $|x_M|/|y_M|$ is bounded as M goes to infinity.

where $\mathbb{E}$ stands for the expectation over the training set used to evaluate the approximated optimal policies $(\hat{a}_k^M)_{n \leq k \leq N-1}$, as well as the path $(X_n)_{n \leq k \leq N}$ controlled by the latter. Moreover, as $M \to \infty$[d]

$$\mathbb{E}_M\big[\hat{V}_n^M(X_n) - V_n(X_n)\big] = \mathcal{O}_{\mathbb{P}}\bigg(\rho_M^{N-n-1}\gamma_M^{N-n-1}\eta_M^{N-n-2}\sqrt{\frac{\log(M)}{M}}$$

$$+ \sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E}\Big[|A(X_k) - a_k^{\mathrm{opt}}(X_k)|\Big]\bigg), \tag{4.7}$$

where $\mathbb{E}_M$ stands for the expectation conditioned by the training set used to estimate the optimal policies $(\hat{a}_k^M)_{n \leq k \leq N-1}$.

Remark 4.7 1. The term $\frac{\rho_M^{N-n-1}\gamma_M^{N-n-1}\eta_M^{N-n-2}}{\sqrt{M}}$ should be seen as the estimation error. It is due to the approximation of the optimal controls by means of neural networks in $\mathcal{A}_M$ using *empirical* cost functional in (4.4). We show in section A.2 that this term disappears in the ideal case where the real cost functional (i.e. not the empirical one) is minimized.

2. The rate of convergence depends dramatically on N since it becomes exponentially slower when N goes to infinity. This is a huge drawback for this performance iteration-based algorithm. We will see in the next section that the rate of convergence of value iteration-based algorithms do not suffer from this dramatical dependence on N. □

Comment: Since we clearly have $V_n \leq \hat{V}_n^M$, estimation (4.6) implies the convergence in L^1 norm of the NNcontPI algorithm, under condition (4.3), and in the case where $\sup_{n \leq k \leq N} \inf_{A \in \mathcal{A}_M} \mathbb{E}\big[|A(X_k) - a_k^{\mathrm{opt}}(X_k)|\big] \xrightarrow[M \to +\infty]{} 0$. This is actually the case under some regularity assumptions on the optimal controls, as stated in the following proposition.

Proposition 4.1 *The two following assertions hold:*

1. *Assume that $a_k^{\mathrm{opt}}(X_k) \in \mathbb{L}^1(\mu)$ for $k = n, ..., N-1$. Then*

$$\sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E}\big[|A(X_k) - a_k^{\mathrm{opt}}(X_k)|\big] \xrightarrow[M \to +\infty]{} 0. \tag{4.8}$$

2. *Assume that the function a_k^{opt} is c-Lipschitz for $k = n, ..., N-1$. Then*

$$\sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E}\big[|A(X_k) - a_k^{\mathrm{opt}}(X_k)|\big] < c\left(\frac{\gamma_M}{c}\right)^{-2d/(d+1)} \log\left(\frac{\gamma_M}{c}\right) + \gamma_M K_M^{-(d+3)/(2d)}. \tag{4.9}$$

Proof. The first statement of Proposition 4.1 relies essentially on the universal approximation theorem, and the second assertion is stated and proved in [1]. For sake of completeness, we recall the details in Section A.5 in the Appendix. □

---

[d]The notation $x_M = \mathcal{O}_{\mathbb{P}}(y_M)$ as $M \to \infty$, means that there exists $c > 0$ such that $\mathbb{P}\big(|x_M| > c|y_M|\big) \to 0$ as M goes to infinity.

Remark 4.8 Note that the second statement in the above proposition is stronger than the first one since it provides a rate of convergence of the approximation error. Fixing K_M and minimizing the r.h.s. of (4.9) over γ_M, results in

$$\sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E}\big[|A(X_k) - a_k^{\mathrm{opt}}(X_k)|\big] < cK_M^{-\frac{1}{d}}\left(1 + \frac{d+1}{2d}\log\left(K_M\right)\right),$$

when we take $\gamma_M = cK_M^{\frac{d+1}{2d}}$. Hence, for such a value of γ_M, the l.h.s. decreases to 0 with rate proportional to $\log(K_M)/\sqrt[d]{K_M}$. $\hfill\square$

The rest of this section is devoted to the proof of Theorem 4.1. Let us introduce some useful notations. Denote by $\mathbb{A}^{\mathcal{X}}$ the set of Borelian functions from the state space $\mathcal{X}$ into the control space $\mathbb{A}$. For $n = 0, \ldots, N-1$, and given a feedback control (policy) represented by a sequence $(A_k)_{k=n,\ldots,N-1}$, with A_k in $\mathbb{A}^{\mathcal{X}}$, we denote by $J_n^{(A_k)_{k=n}^{N-1}}$ the cost functional associated to the policy $(A_k)_k$. Notice that with this notation, we have $\hat{V}_n^M = J_n^{(\hat{a}_k^M)_{k=n}^{N-1}}$. We define the *estimation error* at time n associated to the NNContPI algorithm by

$$\varepsilon_{\mathrm{PI},n}^{\mathrm{esti}} \quad := \quad \sup_{A \in \mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^M\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right] - \mathbb{E}_M\big[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\big]\right|,$$

with $X_n \rightsquigarrow \mu$: It measures how well the chosen estimator (e.g. mean square estimate) can approximate a certain quantity (e.g. the conditional expectation). Of course we expect the latter to cancel when the size of the training set used to build the estimator goes to infinity. Actually, we have

Lemma 4.1 *For $n = 0, \ldots, N-1$, we have*

$$\mathbb{E}[\varepsilon_{\mathrm{PI},n}^{\mathrm{esti}}] \leq \left(\sqrt{2} + 16\right)\frac{\left((N-n)\|f\|_\infty + \|g\|_\infty\right)}{\sqrt{M}}$$

$$+ \frac{16\gamma_M}{\sqrt{M}}\left\{[f]_L\left(1 + \rho_M\frac{1 - \rho_M^{N-n-1}\left(1 + \eta_M\gamma_M\right)^{N-n-1}}{1 - \rho_M\left(1 + \eta_M\gamma_M\right)}\right)\right.$$

$$\left. + \left(1 + \eta_M\gamma_M\right)^{N-n-1}\rho_M^{N-n}[g]_L\right\} \quad (4.10)$$

$$= \mathcal{O}\left(\frac{\rho_M^{N-n-1}\gamma_M^{N-n-1}\eta_M^{N-n-2}}{\sqrt{M}}\right), \qquad as\ M \to \infty.$$

This implies in particular that

$$\varepsilon_{\mathrm{PI},n}^{\mathrm{esti}} = \mathcal{O}_\mathbb{P}\left(\rho_M^{N-n-1}\gamma_M^{N-n-1}\eta_M^{N-n-2}\sqrt{\frac{\log(M)}{M}}\right), \qquad as\ M \to \infty, \qquad (4.11)$$

where we remind that $\rho_M = \mathbb{E}\left[\displaystyle\sup_{1 \leq m \leq M} C(\varepsilon^m)\right]$ is defined in **(HF)**.

Proof. The relation (4.10) states that the estimation error cancels when $M \to \infty$ with a rate of convergence of order $\mathcal{O}\left(\frac{\rho_M^{N-n-1}\gamma_M^{N-n-1}\eta_M^{N-n-2}}{\sqrt{M}}\right)$. The proof is in the spirit of the one that can be found in chapter 9 of [12]. It relies on a technique of symmetrization by a ghost sample, and a wise introduction of additional randomness by random signs. The details are postponed in Section A.3 in the Appendix. The proof of (4.11) follows from (4.10) by a direct application of Markov inequality. $\square$

Let us also define the *approximation error* at time n associated to the NNContPI algorithm by

$$\varepsilon_{\text{PI},n}^{\text{approx}} := \inf_{A \in \mathcal{A}_M} \mathbb{E}_M\left[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right] - \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M\left[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right], \qquad (4.12)$$

where we recall that $\mathbb{E}_M$ denotes the expectation conditioned by the training set used to compute the estimates $(\hat{a}_k^M)_{k=n+1}^{N-1}$ and the one of $X_n \rightsquigarrow \mu$.
$\varepsilon_{\text{PI},n}^{\text{approx}}$ measures how well the regression function can be approximated by means of neural networks functions in $\mathcal{A}_M$ (notice that the class of neural networks is not dense in the set $\mathbb{A}^{\mathcal{X}}$ of all Borelian functions).

Lemma 4.2 *For* $n = 0, \ldots, N-1$, *it holds as* $M \to \infty$,

$$\mathbb{E}[\varepsilon_{\text{PI},n}^{\text{approx}}] = \mathcal{O}\left(\frac{\rho_M^{N-n-1}\gamma_M^{N-n-1}\eta_M^{N-n-2}}{\sqrt{M}} + \sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E}\left[|A(X_k) - a_k^{\text{opt}}(X_k)|\right]\right). \tag{4.13}$$

This implies in particular

$$\varepsilon_{\text{PI},n}^{\text{approx}} = \mathcal{O}_{\mathbb{P}}\left(\rho_M^{N-n-1}\gamma_M^{N-n-1}\eta_M^{N-n-2}\sqrt{\frac{\log(M)}{M}} + \sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E}\left[|A(X_k) - a_k^{\text{opt}}(X_k)|\right]\right). \tag{4.14}$$

Proof. See Section A.4 in Appendix for the proof of (4.13). The proof of (4.14) then follows by a direct application of Markov inequality. $\square$

Proof of Theorem 4.1.
Step 1. Let us denote by

$$\hat{J}_{n,M}^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}} := \frac{1}{M}\sum_{m=1}^{M}\left[f\left(X_n^{(m)}, A(X_n^{(m)})\right) + \hat{Y}_{n+1}^{(m),A}\right],$$

the empirical cost function, from time n to N, associated to the sequence of controls $(A, (\hat{a}_k^M)_{k=n+1}^{N-1}, , \hat{a}_{N-1}^M)$ and the training set, where we recall that $\hat{Y}_{n+1}^{(m),A}$ is defined in (4.5). We then have

$$\mathbb{E}_M\left[\hat{V}_n^M(X_n)\right] = \mathbb{E}_M\left[J_n^{(\hat{a}_k^M)_{k=n}^{N-1}}(X_n)\right] - \hat{J}_{n,M}^{(\hat{a}_k^M)_{k=n}^{N-1}} + \hat{J}_{n,M}^{(\hat{a}_k^M)_{k=n}^{N-1}}$$

$$\leq \varepsilon_{\text{PI},n}^{\text{esti}} + \hat{J}_{n,M}^{(\hat{a}_k^M)_{k=n}^{N-1}}, \tag{4.15}$$

22

by definition of $\hat{V}_n^M$ and $\varepsilon_{\mathrm{PI},n}^{\mathrm{esti}}$. Moreover, for any $A \in \mathcal{A}_M$,

$$
\hat{J}_{n,M}^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}} = \hat{J}_{n,M}^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}} - \mathbb{E}_M\Big[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\Big] + \mathbb{E}_M\Big[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\Big]
$$
$$
\leq \varepsilon_{\mathrm{PI},n}^{\mathrm{esti}} + \mathbb{E}_M\Big[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\Big]. \tag{4.16}
$$

Recalling that

$$
\hat{a}_n^M = \operatorname*{argmin}_{A \in \mathcal{A}_M} \hat{J}_{n,M}^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}},
$$

and taking the infimum over $\mathcal{A}_M$ in the l.h.s. of (4.16) first, and in the r.h.s. secondly, we then get

$$
\hat{J}_{n,M}^{(\hat{a}_k^M)_{k=n}^{N-1}} \leq \varepsilon_{\mathrm{PI},n}^{\mathrm{esti}} + \inf_{A \in \mathcal{A}_M} \mathbb{E}_M\Big[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\Big].
$$

Plugging this last inequality into (4.15) yields the following estimate

$$
\mathbb{E}_M\big[\hat{V}_n^M(X_n)\big] - \inf_{A \in \mathcal{A}_M} \mathbb{E}_M\Big[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\Big] \leq 2\varepsilon_{\mathrm{PI},n}^{\mathrm{esti}}. \tag{4.17}
$$

Step 2. By definition (4.12) of the approximation error, using the law of iterated conditional expectations for J_n, and the dynamic programming principle for V_n with the optimal control a_n^{opt} at time n, we have

$$
\inf_{A \in \mathcal{A}_M} \mathbb{E}_M\big[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\big] - \mathbb{E}_M[V_n(X_n)]
$$
$$
= \varepsilon_{\mathrm{PI},n}^{\mathrm{approx}} + \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M\big\{f(X_n, A(X_n)) + \mathbb{E}_n^A\big[J_{n+1}^{(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_{n+1})\big]\big\}
$$
$$
- \mathbb{E}_M\Big[f(X_n, a_n^{\mathrm{opt}}(X_n)) + \mathbb{E}_n^{a_n^{\mathrm{opt}}}\big[V_{n+1}(X_{n+1})\big]\Big]
$$
$$
\leq \varepsilon_{\mathrm{PI},n}^{\mathrm{approx}} + \mathbb{E}_M \mathbb{E}_n^{a_n^{\mathrm{opt}}}\Big[J_{n+1}^{(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_{n+1}) - V_{n+1}(X_{n+1})\Big],
$$

where $\mathbb{E}_n^A[.]$ stands for the expectation conditioned by X_n at time n and the training set, when strategy A is followed at time n. Under the bounded density assumption in **(Hd)**, we then get

$$
\inf_{A \in \mathcal{A}_M} \mathbb{E}_M\big[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\big] - \mathbb{E}_M[V_n(X_n)]
$$
$$
\leq \varepsilon_{\mathrm{PI},n}^{\mathrm{approx}} + \|r\|_\infty \int \big[J_{n+1}^{(\hat{a}_k^M)_{k=n+1}^{N-1}}(x') - V_{n+1}(x')\big]\mu(dx')
$$
$$
\leq \varepsilon_{\mathrm{PI},n}^{\mathrm{approx}} + \|r\|_\infty \mathbb{E}_M\Big[\hat{V}_{n+1}^M(X_{n+1}) - V_{n+1}(X_{n+1})\Big], \text{ with } X_{n+1} \sim \mu. \tag{4.18}
$$

Step 3. From (4.17) and (4.18), we have

$$
\mathbb{E}_M\big[\hat{V}_n^M(X_n) - V_n(X_n)\big] = \mathbb{E}_M\big[\hat{V}_n^M(X_n)\big] - \inf_{A \in \mathcal{A}_M} \mathbb{E}_M\big[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\big]\big]
$$
$$
+ \inf_{A \in \mathcal{A}_M} \mathbb{E}_M\big[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\big] - \mathbb{E}_M[V_n(X_n)]
$$
$$
\leq 2\varepsilon_{\mathrm{PI},n}^{\mathrm{esti}} + \varepsilon_{\mathrm{PI},n}^{\mathrm{approx}}
$$
$$
+ \|r\|_\infty \mathbb{E}_M\Big[\hat{V}_{n+1}^M(X_{n+1}) - V_{n+1}(X_{n+1})\Big]. \tag{4.19}
$$

By induction, this implies

$$\mathbb{E}_M\big[\hat{V}_n^M(X_n) - V_n(X_n)\big] \leq \sum_{k=n}^{N-1} \big(2\varepsilon_{\text{PI},k}^{\text{esti}} + \varepsilon_{\text{PI},k}^{\text{approx}}\big).$$

Use the estimations (4.11) for $\varepsilon_{\text{PI},n}^{\text{esti}}$ in Lemma 4.1, and (4.14) for $\varepsilon_{\text{PI},n}^{\text{approx}}$ in Lemma 4.2, and observe that $\hat{V}_n(X_n) \geq V_n(X_n)$ holds a.s., to complete the proof of (4.7). Finally, the proof of (4.6) is obtained by taking expectation in (4.19), and using estimations (4.10) and (4.13). □

4.2 Hybrid-Now algorithm

In this paragraph, we analyze the convergence of the hybrid-now algorithm as described in Section 3.2.1. We shall consider neural networks for the value function estimation with one hidden layer, K neurons with total variation γ, kernel bounded by η, a sigmoid activation function σ for the hidden layer, and no activation function for the output layer (i.e. the last layer): this is represented by the parametric set of functions

$$\substack{\eta}\mathcal{V}_K^\gamma \;\; := \;\; \Big\{ x \in \mathcal{X} \mapsto \Phi(x;\theta) \;=\; \sum_{i=1}^K c_i\sigma(a_i.x+b_i) + c_0,$$

$$\theta = (a_i,b_i,c_i)_i, \quad \|a_i\| \leq \eta, \; b_i \in \mathbb{R}, \; \sum_{i=0}^K |c_i| \leq \gamma \Big\}.$$

Let η_M, K_M and γ_M be integers such that:

$$\begin{aligned}
&\eta_M \xrightarrow[M\to\infty]{} \infty &,\quad& \gamma_M \xrightarrow[M\to\infty]{} +\infty &,\quad& K_M \xrightarrow[M\to\infty]{} \infty, \\
&\frac{\gamma_M^4 K_M \log(M)}{M} \xrightarrow[M\to\infty]{} 0 &,\quad& \frac{\gamma_M^4 \rho_M^2 \eta_M^2 \log(M)}{M} \xrightarrow[M\to\infty]{} 0,
\end{aligned} \tag{4.20}$$

where we remind that ρ_M is defined in **(HF)**.

In the sequel we denote by $\mathcal{V}_M := {}^{\eta_M}\mathcal{V}_{K_M}^{\gamma_M}$ the space of neural networks for the estimated value functions at time $n = 0, \ldots, N-1$, parametrized by the values η_M, γ_M and K_M that satisfy (4.20). We also consider the class $\mathcal{A}_M$ of neural networks for estimated feedback optimal control at time $n = 0, \ldots, N-1$, as described in Section 4.1, with the same parameters η_M, γ_M and K_M.

Recall that the approximation of the value function and optimal policy in the hybrid-now algorithm is computed in backward induction as follows:

- Initialize $\hat{V}_N^M = g$

- For $n = N-1, \ldots, 0$, generate a training sample $X_n^{(m)}$, $m = 1, \ldots, M$ from the training distribution μ, and a training sample for the exogenous noise $\varepsilon_{n+1}^{(m)}$, $m = 1, \ldots, M$.

  (i) compute the approximated policy at time n

  $$\hat{a}_n^M \;\; \in \;\; \operatorname*{argmin}_{A \in \mathcal{A}_M} \frac{1}{M}\sum_{m=1}^M \big[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{V}_{n+1}^M(X_{n+1}^{(m),A}) \big]$$

  where $X_{n+1}^{(m),A} = F(X_n^{(m)}, A(X_n^{(m)}), \varepsilon_{n+1}^{(m)}) \rightsquigarrow P^{A(X_n^{(m)})}(X_n^{(m)}, dx')$.

(ii) compute the untruncated estimation of the value function at time n

$$\tilde{V}_n^M \quad \in \quad \underset{\Phi \in \mathcal{V}_M}{\operatorname{argmin}} \frac{1}{M} \sum_{m=1}^{M} \left[f(X_n^{(m)}, \hat{a}_n^M(X_n^{(m)})) + \hat{V}_{n+1}^M(X_{n+1}^{(m),\hat{a}_n^M}) - \Phi(X_n^{(m)}) \right]^2$$

and set the truncated estimated value function at time n

$$\hat{V}_n^M \quad = \quad \max\left(\min\left(V_n^M, \|V_n\|_\infty \right), -\|V_n\|_\infty \right). \tag{4.21}$$

Remark 4.9 Notice that we have truncated the estimated value function in (4.21) by an *a priori* bound on the true value function. This truncation step is natural from a practical implementation point of view, and is also used for simplifying the proof of the convergence of the algorithm. The conditions in (4.20) for the parameters are weaker than those required in (4.3) for the NNcontPI algo by performance iteration, which implies a much faster convergence w.r.t. the size of the training set. However, one should keep in mind that unlike the performance iteration procedure, the value iteration one is biased since the computation of $\hat{V}_{n+1}^M(X_{n+1}^A)$ are biased future rewards when decision A is taken at time n and estimated optimal strategies are taken at time $k \geq n+1$. $\qquad\square$

We now state our main result about the convergence of the Hybrid-Now algorithm.

Theorem 4.2 *Assume that there esxists an optimal feedback control* $(a_k^{\mathrm{opt}})_{k=n,\dots,N-1}$ *for the control problem with value function* V_n, $n = 0,\dots,N$, *and let* $X_n \rightsquigarrow \mu$. *Then, as* $M \to +\infty$

$$\mathbb{E}_M\left[|\hat{V}_n^M(X_n) - V_n(X_n)| \right] = \mathcal{O}_\mathbb{P}\left(\left(\gamma_M^4 \frac{K_M \log(M)}{M} \right)^{\frac{1}{2(N-n)}} + \left(\gamma_M^4 \frac{\rho_M^2 \eta_M^2 \log(M)}{M} \right)^{\frac{1}{4(N-n)}} \right.$$

$$+ \sup_{n \leq k \leq N} \inf_{\Phi \in \mathcal{V}_M} \left(\mathbb{E}_M\left[|\Phi(X_k) - V_k(X_k)|^2 \right] \right)^{\frac{1}{2(N-n)}}$$

$$\left. + \sup_{n \leq k \leq N} \inf_{A \in \mathcal{A}_M} \left(\mathbb{E}\left[|A(X_k) - a_k^{\mathrm{opt}}(X_k)| \right] \right)^{\frac{1}{2(N-n)}} \right),$$

$$\tag{4.22}$$

where $\mathbb{E}_M$ *stands for the expectation conditioned by the training set used to estimate the optimal policies* $(\hat{a}_k^M)_{n \leq k \leq N-1}$.

Comment: Theorem 4.2 states that the estimator for the value function provided by hybrid-now algorithm converges in $\mathbb{L}^1(\mu)$ when the size of the training set goes to infinity. Note that the term $\left(\gamma_M^4 \frac{K_M \log(M)}{M} \right)^{\frac{1}{2(N-n)}}$ stands for the estimation error made by estimating *empirically* the value functions using neural networks, and $\left(\gamma_M^4 \frac{\rho_M^2 \eta_M^2 \log(M)}{M} \right)^{\frac{1}{4(N-n)}}$ stands for the estimation error made by estimating *empirically* the optimal control using neural networks. The term $\sup_{n \leq k \leq N} \inf_{\Phi \in \mathcal{V}_M} \sqrt{\mathbb{E}\left[|\Phi(X_k) - V_k(X_k)|^2 \right]}$ stands for the approximation error made by estimating the value function as a neural network function in $\mathcal{V}_M$,

and $\sup_{n \le k \le N} \inf_{A \in \mathcal{A}_M} \mathbb{E}\Big[\big|A(X_k) - a_k^{\mathrm{opt}}(X_k)\big|\Big]$ is the one made by estimating the optimal control as a neural network function in $\mathcal{A}_M$.

In order to prove Theorem 4.2, let us first introduce the estimation error at time n associated to the Hybrid-Now algorithm by

$$\varepsilon_{\mathrm{HN},n}^{\mathrm{esti}} := \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^{M} \left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] \right.$$
$$\left. - \mathbb{E}_{M,n,X_n}^A \left[f(X_n, A(X_n)) + \hat{V}_{n+1}^M(X_{n+1}) \right] \right|,$$

where

$$\hat{Y}_{n+1}^{(m),A} = \hat{V}_{n+1}^M(X_{n+1}^{(m),A}),$$

and $X_{n+1}^{(m),A} = F\left(X_n^{(m)}, A(X_n^{(m)}), \varepsilon_{n+1}^m \right)$.

We have the following bound on this estimation error:

Lemma 4.3 *For $n = 0, ..., N - 1$, it holds:*

$$\mathbb{E}\left[\varepsilon_{\mathrm{HN},n}^{\mathrm{esti}}\right] \le \frac{\left(\sqrt{2} + 16\right)\left((N-n)\|f\|_\infty + \|g\|_\infty\right) + 16[f]_L}{\sqrt{M}} + 16 \frac{\rho_M \eta_M \gamma_M^2}{\sqrt{M}}$$
$$\underset{M \to \infty}{=} \mathcal{O}\left(\frac{\rho_M \eta_M \gamma_M^2}{\sqrt{M}} \right). \tag{4.23}$$

Proof. See Section A.6 in Appendix. □

Remark 4.10 The result stated by lemma 4.3 is sharper than the one stated in Lemma 4.1 for the performance iteration procedure. The main reason is that we can make use of the $\gamma_M \eta_M$-Lipschitz-continuity of the estimation of the value function at time $n+1$. □

We secondly introduce the approximation error at time n associated to the Hybrid-Now algorithm by

$$\varepsilon_{\mathrm{HN},n}^{\mathrm{approx}} := \inf_{A \in \mathcal{A}_M} \mathbb{E}_M\left[f\big(X_n, A(X_n)\big) + \hat{Y}_{n+1}^A\right] - \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M\left[f\big(X_n, A(X_n)\big) + \hat{Y}_{n+1}^A\right],$$

where $\hat{Y}_{n+1}^A := \hat{V}_{n+1}^M\left(F\left(X_n, A(X_n), \varepsilon_{n+1}\right)\right)$.

We have the following bound on this approximation error:

Lemma 4.4 *For $n = 0, ..., N - 1$, it holds:*

$$\varepsilon_{\mathrm{HN},n}^{\mathrm{approx}} \le ([f]_L + \|V_{n+1}\|_\infty [r]_L) \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M\left[\big|A(X_n) - a_n^{\mathrm{opt}}(X_n)\big|\right]$$
$$+ 2\|r\|_\infty \mathbb{E}_M\left[\Big|V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1})\Big|\right]. \tag{4.24}$$

26

Proof. See Section A.7 in Appendix. □

Proof of Theorem 4.2

Observe that not only the optimal strategy but also the value function is estimated at each time step $n = 0, ..., N - 1$ using neural networks in the hybrid algorithm. It spurs us to introduce the following auxiliary process $(\bar{V}_n^M)_{n=0}^N$ defined by backward induction as:

$$
\begin{cases}
\bar{V}_N^M(x) &= g(x), \quad \text{for } x \in \mathcal{X}, \\
\bar{V}_n^M(x) &= f(x, \hat{a}_n^M(x)) + \mathbb{E}\left[\hat{V}_{n+1}^M(F(x, \hat{a}_n^M(x), \varepsilon_{n+1}))\right], \quad \text{for } x \in \mathcal{X},
\end{cases}
$$

and we notice that for $n = 0, ..., N - 1$, $\bar{V}_n^M$ is the quantity estimated by $\hat{V}_n^M$.

Step 1. We state the following estimates: for $n = 0, ..., N - 1$,

$$
0 \leq \mathbb{E}_M \left[\bar{V}_n^M(X_n) - \inf_{a \in A} \left\{ f(X_n, a) + \mathbb{E}_{M,n,X_n}^a \left[\hat{V}_{n+1}^M(X_{n+1})\right] \right\} \right] \leq 2\varepsilon_{\mathrm{HN},n}^{\mathrm{esti}} + \varepsilon_{\mathrm{HN},n}^{\mathrm{approx}}, \quad (4.25)
$$

and,

$$
\mathbb{E}_M \left[\left| \bar{V}_n^M(X_n) - \inf_{a \in A} \left\{ f(X_n, a) + \mathbb{E}_{M,n,X_n}^a \left[\hat{V}_{n+1}^M(X_{n+1})\right] \right\} \right|^2 \right]
$$
$$
\leq 2\left((N - n)\|f\|_\infty + \|g\|_\infty\right) \left(2\varepsilon_{\mathrm{HN},n}^{\mathrm{esti}} + \varepsilon_{\mathrm{HN},n}^{\mathrm{approx}}\right), \quad (4.26)
$$

where $\mathbb{E}_{M,n,X_n}$ stands for the expectation conditioned by the training set and X_n.

Let us first show the estimate (4.25). Note that inequality

$$
\bar{V}_n^M(X_n) - \inf_{a \in A} \left\{ f(X_n, a) + \mathbb{E}_{M,n,X_n}^a \left[\hat{V}_{n+1}^M(X_{n+1})\right] \right\} \geq 0
$$

holds because $\hat{a}_n^M$ cannot do better than the optimal strategy. Take its expectation to get the first inequality in (4.25). Moreover, we write

$$
\mathbb{E}_M \left[\bar{V}_n^M(X_n)\right] \leq \mathbb{E}_M \left[f\left(X_n, \hat{a}_n^M(X_n)\right) + \hat{V}_{n+1}^M \left(X_{n+1}^{\hat{a}_n^M}\right) \right]
$$
$$
\leq \inf_{A \in \mathcal{A}_M} \mathbb{E}_M \left[f\left(X_n, A(X_n)\right) + \hat{V}_{n+1}^M \left(X_{n+1}^A\right) \right] + 2\varepsilon_{\mathrm{HN},n}^{\mathrm{esti}},
$$

which holds by the same arguments as those used to prove (4.17). We deduce that

$$
\mathbb{E}_M \left[\bar{V}_n^M(X_n)\right] \leq \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M \left[f\left(X_n, A(X_n)\right) + \hat{V}_{n+1}^M \left(X_{n+1}^A\right) \right] + \varepsilon_{\mathrm{HN},n}^{\mathrm{approx}} + 2\varepsilon_{\mathrm{HN},n}^{\mathrm{esti}}
$$
$$
\leq \mathbb{E}_M \left[\inf_{a \in A} \left\{ f\left(X_n, a\right) + \mathbb{E}_M^a \left[\hat{V}_{n+1}^M\left(X_{n+1}\right) \big| X_n\right] \right\} \right] + \varepsilon_{\mathrm{HN},n}^{\mathrm{approx}} + 2\varepsilon_{\mathrm{HN},n}^{\mathrm{esti}}.
$$

This completes the proof of the second inequality stated in (4.25). On the other hand, noting $\left| \bar{V}_n^M(X_n) - \inf_{a \in A} \left\{ f(X_n, a) + \mathbb{E}_M^a \left[\hat{V}_{n+1}^M(X_{n+1}) \big| X_n\right] \right\} \right| \leq 2\left((N - n)\|f\|_\infty + \|g\|_\infty\right)$ and applying (4.25), we obtain the inequality (4.26).

Step 2. We state the following estimation: for all $n \in \{0, ..., N\}$

$$\left\| \hat{V}_n^M(X_n) - \bar{V}_n^M(X_n) \right\|_{M,1} = \mathcal{O}_{\mathbb{P}} \left(\gamma_M^2 \sqrt{K_M \frac{log(M)}{M}} + \inf_{\Phi \in \mathcal{V}_M} \sqrt{\| \Phi(X_n) - V_n^M(X_n) \|_{M,1}} \right.$$
$$+ \inf_{A \in \mathbb{A}^{\mathcal{X}}} \sqrt{\left\| A(X_n) - a_n^{\text{opt}}(X_n) \right\|_{M,1}}$$
$$\left. + \sqrt{\left\| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_{M,1}} \right),$$
(4.27)

where $\|.\|_{M,p}$ stands for the $\mathbb{L}^p$ norm conditioned by the training set, i.e. $\|.\|_{M,p} = \left(\mathbb{E}_M \left[|.|^p \right] \right)^{\frac{1}{p}}$, for $p \in \{1, 2\}$. The proof relies on Lemma A.1 and Lemma A.2 (see Section A.8 in Appendix) which are proved respectively in [18] (see their Theorem 3) and [19].

Let us first show the following relation:

$$\mathbb{E}_M \left[|\hat{V}_n^M(X_n) - \bar{V}_n^M(X_n)|^2 \right] = \mathcal{O}_{\mathbb{P}} \left(\gamma_M^4 K_M \frac{log(M)}{M} + \inf_{\Phi \in \mathcal{V}_M} \mathbb{E} \left[|\Phi(X_n) - \bar{V}_n^M(X_n)|^2 \right] \right).$$
(4.28)

For this, take $\delta_M = \gamma_M^4 K_M \frac{\log(M)}{M}$, and let $\delta > \delta_M$. Apply Lemma A.2 to obtain:

$$\int_{c_2 \delta / \gamma_M^2}^{\sqrt{\delta}} \log \left(\mathcal{N}_2 \left(\frac{u}{4\gamma_M}, \left\{ f - g : f \in \mathcal{V}_M, \frac{1}{M} \sum_{m=1}^M |f(x_m) - g(x_m)|^2 \leq \frac{\delta}{\gamma_M^2} \right\}, x_1^M \right) \right)^{1/2} du$$
$$\leq \int_{c_2 \delta / \gamma_M^2}^{\sqrt{\delta}} \log \left(\mathcal{N}_2 \left(\frac{u}{4\gamma_M}, \mathcal{V}_M, x_1^M \right) \right)^{1/2} du$$
$$\leq \int_{c_2 \delta / \gamma_M^2}^{\sqrt{\delta}} ((4d+9)K_M + 1)^{1/2} \left[\log \left(\frac{48e\gamma_M^2 (K_M + 1)}{u} \right) \right]^{1/2} du$$
$$\leq \int_{c_2 \delta / \gamma_M^2}^{\sqrt{\delta}} ((4d+9)K_M + 1)^{1/2} \left[\log \left(48e\frac{\gamma_M^4}{\delta} (K_M + 1) \right) \right]^{1/2} du$$
$$\leq \sqrt{\delta}((4d+9)K_M + 1)^{1/2} \left[\log \left(48e\gamma_M^4 M (K_M + 1) \right) \right]^{1/2}$$
$$\leq c_5 \sqrt{\delta} \sqrt{K_M} \sqrt{\log(M)},$$
(4.29)

where $\mathcal{N}_2(\varepsilon, \mathcal{V}, x_1^M)$ stands for the ε-covering number of $\mathcal{V}$ on x_1^M, which is introduced in section A.8, and where the last line holds since we assumed $\frac{M\delta_M}{\gamma_M^2} \xrightarrow[M \to 0]{} 0$. Since $\delta > \delta_M := \gamma_M^4 K_M \frac{\log(M)}{M}$, we then have $\sqrt{\delta}\sqrt{K_M}\sqrt{\log(M)} \leq \frac{\sqrt{M}\delta}{\gamma_M^2}$, which implies that (A.33) holds by (4.29). Therefore, by application of Lemma A.1, it holds:

$$\mathbb{E}_M \left[|\tilde{V}_n^M(X_n) - \bar{V}_n^M(X_n)|^2 \right] = \mathcal{O}_{\mathbb{P}} \left(\gamma_M^4 K_M \frac{\log(M)}{M} + \inf_{\Phi \in \mathcal{V}_M} \mathbb{E} \left[|\Phi(X_n) - \bar{V}_n^M(X_n)|^2 \right] \right).$$

It remains to note that $\mathbb{E}_M \left[|\hat{V}_n^M(X_n) - \bar{V}_n^M(X_n)|^2 \right] \leq \mathbb{E}_M \left[|\tilde{V}_n^M(X_n) - \bar{V}_n^M(X_n)|^2 \right]$ always holds, and this completes the proof of (4.28).

28

Next, let us show

$$\inf_{\Phi \in \mathcal{V}_M} \left\| \Phi(X_n) - \bar{V}_n(X_n) \right\|_{M,2}$$

$$= \mathcal{O}\left(\gamma_M^2 \sqrt{\frac{K_M \log(M)}{M}} + \sup_{n \le k \le N} \inf_{\Phi \in \mathcal{V}_M} \left\| \Phi(X_n) - V_n(X_n) \right\|_{M,2} \right.$$

$$\left. + \inf_{A \in \mathcal{A}_M} \mathbb{E}_M \left[\left| A(X_n) - a_n^{\mathrm{opt}}(X_n) \right| \right] + \left\| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_{M,2} \right). (4.30)$$

For this, take some arbitrary $\Phi \in \mathcal{V}_M$ and split

$$\inf_{\Phi \in \mathcal{V}_M} \left\| \Phi(X_n) - \bar{V}_n^M(X_n) \right\|_{M,2} \le \inf_{\Phi \in \mathcal{V}_M} \left\| \Phi(X_n) - V_n(X_n) \right\|_{M,2} + \left\| V_n(X_n) - \bar{V}_n^M(X_n) \right\|_{M,2}.$$
$$(4.31)$$

To bound the last term in the r.h.s. of (4.31), we write

$$\left\| V_n(X_n) - \bar{V}_n^M(X_n) \right\|_{M,2} \le \left\| V_n(X_n) - \inf_{a \in A} \left\{ f(X_n, a) + \mathbb{E}_M^a \left[\hat{V}_{n+1}^M(X_{n+1}) \,\middle|\, X_n \right] \right\} \right\|_{M,2}$$

$$+ \left\| \inf_{a \in A} \left\{ f(X_n, a) + \mathbb{E}_M^a \left[\hat{V}_{n+1}^M(X_{n+1}) \,\middle|\, X_n \right] \right\} - \bar{V}_n^M(X_n) \right\|_{M,2}$$

Use the dynamic programming principle, assumption **(Hd)** and (4.26) to get:

$$\left\| V_n(X_n) - \bar{V}_n^M(X_n) \right\|_{M,2} \le \|r\|_\infty \left\| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_{M,2}$$

$$+ \sqrt{2 \left((N-n) \|f\|_\infty + \|g\|_\infty \right) \left(2\varepsilon_{\mathrm{HN},n}^{\mathrm{esti}} + \varepsilon_{\mathrm{HN},n}^{\mathrm{approx}} \right)}.$$

We then notice that

$$\left| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right|^2 \le 2\|r\|_\infty \left((N-n)\|f\|_\infty + \|g\|_\infty \right) \left| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right|$$

holds a.s., so that

$$\left\| V_n(X_n) - \bar{V}_n^M(X_n) \right\|_{M,2} \le \sqrt{2\|r\|_\infty \left((N-n)\|f\|_\infty + \|g\|_\infty \right) \left\| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_{M,1}}$$

$$+ \sqrt{2 \left((N-n)\|f\|_\infty + \|g\|_\infty \right) \left(2\varepsilon_{\mathrm{HN},n}^{\mathrm{esti}} + \varepsilon_{\mathrm{HN},n}^{\mathrm{approx}} \right)},$$

and use Lemma 4.4 to bound $\varepsilon_{\mathrm{HN},n}^{\mathrm{approx}}$. By plugging into (4.31), and using the estimations in Lemmas 4.3 and 4.4, we obtain the estimate (4.30). Together with (4.28), this proves the required estimate (4.27). By induction, we get as $M \to \infty$,

$$\mathbb{E}_M \left[\left| \hat{V}_n^M(X_n) - V_n(X_n) \right| \right] = \mathcal{O}_{\mathbb{P}} \left(\left(\gamma_M^4 \frac{K_M \log(M)}{M} \right)^{\frac{1}{2(N-n)}} + \left(\gamma_M^4 \frac{\rho_M^2 \eta_M^2 \log(M)}{M} \right)^{\frac{1}{4(N-n)}} \right.$$

$$+ \sup_{n \le k \le N} \inf_{\Phi \in \mathcal{V}_M} \left(\mathbb{E}_M \left[|\Phi(X_k) - V_k(X_k)|^2 \right] \right)^{\frac{1}{2(N-n)}}$$

$$\left. + \sup_{n \le k \le N} \inf_{A \in \mathcal{A}_M} \left(\mathbb{E} \left[\left| A(X_k) - a_k^{\mathrm{opt}}(X_k) \right| \right] \right)^{\frac{1}{2(N-n)}} \right),$$

which completes the proof of Theorem 4.2. □

4.3   Hybrid-LaterQ algorithm

In this paragraph, we analyze the convergence of the Hybrid-LaterQ algorithm described in Section 3.2.2.

We shall make the following assumption on F to ensure the convergence of the Hybrid-LaterQ algorithm.

(HF-LQ) Assume F to be such that:

1. (Estimation error Assumption) **(HF)** holds, i.e. for all $e \in E$, there exists $C(e) > 0$ such that for all couples (x, a) and (x', a') in $\mathcal{X} \times \mathbb{A}$:

$$\left| F(x, a, \varepsilon) - F(x', a', \varepsilon) \right| \le C(\varepsilon) \left(|x - x'| + |a - a'| \right).$$

   Recall that for all integer $M > 0$, ρ_M is defined as

$$\rho_M = \mathbb{E}\Big[\sup_{1 \le m \le M} C(\varepsilon^m) \Big],$$

   where the $(\varepsilon^m)_m$ is a i.i.d. sample of the noise ε.

2. (Quantization Assumption) There exists a constant $[F]_L > 0$ such that for all $(x, a) \in \mathcal{X} \times \mathbb{A}$ and all pair of r.v. $(\varepsilon, \varepsilon')$, it holds:

$$\|F(x, a, \varepsilon) - F(x, a, \varepsilon')\|_2 \le [F]_L \|\varepsilon - \varepsilon'\|_2.$$

As for the hybrid-now algorithm, we shall consider neural networks for the value function estimation with one hidden layer, K neurons with total variation γ, kernel bounded by η, a sigmoid activation function σ for the hidden layer, and no activation function for the output layer (i.e. the last layer), which is represented by the parametric set of function $^\eta\mathcal{V}_K^\gamma$. Let η_M, K_M and γ_M be integers such that:

$$K_M \xrightarrow[M\to\infty]{} \infty \quad, \qquad \gamma_M \xrightarrow[M\to\infty]{} \infty \qquad, \quad \eta_M \xrightarrow[M\to\infty]{} \infty$$
$$\rho_M \eta_M \gamma_M^2 \sqrt{\frac{\log(M)}{M}} \xrightarrow[M\to\infty]{} 0. \tag{4.32}$$

In the sequel we denote by $\mathcal{V}_M := {}^{\eta_M}\mathcal{V}_{K_M}^{\gamma_M}$ the space of neural networks parametrized by the values η_M, γ_M and K_M that satisfy (4.32). We also consider the class $\mathcal{A}_M$ of neural networks for estimated feedback optimal control at time $n = 0, \ldots, N - 1$, as described in Section 4.1, with the same parameters η_M, γ_M and K_M.

Recall that the approximation of the value function and optimal policy in the Hybrid-LaterQ algorithm is computed in backward induction as follows:

- Initialize $\hat{V}_N^M = g$

- For $n = N-1, \ldots, 0$, generate a training sample $X_n^{(m)}$, $m = 1, \ldots, M$ from the training distribution μ, and a training sample for the exogenous noise $\varepsilon_{n+1}^{(m)}$, $m = 1, \ldots, M$.

(i) compute the approximated policy at time n

$$\hat{a}_n^M \quad \in \quad \underset{A \in \mathcal{A}_M}{\operatorname{argmin}} \frac{1}{M} \sum_{m=1}^{M} \left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{V}_{n+1}^M(X_{n+1}^{(m),A}) \right]$$

where $X_{n+1}^{(m),A} = F(X_n^{(m)}, A(X_n^{(m)}), \varepsilon_{n+1}^{(m)}) \rightsquigarrow P^{A(X_n^{(m)})}(X_n^{(m)}, dx')$.

(ii) compute an untruncated interpolation of the value function at time $n+1$

$$\tilde{V}_{n+1}^M \quad \in \quad \underset{\Phi \in \mathcal{V}_M}{\operatorname{argmin}} \frac{1}{M} \sum_{m=1}^{M} \left[\hat{V}_{n+1}^M(X_{n+1}^{(m),\hat{a}_n^M}) - \Phi\big(X_{n+1}^{(m),\hat{a}_n^M}\big) \right]^2, \qquad (4.33)$$

and set the truncated interpolation of the value function at time $n+1$

$$\tilde{V}_{n+1}^{\mathrm{trun}} \quad = \quad \max \Big(\min \big(\tilde{V}_{n+1}^M, \|V_{n+1}\|_\infty \big), -\|V_{n+1}\|_\infty \Big).$$

(iii) update/compute the estimated value function

$$\hat{V}_n^M(x) \quad = \quad f(x, \hat{a}_n^M(x)) + \sum_{\ell=1}^{L} p_\ell \tilde{V}_{n+1}^{\mathrm{trun}}\big(F(x, \hat{a}_n^M(x), e_\ell)\big),$$

where $\hat{\varepsilon}_n$ is a L-optimal quantizer of ε_n on the grid $\{e_1, \ldots, e_L\}$ with weights $(p_1, \ldots, p_L)$.

Remark 4.11 1. It is straightforward to see that the neuronal network functions in $\mathcal{V}_M$ are Lipschitz with Lipschitz coefficient bounded by $\eta_M \gamma_M$. We highly rely on this property to show the convergence of the Hybrid-LaterQ algorithm.

2. Note that (4.33) is an interpolation step. In the pseudo-code above, we decided to interpolate the value function $\tilde{V}_n^Q$ using neural networks in $\mathcal{V}_M$ by reducing an empirical quadratic norm. However, we could have chosen other families of functions and other loss criterion to minimize. Gaussian processes have been recently reconsidered to interpolate functions, see [25]. $\qquad \square$

We now state our main result about the convergence of the Hybrid-LaterQ algorithm.

Theorem 4.3 *Assume that there esxists an optimal feedback control $(a_k^{\mathrm{opt}})_{k=n,\ldots,N-1}$ for the control problem with value function V_n, $n = 0, \ldots, N$. Take $X_n \rightsquigarrow \mu$, and let L_M be a sequence of integers such that*

$$L_M \xrightarrow[M \to \infty]{} \infty, \quad and \quad \frac{\eta_M \gamma_M}{L_M^{1/d}} \xrightarrow[M \to \infty]{} 0.$$

Take L_M points for the optimal quantization of the exogenous noise. Then, it holds as $M \to \infty$:

$$\mathbb{E}_M\big[|\hat{V}_n^M(X_n) - V_n(X_n)|\big] = \mathcal{O}_{\mathbb{P}}\Bigg(\rho_M \eta_M \gamma_M^2 \sqrt{\frac{\log(M)}{M}} + \frac{\eta_M \gamma_M}{L_M^{1/d}}$$

$$+ \sup_{n \leq k \leq N} \inf_{A \in \mathcal{A}_M} \mathbb{E}\big[|A(X_k) - a_k^{\mathrm{opt}}(X_k)|\big]$$

$$+ \sup_{n+1 \leq k \leq N} \inf_{\Phi \in \mathcal{V}_M} \mathbb{E}\left[|\Phi(X_k) - V_k(X_k)|\right] \Bigg). \qquad (4.34)$$

31

Comment: Theorem 4.3, combined to Proposition 4.1, show that estimator $\hat{V}_n^M$ provided by Hybrid-LaterQ algorithm is consistent, i.e. converges in $\mathbb{L}^1$ toward the value function V_n at time n when the number of points for the regression and quantization goes to infinity.

Remark 4.12 Note that the dimension d of the state space appears (explicitly) in the quantization error written in (4.34), as well as (implicitly) in the approximation errors associated to the value functions and optimal control learning. See for example (4.9) for an explicit dependence of the approximation error on d. $\square$

In order to prove Theorem 4.3, let us introduce the estimation error at time n associated to the Hybrid-LaterQ algorithm by

$$\varepsilon_{\text{LQ},n}^{\text{esti}} := \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^{M} \left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] \right.$$

$$\left. - \mathbb{E}_{M,n,X_n}^A \left[f(X_n, A(X_n)) + \hat{V}_{n+1}^M(X_{n+1}) \right] \right|,$$

where $\hat{Y}_{n+1}^{(m),A} = \hat{V}_{n+1}^M(X_{n+1}^{(m),A})$, and $\mathbb{E}_{M,n,X_n}^A[.]$ stands for the expectation conditioned by the training set and X_n when decision A has been taken at time n.

We have the following bound on this estimation error:

Lemma 4.5 For $n = 0, \ldots, N - 1$, it holds:

$$\mathbb{E}\left[\varepsilon_{\text{LQ},n}^{\text{esti}}\right] \leq \frac{(\sqrt{2} + 16)\left((N-n)\|f\|_\infty + \|g\|_\infty\right) + 16[f]_L}{\sqrt{M}} + 16\frac{\rho_M \eta_M \gamma_M^2}{\sqrt{M}}$$

$$\underset{M \to \infty}{=} \mathcal{O}\left(\frac{\rho_M \eta_M \gamma_M^2}{\sqrt{M}}\right). \tag{4.35}$$

Moreover,

$$\mathbb{E}_M\left[\varepsilon_{\text{LQ},n}^{\text{esti}}\right] \underset{M \to \infty}{=} \mathcal{O}\left(\rho_M \eta_M \gamma_M^2 \sqrt{\frac{\log(M)}{M}}\right). \tag{4.36}$$

Remark 4.13 The result stated in Lemma 4.5 is the same as the one stated in Lemma 4.3 for the hybrid-now algorithm. This result can actually be proved using the same arguments, so we omit the proof here. $\square$

Next, we introduce the approximation error at time n associated to the Hybrid-LaterQ algorithm by

$$\varepsilon_{\text{LQ},n}^{\text{approx}} = \inf_{A \in \mathcal{A}_M} \mathbb{E}_M\left[f\left(X_n, A(X_n)\right) + \hat{Y}_{n+1}^A\right] - \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M\left[f\left(X_n, A(X_n)\right) + \hat{Y}_{n+1}^A\right],$$

where $\hat{Y}_{n+1}^A := \hat{V}_{n+1}^M\left(F\left(X_n, A(X_n), \varepsilon_{n+1}\right)\right)$.

We have the following bound on this approximation error, which is similar to the one stated in Lemma 4.4 for the Hybrid-Now algorithm. The proof is similar and is thus omitted here.

Lemma 4.6 *For $n = 0, ..., N - 1$, it holds:*

$$\varepsilon_{\text{LQ},n}^{\text{approx}} \leq ([f]_L + [r]_L \|V_{n+1}\|_\infty) \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M \left[|A(X_n) - a_n^{\text{opt}}(X_n)| \right]$$
$$+ 2\|r\|_\infty \mathbb{E}_M \left[\left| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right| \right]. \tag{4.37}$$

Proof of Theorem 4.3.

We split the L^1 norm as follows:

$$\left\| \hat{V}_n^M(X_n) - V_n(X_n) \right\|_{M,1} \leq \left\| \hat{V}_n^M - \bar{V}_n^M \right\|_{M,1} + \left\| \bar{V}_n^M - \bar{V}_n^{\text{opt}} \right\|_{M,1} \tag{4.38}$$
$$+ \left\| \bar{V}_n^{\text{opt}} - V_n \right\|_{M,1},$$

where $(\bar{V}_n^M)_n$ is defined as:

$$\begin{cases} \bar{V}_N^M(x) &= g(x) \\ \bar{V}_n^M(x) &= f(x, \hat{a}_n^M(x)) + \mathbb{E}_M \left[\tilde{V}_{n+1}^{\text{trunc}} \left(F(x, \hat{a}_n^M(x), \varepsilon_{n+1}) \right) \right], \quad n = 0, \ldots, N-1, \end{cases}$$

and $(\bar{V}_n^{\text{opt}})_n$ is defined as:

$$\begin{cases} \bar{V}_N^{\text{opt}}(x) &= g(x) \\ \bar{V}_n^{\text{opt}}(x) &= \inf_{a \in A} \left\{ f(x, a) + \mathbb{E}_M \left[\tilde{V}_{n+1}^{\text{trunc}} \left(F(x, a, \varepsilon_{n+1}) \right) \right] \right\}, \quad n = 0, \ldots, N-1. \end{cases}$$

Recall that $\|.\|_{M,p} = (\mathbb{E}_M[|.|^p])^{\frac{1}{p}}$ stands for the $\mathbb{L}^p$-norm conditioned by the training set, for $p \in \{1, 2\}$.

Step 1: The first term in the r.h.s. of (4.38) is the quantization error. We show that

$$\left\| \hat{V}_n^M - \bar{V}_n^M \right\|_{M,1} = \mathcal{O}_{\mathbb{P}} \left(\frac{\eta_M \gamma_M}{L_M^{1/d}} \right), \qquad \text{as } M \to \infty. \tag{4.39}$$

Denote by $\varepsilon_p^Q := \|\hat{V}_n^M(X_n) - \bar{V}_n^M(X_n)\|_p$ the $\mathbb{L}^p$-quantization error, for $p \in \{1, 2\}$. Since $\tilde{V}_n^{\text{trunc}}$ is Lipschitz, for $n \in \{0, ..., N\}$, with its Lipschitz coefficient bounded by $\eta_M \gamma_M$, we thus get:

$$\varepsilon_2^Q := \|\hat{V}_n^M(X_n) - \bar{V}_n^M(X_n)\|_2 \leq \eta_M \gamma_M [F]_L \|\hat{\varepsilon}_{n+1} - \varepsilon_{n+1}\|_2, \tag{4.40}$$

from assumption **(HF-LQ)**. Now, recall by Zador theorem about optimal quantization (see [10]) that there exists some positive constant C such that

$$\lim_{M \to +\infty} \left(L_M^{\frac{2}{d}} \|\hat{\varepsilon}_{n+1} - \varepsilon_{n+1}\|_2^2 \right) = C.$$

By using Zador theorem in (4.40) and with inequality $\varepsilon_1^Q \leq \varepsilon_2^Q$, we obtain the bound (4.39) for the quantization error.

Step 2: We show: as $M \to \infty$,

$$\left\| \tilde{V}_{n+1}^{\text{trunc}}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_{M,1} = \mathcal{O}_{\mathbb{P}} \left(\sqrt{\frac{\log(M)}{M}} + \inf_{\Phi \in \mathcal{V}_M} \|\Phi(X_{n+1}) - V_{n+1}(X_{n+1})\|_{M,1} \right.$$
$$\left. + \left\| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_2^2 \right). \tag{4.41}$$

Denote by

$$\hat{R}_{n+1}\left(\tilde{V}_{n+1}^{\text{trunc}}\right) = \frac{1}{M}\sum_{m=1}^{M}\left|\tilde{V}_{n+1}^{\text{trunc}}(X_{n+1}^{(m)}) - \hat{V}_{n+1}^{M}(X_{n+1}^{(m)})\right|^2$$

the empirical quadratic risk, and by

$$R_{n+1}\left(\tilde{V}_{n+1}^{\text{trunc}}\right) = \mathbb{E}_M\left[\left|\tilde{V}_{n+1}^{\text{trunc}}(X_{n+1}) - \hat{V}_{n+1}^{M}(X_{n+1})\right|^2\right]$$

its associated quadratic risk. From the central limit theorem, we have

$$\frac{\hat{R}_{n+1}\left(\tilde{V}_{n+1}^{\text{trunc}}\right) - R_{n+1}\left(\tilde{V}_{n+1}^{\text{trunc}}\right)}{\sigma_{M,n+1}\sqrt{M}} \xrightarrow[M\to\infty]{\mathcal{L}} \mathcal{N}(0,1)$$

where $\sigma_{M,n+1}$ is the standard variation conditioned by the training set, defined as

$$\sigma_{M,n+1}^2 = \text{Var}_M\left(\left|\tilde{V}_{n+1}^{\text{trunc}}(X_{n+1}) - \hat{V}_{n+1}^{M}(X_{n+1})\right|^2\right).$$

Use inequality $\tilde{V}_{n+1}^{\text{trunc}}(X_{n+1}) - \hat{V}_{n+1}^{M}(X_{n+1}) \leq (N-n)\|f\|_\infty + \|g\|_\infty$ to bound $\sigma_{M,n+1}$ by a constant that does not depend on M, and get

$$R_{n+1}\left(\tilde{V}_{n+1}^{\text{trunc}}\right) = \mathcal{O}_\mathbb{P}\left(\sqrt{\frac{\log(M)}{M}} + \frac{1}{M}\sum_{m=1}^{M}\left|\tilde{V}_{n+1}^{\text{trunc}}(X_{n+1}^{(m)}) - \hat{V}_{n+1}^{M}(X_{n+1}^{(m)})\right|^2\right),$$

which, after noticing that

$$\frac{1}{M}\sum_{m=1}^{M}\left|\tilde{V}_{n+1}^{\text{trunc}}(X_{n+1}^{(m)}) - \hat{V}_{n+1}^{M}(X_{n+1}^{(m)})\right|^2 \leq \frac{1}{M}\sum_{m=1}^{M}\left|\tilde{V}_{n+1}^{M}(X_{n+1}^{(m)}) - \hat{V}_{n+1}^{M}(X_{n+1}^{(m)})\right|^2,$$

implies:

$$R_{n+1}\left(\tilde{V}_{n+1}^{\text{trunc}}\right) = \mathcal{O}_\mathbb{P}\left(\sqrt{\frac{\log(M)}{M}} + \inf_{\Phi\in\mathcal{V}_M}\frac{1}{M}\sum_{m=1}^{M}\left|\Phi(X_{n+1}^{(m)}) - \hat{V}_{n+1}^{M}(X_{n+1}^{(m)})\right|^2\right). \quad (4.42)$$

Once again from the central limit theorem, we derive:

$$\inf_{\Phi\in\mathcal{V}_M}\frac{1}{M}\sum_{m=1}^{M}\left|\Phi(X_{n+1}^{(m)}) - \hat{V}_{n+1}^{M}(X_{n+1}^{(m)})\right|^2 = \mathcal{O}_\mathbb{P}\left(\sqrt{\frac{\log(M)}{M}} + \inf_{\Phi\in\mathcal{V}_M}\left\|\Phi(X_{n+1}) - \hat{V}_{n+1}^{M}(X_{n+1})\right\|_2^2\right).$$
$$(4.43)$$

Indeed, first write

$$\mathbb{P}\left(\inf_{\Phi\in\mathcal{V}_M}\frac{1}{M}\sum_{m=1}^{M}\left|\Phi(X_{n+1}^{(m)}) - \hat{V}_{n+1}^{M}(X_{n+1}^{(m)})\right|^2 \leq \sqrt{\frac{\log(M)}{M}} + \inf_{\Phi\in\mathcal{V}_M}\left\|\Phi(X_{n+1}) - \hat{V}_{n+1}^{M}(X_{n+1})\right\|_2^2\right)$$

$$\leq \mathbb{P}\left(\frac{1}{M}\sum_{m=1}^{M}\left|\Phi(X_{n+1}^{(m)}) - \hat{V}_{n+1}^{M}(X_{n+1}^{(m)})\right|^2 \leq \sqrt{\frac{\log(M)}{M}} + \inf_{\Phi\in\mathcal{V}_M}\left\|\Phi(X_{n+1}) - \hat{V}_{n+1}^{M}(X_{n+1})\right\|_2^2\right)$$

$$\text{for all } \Phi \in \mathcal{V}_M,$$

$$\leq \mathbb{P}\left(\frac{1}{M}\sum_{m=1}^{M}\left|\tilde{\Phi}(X_{n+1}^{(m)}) - \hat{V}_{n+1}^{M}(X_{n+1}^{(m)})\right|^2 \leq \sqrt{\frac{\log(M)}{M}} + \left\|\tilde{\Phi}(X_{n+1}) - \hat{V}_{n+1}^{M}(X_{n+1})\right\|_2^2\right),$$

34

where $\tilde{\Phi} = \underset{\Phi \in \mathcal{V}_M}{\operatorname{argmin}} \left\| \Phi(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_2^2$. Then apply the Central limit theorem to get (4.43).

Plugging (4.43) into (4.42) leads to

$$
R_{n+1}\left(\tilde{V}_{n+1}^{\text{trunc}}\right) = \mathcal{O}_{\mathbb{P}}\left(\sqrt{\frac{\log(M)}{M}} + \inf_{\Phi \in \mathcal{V}_M} \left\| \Phi(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_2^2 \right).
$$

Apply the triangular inequality to finally obtain:

$$
R_{n+1}\left(\tilde{V}_{n+1}^{\text{trunc}}\right) = \mathcal{O}_{\mathbb{P}}\left(\sqrt{\frac{\log(M)}{M}} + \inf_{\Phi \in \mathcal{V}_M} \left\| \Phi(X_{n+1}) - V_{n+1}(X_{n+1}) \right\|_2^2 \right.
$$
$$
\left. + \left\| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_2^2 \right).
$$

It remains to notice that

$$
\left\| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_2^2 \leq \left((N-n-1)\|f\|_\infty + \|g\|_\infty\right) \left\| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_{M,1},
$$

to obtain inequality (4.41).

Step 3: let us show

$$
\left\| \bar{V}_n^M - \bar{V}_n^{\text{opt}} \right\|_{M,1} = \mathcal{O}_{\mathbb{P}}\left(\rho_M \eta_M \gamma_M^2 \sqrt{\frac{\log(M)}{M}} + \inf_{A \in \mathcal{A}_M} \left\| A(X_n) - a_n^{\text{opt}}(X_n) \right\|_{M,1} \right.
$$
$$
\left. + \left\| \tilde{V}_{n+1}^{\text{trunc}}(X_n) - V_n(X_n) \right\|_{M,1} \right). \tag{4.44}
$$

Note that once again it holds

$$
\left\| \bar{V}_n^M - \bar{V}_n^{\text{opt}} \right\|_{M,1} \leq 2\varepsilon_n^{\text{esti}} + \varepsilon_n^{\text{approx}},
$$

which can be shown by similar arguments as those used to prove of inequality (4.25). It remains to bound the estimation and approximation errors by using estimations (4.36) and (4.37) to get (4.44).

Step 4: We show

$$
\left\| \bar{V}_n^{\text{opt}}(X_n) - V_n(X_n) \right\|_{M,1} \leq \|r\|_\infty \left\| \hat{V}_{n+1}^M(X_{n+1}) - V_{n+1}(X_{n+1}) \right\|_{M,1} \tag{4.45}
$$
$$
+ \|r\|_\infty \left\| \tilde{V}_{n+1}^{\text{trunc}}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_{M,1},
$$

where $X_{n+1} \sim \mu$. For this, denote by $(\bar{V}_n')_{0 \leq n \leq N}$ the following auxiliary process:

$$
\begin{cases}
\bar{V}_N'(x) &= g(x) \\
\bar{V}_n'(x) &= \inf_{a \in A} \left\{ f(x,a) + \mathbb{E}_M\left[\hat{V}_{n+1}^M\left(F(x,a,\varepsilon_{n+1})\right) \right] \right\}, \quad n = 0, \ldots, N-1,
\end{cases}
$$

and notice that we have under assumption **(Hd)**:

$$\left\|\bar{V}_n^{\mathrm{opt}}(X_n) - V_n(X_n)\right\|_{M,1} \leq \left\|\bar{V}_n^{\mathrm{opt}}(X_n) - \bar{V}_n'(X_n)\right\|_{M,1} + \left\|\bar{V}_n'(X_n) - V_n(X_n)\right\|_{M,1}$$

$$\leq \|r\|_\infty \left\|\hat{V}_{n+1}^M(X_{n+1}) - V_{n+1}(X_{n+1})\right\|_{M,1}$$

$$+ \|r\|_\infty \left\|\tilde{V}_{n+1}^{\mathrm{trunc}}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1})\right\|_{M,1},$$

as stated in (4.45).

Step 5 Conclusion: By plugging (4.39), (4.44) and (4.45) into (4.38), we derive the following bound for the l.h.s. of (4.38):

$$\left\|\hat{V}_n^M(X_n) - V_n(X_n)\right\|_{M,1} = \mathcal{O}_{\mathbb{P}}\left(\frac{\eta_M \gamma_M}{L_M^{1/d}} + \rho_M \eta_M \gamma_M^2 \sqrt{\frac{\log(M)}{M}}\right.$$

$$+ \inf_{\Phi \in \mathcal{V}_M} \|\Phi(X_{n+1}) - V_{n+1}(X_{n+1})\|_{M,1} + \inf_{A \in \mathcal{A}_M} \left\|A(X_n) - a_n^{\mathrm{opt}}(X_n)\right\|_{M,1}$$

$$\left. + \left\|\hat{V}_{n+1}^M(X_n) - V_{n+1}(X_{n+1})\right\|_{M,1}\right), \quad \text{as } M \to +\infty.$$

By induction, we get for $n = 0, \ldots, N-1$:

$$\left\|\hat{V}_n^M(X_n) - V_n(X_n)\right\|_{M,1} = \mathcal{O}_{\mathbb{P}}\left(\frac{\eta_M \gamma_M}{L_M^{1/d}} + \rho_M \eta_M \gamma_M^2 \sqrt{\frac{\log(M)}{M}}\right.$$

$$+ \sup_{n \leq k \leq N} \inf_{A \in \mathcal{A}_M} \left\|A(X_k) - a_k^{\mathrm{opt}}(X_k)\right\|_{M,1}$$

$$\left. + \sup_{n+1 \leq k \leq N} \inf_{\Phi \in \mathcal{V}_M} \|\Phi(X_k) - V_k(X_k)\|_{M,1}\right),$$

which is the result stated in Theorem 4.3. □

A Appendix

A.1 Localization

In this section, we show how to relax the boundedness condition on the state space by a localization argument.

Let $R > 0$. Consider the localized state space $\bar{B}_d^{\mathcal{X}}(0,R) := \mathcal{X} \cap \{\|x\|_d \leq R\}$, where $\|.\|_d$ is the Euclidean norm of $\mathbb{R}^d$. Let $(\bar{X}_n)_{0 \leq n \leq N}$ be the Markov chain defined by its transition probabilities as

$$\mathbb{P}\left(\bar{X}_{n+1} \in O \middle| \bar{X}_n = x, a\right) = \int_O r(x, a; y) d\pi_R \circ \mu(y), \quad \text{for } n = 0, \ldots, N-1,$$

for all Borelian O in $\bar{B}_d^{\mathcal{X}}(0,R)$, where π_R is the Euclidean projection of $\mathbb{R}^d$ on $\bar{B}_d^{\mathcal{X}}(0,R)$, and $\pi_R \circ \mu$ is the pushforward measure of μ. Notice that the transition probability of $\bar{X}$

admits the same density r, for which **(Hd)** holds, w.r.t. $\pi_R \circ \mu$.

Define $(\bar{V}_n^R)_n$ as the value function associated to the following stochastic control problem for $(\bar{X}_n)_{0 \leq n \leq N}$:

$$\begin{cases} \bar{V}_N^R(x) & = g(x), \\ \bar{V}_n^R(x) & = \inf_{\alpha \in \mathcal{C}} \mathbb{E}\left[\sum_{k=n}^{N-1} f\left(\bar{X}_k, \alpha_k\right) + g\left(\bar{X}_N\right)\right], \text{ for } n = 0, \dots, N-1, \end{cases} \qquad \text{(A.1)}$$

for $x \in \bar{B}_d^{\mathcal{X}}(0, R)$. By the dynamic programming principle, $(\bar{V}_n^R)_n$ is solution of the following Bellman backward equation:

$$\begin{cases} \bar{V}_N^R(x) = g(x) \\ \bar{V}_n^R(x) = \inf_{a \in \mathbb{A}} \left\{ f(x, a) + \mathbb{E}_n^a\left[\bar{V}_{n+1}^R\left(\pi_R\left(F(x, a, \varepsilon_{n+1})\right)\right)\right]\right\}, \qquad \forall x \in B_d^{\mathcal{X}}(0, R), \end{cases}$$

where, again, π_R is the Euclidean projection on $B_d^{\mathcal{X}}(0, R)$.

We shall assume two conditions on the measure μ.
(Hloc) μ is such that:

$$\mathbb{E}\big[|\pi_R(X) - X|\big] \xrightarrow[R \to \infty]{} 0 \quad \text{and} \quad \mathbb{P}\left(|X| > R\right) \xrightarrow[R \to \infty]{} 0, \quad \text{where } X \sim \mu.$$

Using the dominated convergence theorem, it is straightforward to see that **(Hloc)** holds if μ admits a moment of order 1.

Proposition A.1 *Let $X_n \sim \mu$. It holds:*

$$\mathbb{E}\left[\left|\bar{V}_n^R\left(\pi_R(X_n)\right) - V_n\left(X_n\right)\right|\right] \leq \|V\|_\infty \left([r]_L \mathbb{E}\left[|\pi_R(X_n) - X_n|\right] + 2\mathbb{P}\left(|X_n| > R\right)\right) \frac{1 - \|r\|_\infty^{N-n}}{1 - \|r\|_\infty}$$

$$+ [g]_L \|r\|_\infty^{N-n} \mathbb{E}\left[|\pi_R(X_n) - X_n|\right],$$

*where we denote $\|V\|_\infty = \sup_{0 \leq k \leq N} \|V_k\|_\infty$, and use the convention $\frac{1-x^p}{1-x} = p$ for $x = 0$ and $p > 1$. Consequently, for all $n = 0, ..., N$, we get under **(Hloc)**:*

$$\mathbb{E}\left[\left|\bar{V}_n^R\left(\pi_R(X_n)\right) - V_n\left(X_n\right)\right|\right] \xrightarrow[R \to \infty]{} 0, \quad \text{where } X_n \sim \mu.$$

Comment: Proposition A.1 states that if $\mathcal{X}$ is not bounded, the control problem (A.1) associated to a bounded controlled process $\bar{X}$ can be as close as desired, in $\mathbb{L}^1(\mu)$ sense, to the original control problem by taking R large enough. Moreover, as stated before, the transition probability of $\bar{X}$ admits the same density r as X w.r.t. the pushforward measure $\pi_R \circ \mu$.

Proof of Proposition A.1. Take $X_n \sim \mu$ and write:

$$\mathbb{E}\left[\left|\bar{V}_n^R(\pi_R(X_n)) - V_n(X_n)\right|\right] \leq \mathbb{E}\left[\left|\bar{V}_n^R(X_n) - V_n(X_n)\right|\mathbb{1}_{|X_n| \leq R}\right]$$

$$+ \mathbb{E}\left[\left|\bar{V}_n^R(\pi_R(X_n)) - V_n(X_n)\right|\mathbb{1}_{|X_n| \geq R}\right]. \qquad \text{(A.2)}$$

Let us first bound the first term in the r.h.s. of (A.2), by showing that, for $n = 0, \ldots, N-1$:

$$\mathbb{E}\left[\left|\bar{V}_n^R(X_n) - V_n(X_n)\right|\mathbb{1}_{|X_n|\leq R}\right] \leq \|r\|_\infty \mathbb{E}\left[\left|\bar{V}_{n+1}^R(\pi_R(X_{n+1})) - V_{n+1}(X_{n+1})\right|\right]$$
$$+ [r]_L\|V_{n+1}\|_\infty \mathbb{E}\left[|\pi_R(X_{n+1}) - X_{n+1}|\right], \quad \text{with } X_{n+1} \sim \mu. \tag{A.3}$$

Take $x \in \bar{B}_d(0, R)$ and notice that

$$\left|\bar{V}_n^R(x) - V_n(x)\right| \leq \inf_{a\in A}\left\{\int_A \left|\bar{V}_{n+1}^R(\pi_R(y)) - V_{n+1}(y)\right| r(x, a; \pi_R(y))\, d\mu(y)\right.$$
$$\left. + \int |V_{n+1}(y)| \left|r(x, a; \pi_R(y)) - r(x, a; y)\right| d\mu(y)\right\}$$
$$\leq \|r\|_\infty \mathbb{E}\left[\left|\bar{V}_{n+1}^R(\pi(X_{n+1})) - V_{n+1}(X_{n+1})\right|\right]$$
$$+ [r]_L\|V_{n+1}\|_\infty \mathbb{E}\left[|\pi_R(X_{n+1}) - X_{n+1}|\right], \quad \text{where } X_{n+1} \sim \mu.$$

It remains to inject this bound in the expectation to obtain (A.3).

To bound the second term in the r.h.s. of (A.2), notice that

$$\left|\bar{V}_n^R(\pi_R(X_n)) - V_n(X_n)\right| \leq 2\|V_n\|_\infty$$

holds a.s., which implies:

$$\mathbb{E}\left[\left|\bar{V}_n^R(\pi_R(X_n)) - V_n(X_n)\right|\mathbb{1}_{|X_n|\geq R}\right] \leq 2\|V_n\|_\infty \mathbb{P}\left(|X_n| > R\right). \tag{A.4}$$

Plugging (A.3) and (A.4) into (A.2) yields:

$$\mathbb{E}\left[\left|\bar{V}_n^R(\pi_R(X_n)) - V_n(X_n)\right|\right] \leq \|r\|_\infty \mathbb{E}\left[\left|\bar{V}_{n+1}^R(\pi_R(X_{n+1})) - V_{n+1}(X_{n+1})\right|\right]$$
$$+ [r]_L\|V_{n+1}\|_\infty \mathbb{E}\left[|\pi_R(X_{n+1}) - X_{n+1}|\right] + 2\|V_n\|_\infty \mathbb{P}\left(|X_n| > R\right),$$

with X_n and X_{n+1} i.i.d. following the law μ. The result stated in proposition A.1 then follows by induction. $\qquad\square$

A.2 Forward evaluation of the optimal controls in $\mathcal{A}_M$

We evaluate in this section the real performance of the best controls in $\mathcal{A}_M$.

Let $(a_n^{\mathcal{A}_M})_{n=0}^{N-1}$ be the sequence of optimal controls in the class of neural networks $\mathcal{A}_M$, and denote by $(J_n^{\mathcal{A}_M})_{0\leq n\leq N}$ the cost functional sequence associated to $(a_n^{\mathcal{A}_M})_{n=0}^{N-1}$ and characterized as solution of the Bellman equation:

$$\begin{cases} J_N^{\mathcal{A}_M}(x) = g(x) \\ J_n^{\mathcal{A}_M}(x) = \inf_{A\in\mathcal{A}_M}\left\{f(x, A(x)) + \mathbb{E}_{n, X_n}^A[J_{n+1}^{\mathcal{A}_M}(X_{n+1})]\right\}, \end{cases}$$

where $\mathbb{E}_{n, X_n}^A[\cdot]$ stands for the expectation conditioned by X_n and when the control A is applied at time n.

In this section, we are interested in comparing $J_n^{\mathcal{A}_M}$ to V_n. Note that $V_n(x) \leq J_n^{\mathcal{A}_M}(x)$ holds for all $x \in \mathcal{X}$, since $\mathcal{A}_M$ is included in the set of the Borelian functions of $\mathcal{X}$. We can actually show the following:

Proposition A.2 *Assume that there exists a sequence of optimal feedback controls $(a_n^{\mathrm{opt}})_{0 \leq n \leq N-1}$ for the control problem with value function V_n, $n = 0, \ldots, N$. Then it holds, as $M \to \infty$:*

$$\mathbb{E}\left[J_n^{\mathcal{A}_M}(X_n) - V_n(X_n)\right] = \mathcal{O}\left(\sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E}\left[|A(X_k) - a_k^{\mathrm{opt}}(X_k)|\right]\right). \tag{A.5}$$

Remark A.1 Notice that there is no estimation error term in (A.5), since the optimal strategies in $\mathcal{A}_M$ are defined as those minimizing the real cost functionals in $\mathcal{A}_M$, and not the empirical ones. □

Proof of Proposition A.2. Let $n \in \{0, ..., N-1\}$, and $X_n \sim \mu$. Take $A \in \mathcal{A}_M$, and denote $J_n^A(X_n) = f(x, A(x)) + \mathbb{E}_{n,X_n}^A[J_{n+1}^{\mathcal{A}_M}(X_{n+1})]$. Clearly, we have $J_n^{\mathcal{A}_M} = \min_{A \in \mathcal{A}_M} J_n^A$. Moreover:

$$
\begin{aligned}
\mathbb{E}\left[J_n^A(X_n) - V_n(X_n)\right] &\leq \mathbb{E}\left[|f(X_n, A(X_n)) - f(X_n, a_n^{\mathrm{opt}}(X_n))|\right] \\
&\quad + \mathbb{E}\left[|J_{n+1}^{\mathcal{A}_M}(F(X_n, A(X_n), \varepsilon_{n+1})) - V_{n+1}(F(X_n, a_n^{\mathrm{opt}}(X_n), \varepsilon_{n+1}))|\right] \\
&\leq [f]_L \mathbb{E}\left[|a_n^{\mathrm{opt}}(X_n) - A(X_n)|\right] \\
&\quad + \mathbb{E}\left[|V_{n+1}(F(X_n, A(X_n), \varepsilon_{n+1})) - V_{n+1}(F(X_n, a_n^{\mathrm{opt}}(X_n), \varepsilon_{n+1}))|\right] \\
&\quad + \mathbb{E}\left[|J_{n+1}^{\mathcal{A}_M}(F(X_n, A(X_n), \varepsilon_{n+1})) - V_{n+1}(F(X_n, A(X_n), \varepsilon_{n+1}))|\right].
\end{aligned} \tag{A.6}
$$

Applying assumption **(Hd)** to bound the last term in the r.h.s. of (A.6) yields

$$
\begin{aligned}
\mathbb{E}\left[J_n^A(X_n) - V_n(X_n)\right] &\leq \left([f]_L + \|V_{n+1}\|_\infty [r]_L\right) \mathbb{E}\left[|a_n^{\mathrm{opt}}(X_n) - A(X_n)|\right] \\
&\quad + \|r\|_\infty \mathbb{E}\left[|J_{n+1}^{\mathcal{A}_M}(X_{n+1}) - V_{n+1}(X_{n+1})|\right],
\end{aligned}
$$

which holds for all $A \in \mathcal{A}_M$, so that:

$$
\begin{aligned}
\mathbb{E}\left[J_n^{\mathcal{A}_M}(X_n) - V_n(X_n)\right] &\leq \left([f]_L + \|V_{n+1}\|_\infty [r]_L\right) \inf_{A \in \mathcal{A}_M} \mathbb{E}\left[|a_n^{\mathrm{opt}}(X_n) - A(X_n)|\right] \\
&\quad + \|r\|_\infty \mathbb{E}\left[|J_{n+1}^{\mathcal{A}_M}(X_{n+1}) - V_{n+1}(X_{n+1})|\right].
\end{aligned}
$$

(A.5) then follows directly by induction. □

A.3 Proof of Lemma 4.1

The proof is divided into four steps.

Step 1: Symmetrization by a ghost sample. We take $\varepsilon > 0$ and show that for $M > 2\frac{\left((N-n)\|f\|_\infty + \|g\|_\infty\right)^2}{\varepsilon^2}$, it holds:

$$
\begin{aligned}
&\mathbb{P}\left[\sup_{A \in \mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^M \left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right] - \mathbb{E}\left[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right]\right| > \varepsilon\right] \\
&\leq 2\mathbb{P}\left[\sup_{A \in \mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^M \left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} - f(X_n'^{(m)}, A(X_n'^{(m)})) - \hat{Y}_{n+1}'^{(m),A}\right]\right| > \frac{\varepsilon}{2}\right],
\end{aligned} \tag{A.7}
$$

39

where:

- $(X_k^{'(m)})_{1\le m\le M, n\le k\le N}$ is a copy of $(X_k^{(m)})_{1\le m\le M, n\le k\le N}$ generated from an independent copy of the exogenous noises $(\varepsilon_k^{'(m)})_{1\le m\le M, n\le k\le N}$, and independent copy of initial positions $(X_n^{'(m)})_{1\le m\le M}$, following the same control $\hat{a}_k^M$ at time $k=n+1,\dots,N-1$, and control A at time n,

- We remind that $Y_{n+1}^{(m),A}$ has already been defined in (4.5), and we similarly define

$$Y_{n+1}^{'(m),A} := \sum_{k=n+1}^{N-1} f(X_k^{'(m),A}, \hat{a}_k^M(X_k^{'(m),A})) + g(X_N^{'(m),A}).$$

Let $A^* \in \mathcal{A}_M$ be such that:

$$\left| \frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A^*(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A^*}\right] - \mathbb{E}\left[J_n^{A^*,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right]\right| > \varepsilon$$

if such a function exists, and an arbitrary function in $\mathcal{A}_M$ if such a function does not exist. Note that $\frac{1}{M}\sum_{m=1}^M \left[f(X_n^{(m)}, A^*(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A^*}\right] - \mathbb{E}\left[J_n^{A^*,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right]$ is a r.v., which implies that A^* also depends on $\omega \in \Omega$. Denote by $\mathbb{P}_M$ the probability conditioned by the training set of exogenous noises $(\varepsilon_k^{(m)})_{1\le m\le M, n\le k\le N}$ and initial positions $(X_k^{(m)})_{1\le m\le M, n\le k\le N}$, and recall that $\mathbb{E}_M$ stands for the expectation conditioned by the latter. Application of Chebyshev's inequality yields

$$\mathbb{P}_M\left[\left|\mathbb{E}_M\left[J_n^{A^*,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n')\right] - \frac{1}{M}\sum_{m=1}^M\left[f(X_n^{'(m)}, A^*(X_n^{'(m)})) + \hat{Y}_{n+1}^{'(m),A^*}\right]\right| > \frac{\varepsilon}{2}\right]$$

$$\le \frac{\mathrm{Var}_M\left[J_n^{A^*,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n')\right]}{M(\varepsilon/2)^2}$$

$$\le \frac{\left((N-n)\|f\|_\infty + \|g\|_\infty\right)^2}{M\varepsilon^2},$$

where we have used $0 \le \left| J_n^{A^*,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n')\right| \le (N-n)\|f\|_\infty + \|g\|_\infty$ which implies

$$\mathrm{Var}_M\left[J_n^{A^*,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n')\right] = \mathrm{Var}_M\left[J_n^{A^*,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n') - \frac{(N-n)\|f\|_\infty + \|g\|_\infty}{2}\right]$$

$$\le \mathbb{E}\left[\left(J_n^{A^*,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n') - \frac{(N-n)\|f\|_\infty + \|g\|_\infty}{2}\right)^2\right]$$

$$\le \frac{\left((N-n)\|f\|_\infty + \|g\|_\infty\right)^2}{4}.$$

Thus, for $M > 2\frac{\left((N-n)\|f\|_\infty+\|g\|_\infty\right)^2}{\varepsilon^2}$, we have

$$\mathbb{P}_M\left[\left|\mathbb{E}_M\left[J_n^{A^*,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right] - \frac{1}{M}\sum_{m=1}^M\left[f(X_n^{'(m)}, A^*(X_n^{'(m)})) + \hat{Y}_{n+1}^{'(m),A^*}\right]\right| \le \frac{\varepsilon}{2}\right] \ge \frac{1}{2}.$$

$$(A.8)$$

Hence:

$$
\mathbb{P}\left[\sup_{A \in \mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} - f(X_n'^{(m)}, A(X_n'^m)) - \hat{Y}_{n+1}'^{(m),A}\right]\right| > \frac{\varepsilon}{2}\right]
$$

$$
\geq \mathbb{P}\left[\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A^*(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A^*} - f(X_n'^{(m)}, A^*(X_n'^{(m)})) - \hat{Y}_{n+1}'^{(m),A^*}\right]\right| > \frac{\varepsilon}{2}\right]
$$

$$
\geq \mathbb{P}\left[\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A^*(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A^*}\right] - \mathbb{E}_M\left[J_n^{A^*,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right]\right| > \varepsilon,\right.
$$

$$
\left.\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n'^{(m)}, A^*(X_n'^{(m)})) + \hat{Y}_{n+1}'^{(m),A^*}\right] - \mathbb{E}_M\left[J_n^{A^*,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right]\right| \leq \frac{\varepsilon}{2}\right].
$$

Observe that $\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A^*(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A^*}\right] - \mathbb{E}_M\left[J_n^{A^*,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right]$ is measurable w.r.t. the σ-algebra generated by the training set, so that conditioning by the training set and injecting (A.8) yields

$$
\mathbb{P}\left[\sup_{A \in \mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} - f(X_n'^{(m)}, A(X_n'^{(m)})) - \hat{Y}_{n+1}'^{(m),A}\right]\right| > \frac{\varepsilon}{2}\right]
$$

$$
\geq \frac{1}{2}\mathbb{P}\left[\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A^*(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A^*}\right] - \mathbb{E}_M\left[J_n^{A^*,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right]\right| > \varepsilon\right]
$$

$$
= \frac{1}{2}\mathbb{P}\left[\sup_{A \in \mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right] - \mathbb{E}\left[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right]\right| > \varepsilon\right],
$$

for $M > 2\frac{\left((N-n)\|f\|_\infty + \|g\|_\infty\right)^2}{\varepsilon^2}$, where we use the definition of A^* to go from the second-to-last to the last line. The proof of (A.7) is then completed.

Step 2: We show that

$$
\mathbb{E}\left[\sup_{A \in \mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right] - \mathbb{E}\left[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right]\right|\right]
$$

$$
\leq 4\mathbb{E}\left[\sup_{A \in \mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} - f(X_n'^{(m)}, A(X_n'^{(m)})) - \hat{Y}_{n+1}'^{(m),A}\right]\right|\right]
$$

$$
+ \mathcal{O}\left(\frac{1}{\sqrt{M}}\right). \tag{A.9}
$$

Indeed, let $M' = \sqrt{2}\frac{(N-n)\|f\|_\infty + \|g\|_\infty}{\sqrt{M}}$, and notice

$$\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right] - \mathbb{E}\left[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right]\right|\right]$$

$$= \int_0^\infty \mathbb{P}\left[\sup_{A\in\mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right] - \mathbb{E}\left[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right]\right| > \varepsilon\right]d\varepsilon$$

$$= \int_0^{M'} \mathbb{P}\left[\sup_{A\in\mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right] - \mathbb{E}\left[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right]\right| > \varepsilon\right]d\varepsilon$$

$$+ \int_{M'}^\infty \mathbb{P}\left[\sup_{A\in\mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right] - \mathbb{E}\left[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)\right]\right| > \varepsilon\right]d\varepsilon$$

$$\leq \sqrt{2}\frac{(N-n)\|f\|_\infty + \|g\|_\infty}{\sqrt{M}}$$

$$+ 4\int_0^\infty \mathbb{P}\left[\sup_{A\in\mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right.\right.\right.$$

$$\left.\left.\left. - f(X_n'^{(m)}, A(X_n'^{(m)})) - \hat{Y}_{n+1}'^{(m),A}\right]\right| > \varepsilon\right]d\varepsilon. \tag{A.10}$$

The second term in the r.h.s. of (A.10) comes from (A.7). It remains to write the latter as an expectation to obtain (A.9).

Step 3: Introduction of additional randomness by random signs.
Let $(r_m)_{1\leq m\leq M}$ be i.i.d. Rademacher r.v.[e]. We show that:

$$\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} - f(X_n'^{(m)}, A(X_n'^{(m)})) - \hat{Y}_{n+1}'^{(m),A}\right]\right|\right]$$

$$\leq 4\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}r_m\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right]\right|\right]. \tag{A.11}$$

Since for each $m = 1,...,M$ the set of exogenous noises $(\varepsilon_k'^{(m)})_{n\leq k\leq N}$ and $(\varepsilon_k^{(m)})_{n\leq k\leq N}$ are i.i.d., their joint distribution remain the same if one randomly interchanges the correspon-

---

[e]The probability mass function of a Rademacher r.v. is by definition $\frac{1}{2}\delta_{-1} + \frac{1}{2}\delta_1$.

ding components. Thus, it holds for $\varepsilon \geq 0$:

$$\mathbb{P}\left[\sup_{A \in \mathcal{A}_M} \left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} - f(X_n'^m, A(X_n'^m)) - \hat{Y}_{n+1}'^{(m),A}\right]\right| > \varepsilon\right]$$

$$= \mathbb{P}\left[\sup_{A \in \mathcal{A}_M} \left|\frac{1}{M}\sum_{m=1}^{M} r_m\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} - f(X_n'^m, A(X_n'^m)) - \hat{Y}_{n+1}'^{(m),A}\right]\right| > \varepsilon\right]$$

$$\leq \mathbb{P}\left[\sup_{A \in \mathcal{A}_M} \left|\frac{1}{M}\sum_{m=1}^{M} r_m\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right]\right| > \frac{\varepsilon}{2}\right]$$

$$+ \mathbb{P}\left[\sup_{A \in \mathcal{A}_M} \left|\frac{1}{M}\sum_{m=1}^{M} r_m\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right]\right| > \frac{\varepsilon}{2}\right]$$

$$\leq 2\mathbb{P}\left[\sup_{A \in \mathcal{A}_M} \left|\frac{1}{M}\sum_{m=1}^{M} r_m\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right]\right| > \frac{\varepsilon}{2}\right].$$

It remains to integrate on $\mathbb{R}_+$ w.r.t. ε to get (A.11).

Step 4: We show that

$$\mathbb{E}\left[\sup_{A \in \mathcal{A}_M} \left|\frac{1}{M}\sum_{m=1}^{M} r_m\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right]\right|\right]$$

$$\leq \frac{(N-n)\|f\|_\infty + \|g\|_\infty}{\sqrt{M}}$$

$$+ \left([f]_L + [f]_L \sum_{k=n+1}^{N-1} (1 + \eta_M \gamma_M)^{k-n}[F]_L^{k-n} + \eta_M^{N-n}\gamma_M^{N-n}[F]_L^{N-n}[g]_L\right)$$

$$= \mathcal{O}\left(\frac{\gamma_M^{N-n}\eta_M^{N-n}}{\sqrt{M}}\right), \quad \text{as } M \to \infty. \tag{A.12}$$

Adding and removing the cost obtained by control 0 at time n yields:

$$\mathbb{E}\left[\sup_{A \in \mathcal{A}_M} \left|\frac{1}{M}\sum_{m=1}^{M} r_m\left(f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right)\right|\right]$$

$$\leq \mathbb{E}\left[\left|\frac{1}{M}\sum_{m=1}^{M} r_m\left(f(X_n^{(m)}, 0) + \hat{Y}_{n+1}^{(m),0}\right)\right|\right]$$

$$+ \mathbb{E}\left[\sup_{A \in \mathcal{A}_M} \left|\frac{1}{M}\sum_{m=1}^{M} r_m\left(f(X_n^{(m)}, A(X_n^{(m)})) - f(X_n^{(m)}, 0) + \hat{Y}_{n+1}^{(m),A} - \hat{Y}_{n+1}^{(m),0}\right)\right|\right]. \tag{A.13}$$

We now bound the first term of the r.h.s. of (A.13). By Cauchy-Schwartz inequality, and recalling that $(r_m)_{1 \leq m \leq M}$ are i.i.d. with zero mean such that $r_m^2 = 1$, we get

$$\mathbb{E}\left[\left|\frac{1}{M}\sum_{m=1}^{M} r_m\left(f(X_n^{(m)}, 0) + \hat{Y}_{n+1}^{(m),0}\right)\right|\right] \leq \frac{1}{M}\sqrt{\mathbb{E}\left[\left|\sum_{m=1}^{M} r_m\left(f(X_n^{(m)}, 0) + \hat{Y}_{n+1}^{(m),0}\right)\right|^2\right]}$$

$$\leq \frac{1}{\sqrt{M}}\left((N-n)\|f\|_\infty + \|g\|_\infty\right) \tag{A.14}$$

Turn now to the second term of (A.13). By the Lipschitz continuity of f, it stands:

$$\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\sum_{m=1}^M r_m\Big(f(X_n^{(m)},A(X_n^{(m)}))-f(X_n^{(m)},0)+\hat{Y}_{n+1}^{(m),A}-\hat{Y}_{n+1}^{(m),0}\Big)\right|\right]$$

$$\leq [f]_L\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\sum_{m=1}^M r_m A(X_n^{(m)})\right|\right]+\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\sum_{m=1}^M r_m\Big(\hat{Y}_{n+1}^{(m),A}-\hat{Y}_{n+1}^{(m),0}\Big)\right|\right]$$

$$\leq \left([f]_L+[f]_L\sum_{k=n+1}^{N-1}\left(1+\eta_M\gamma_M\right)^{k-n}\mathbb{E}\left[\sup_{1\leq m\leq M}\prod_{j=n+1}^k C\left(\varepsilon_j^m\right)\right]\right.$$

$$\left.+[g]_L\left(1+\eta_M^{N-n}\gamma_M^{N-n}\right)\mathbb{E}\left[\sup_{1\leq m\leq M}\prod_{j=n+1}^N C\left(\varepsilon_j^m\right)\right]\right)\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\sum_{m=1}^M r_m A(X_n^{(m)})\right|\right]$$

$$(A.15)$$

where we condition by the exogenous noise, use assumption **(HF-PI)** and the $\eta_M\gamma_M$-Lipschitz continuity of the estimated optimal controls at time k, for $k=n+1,\ldots,N-1$.

Now, notice first that

$$\mathbb{E}\left[\sup_{1\leq m\leq M}\prod_{k=n+1}^N C\left(\varepsilon_k^m\right)\right]\leq\prod_{k=n+1}^N\mathbb{E}\left[\sup_{1\leq m\leq M}C\left(\varepsilon_k^m\right)\right]\leq\rho_M^{N-n},\qquad (A.16)$$

and moreover:

$$\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\sum_{m=1}^M r_m A(X_n^{(m)})\right|\right]\leq\gamma_M\mathbb{E}\left[\sup_{|v|_2\leq 1/R}\left|\sum_{m=1}^M r_m(v^T X_n^{(m)})_+\right|\right]$$

$$\leq\gamma_M\mathbb{E}\left[\sup_{|v|_2\leq 1/R}\left|\sum_{m=1}^M r_m v^T X_n^{(m)}\right|\right],\qquad (A.17)$$

where $R>0$ is a bound for the state space (see e.g. the discussion on the Frank-Wolfe step p.10 of [1] for a proof of this inequality), which implies by Cauchy-Schwarz inequality:

$$\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\sum_{m=1}^M r_m A(X_n^{(m)})\right|\right]\leq\frac{\gamma_M}{R}\sqrt{\mathbb{E}\left[\left|\sum_{m=1}^M r_m X_n^{(m)}\right|^2\right]}$$

$$\leq\gamma_M\sqrt{M}$$

since the $(r_m)_m$ are i.i.d. Rademacher r.v. Plug first (A.16) and (A.17) into (A.15) to obtain

$$\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\sum_{m=1}^M r_m\Big(f(X_n^{(m)},A(X_n^{(m)}))-f(X_n^{(m)},0)+\hat{Y}_{n+1}^{(m),A}-\hat{Y}_{n+1}^{(m),0}\Big)\right|\right]$$

$$\leq\left([f]_L+[f]_L\sum_{k=n+1}^{N-1}\left(1+\eta_M\gamma_M\right)^{k-n}\rho_M^{k-n}+[g]_L\left(1+\eta_M^{N-n}\gamma_M^{N-n}\right)\rho_M^{N-n}\right)\gamma_M\sqrt{M}.$$

$$(A.18)$$

44

Plug then (A.14) and (A.18) into (A.13) to get (A.12).

Step 5: Conclusion
Plug (A.12) into (A.11) and combine it with (A.9) to obtain the bound on the estimation error, as stated in (4.10) of Lemma 4.1. $\qquad\square$

A.4   Proof of Lemma 4.2

Let $(\hat{a}_k^M)_{k=n+1}^{N-1}$ be the sequence of estimated controls at time $k = n+1, ..., N-1$. Take $A \in \mathcal{A}_M$ and remind that we denote by $J_n^{A,(\hat{a})_{k=n+1}^{N-1}}$ the cost functional associated to the control A at time n, and $\hat{a}_k^M$ at time $k = n+1, \ldots, N-1$. The latter is characterized as solution of the Bellman equation

$$
\begin{cases}
J_N^{A,(\hat{a})_{k=n+1}^{N-1}}(x) = g(x) \\
J_n^{A,(\hat{a})_{k=n+1}^{N-1}}(x) = f(x, A(x)) + \mathbb{E}_{n,x}^A \left[J_{n+1}^{A,(\hat{a})_{k=n+1}^{N-1}}(X_{n+1}) \right],
\end{cases}
$$

where $\mathbb{E}_{n,x}^A[\cdot]$ stands for the expectation conditioned by $\{X_n = x\}$ when feedback control A is followed at time n.

Take $n \in \{1, ..., N\}$. It holds:

$$
\varepsilon_{\mathrm{PI},n}^{\mathrm{approx}} := \inf_{A \in \mathcal{A}_M} \mathbb{E}_M \left[J_n^{A,(\hat{a})_{k=n+1}^{N-1}}(X_n) \right] - \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M \left[J_n^{A,(\hat{a})_{k=n+1}^{N-1}}(X_n) \right]
$$

$$
= \inf_{A \in \mathcal{A}_M} \mathbb{E}_M \left[J_n^{A,(\hat{a})_{k=n+1}^{N-1}}(X_n) \right] - \mathbb{E}\left[V_n(X_n)\right] + \mathbb{E}\left[V_n(X_n)\right] - \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M \left[J_n^{A,(\hat{a})_{k=n+1}^{N-1}}(X_n) \right]
$$

$$
\leq \inf_{A \in \mathcal{A}_M} \mathbb{E}_M \left[J_n^{A,(\hat{a})_{k=n+1}^{N-1}}(X_n) \right] - \mathbb{E}\left[V_n(X_n)\right], \tag{A.19}
$$

where the last inequality stands because the value function is smaller than the cost functional associated to any other strategy. We then apply the dynamic programming principle to obtain:

$$
\min_{A \in \mathcal{A}_M} \mathbb{E}_M \left[J_n^{A,(\hat{a})_{k=n+1}^{N-1}}(X_n) \right] - \mathbb{E}\left[V_n(X_n)\right]
$$

$$
\leq \inf_{A \in \mathcal{A}_M} \mathbb{E}_M \left[f\big(X_n, A(X_n)\big) + \mathbb{E}_n^A \left[J_{n+1}^{(\hat{a})_{k=n+1}^{N-1}}(X_{n+1}) \right] \right]
$$

$$
- \mathbb{E} \left[f\big(X_n, a_n^{\mathrm{opt}}(X_n)\big) + \mathbb{E}_n^{a^{\mathrm{opt}}} \left[V_{n+1}(X_{n+1})\right] \right]. \tag{A.20}
$$

To bound the r.h.s. of (A.20), first observe that for $A \in \mathcal{A}_M$:

$$
\mathbb{E}_M \left[f\big(X_n, A(X_n)\big) + \mathbb{E}_n^A \left[J_{n+1}^{(\hat{a})_{k=n+1}^{N-1}}(X_{n+1}) \right] \right] - \mathbb{E}\left[f\big(X_n, a_n^{\mathrm{opt}}(X_n)\big) + \mathbb{E}_n^{a^{\mathrm{opt}}} \left[V_{n+1}(X_{n+1})\right] \right]
$$

$$
\leq \mathbb{E}\left[|f\big(X_n, A(X_n)\big) - f\big(X_n, a_n^{\mathrm{opt}}(X_n)\big)| \right] + \mathbb{E}_M \left[\mathbb{E}_n^A J_{n+1}^{(\hat{a})_{k=n+1}^{N-1}}(X_{n+1}) - \mathbb{E}_n^{a^{\mathrm{opt}}} V_{n+1}(X_{n+1}) \right]
$$

$$
\leq \big([f]_L + \|V_{n+1}\|_\infty [r]_L\big) \mathbb{E}\left[|A(X_n) - a_n^{\mathrm{opt}}(X_n)| \right] + \|r\|_\infty \mathbb{E}_M \left[J_{n+1}^{(\hat{a})_{k=n+1}^{N-1}}(X_{n+1}) - V_{n+1}(X_{n+1}) \right], \tag{A.21}
$$

where we used twice assumption **(Hd)** at the second-last line of (A.21). Inject inequality

$$\mathbb{E}_M\left[J_{n+1}^{(\hat{a})_{k=n+1}^{N-1}}(X_{n+1})\right] \leq \inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_{n+1}^{A,(\hat{a})_{k=n+2}^{N-1}}(X_{n+1})\right] + 2\varepsilon_{n+1}^{esti}$$

into (A.21) to obtain:

$$\mathbb{E}_M\left[f\big(X_n,A(X_n)\big) + \mathbb{E}_n^A\left[J_{n+1}^{(\hat{a})_{k=n+1}^{N-1}}(X_{n+1})\right]\right] - \mathbb{E}\left[f\big(X_n,a_n^{\mathrm{opt}}(X_n)\big) + \mathbb{E}_n^{a^{\mathrm{opt}}}\left[V_{n+1}(X_{n+1})\right]\right]$$

$$\leq \big([f]_L + \|V_{n+1}\|_\infty[r]_L\big)\mathbb{E}\left[\big|A(X_n) - a_n^{\mathrm{opt}}(X_n)\big|\right]$$

$$+ \|r\|_\infty \inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_{n+1}^{A,(\hat{a})_{k=n+2}^{N-1}}(X_{n+1}) - V_{n+1}(X_{n+1})\right] + 2\|r\|_\infty\varepsilon_{n+1}^{esti}. \text{ (A.22)}$$

Plugging (A.22) into (A.20) yields

$$\inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_n^{A,(\hat{a})_{k=n+1}^{N-1}}(X_n)\right] - \mathbb{E}\left[V_n(X_n)\right]$$

$$\leq \|r\|_\infty \inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_{n+1}^{A,(\hat{a})_{k=n+2}^{N-1}}(X_{n+1}) - V_{n+1}(X_{n+1})\right] + 2\|r\|_\infty\varepsilon_{n+1}^{esti}$$

$$+ \big([f]_L + \|V_{n+1}\|_\infty[r]_L\big)\inf_{A\in\mathcal{A}_M}\mathbb{E}\left[\big|A(X_n) - a_n^{\mathrm{opt}}(X_n)\big|\right],$$

which implies by induction, as $M \to +\infty$:

$$\mathbb{E}\left[\inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_n^{A,(\hat{a})_{k=n+1}^{N-1}}(X_n)\right] - \mathbb{E}\left[V_n(X_n)\right]\right]$$

$$= \mathcal{O}\left(\sup_{n+1\leq k\leq N-1}\mathbb{E}\left[\varepsilon_k^{esti}\right] + \sup_{n\leq k\leq N-1}\inf_{A\in\mathcal{A}_M}\mathbb{E}\left[\big|A(X_n) - a_n^{\mathrm{opt}}(X_n)\big|\right]\right).$$

We now use Lemma 4.1 to bound the expectations of the $\varepsilon_{\mathrm{PI},k}^{esti}$ for $k = n+1,\ldots,N-1$, and plug the result into (A.19) to complete the proof of Lemma 4.2. $\qquad\square$

A.5 Function approximation by neural networks

We assume $a_n^{\mathrm{opt}}(X_n) \in \mathbb{L}^2(\mu)$, and show the relation (4.8) in Proposition 4.1.
The universal approximator theorem applies for

$$\mathcal{A}_\infty := \bigcup_{M=1}^\infty \mathcal{A}_M,$$

and states that for all $\varepsilon > 0$, there exists a neural network a^* in $\mathcal{A}_\infty$ such that:

$$\sup_{n\leq k\leq N-1}\|a_k^{\mathrm{opt}} - a^*\|_\infty < \frac{\varepsilon}{\mathcal{V}_d(\mathcal{X})},$$

where $\mathcal{V}_d(\mathcal{X})$ stands for the volume of compact set $\mathcal{X}$ seen as a compact of the euclidean space $\mathbb{R}^d$. By integrating, we then get:

$$\sup_{n\leq k\leq N-1}\int_{\mathcal{X}}\big|a_k^{\mathrm{opt}}(x) - a^*(x)\big|d\mu(x) < \varepsilon,$$

Also, notice that $(\mathcal{A}_M)_{M \geq 1}$ is increasing, which implies that $\mathcal{A}_\infty = \lim_{M \to +\infty} \mathcal{A}_M$, and gives the existence of $M > 0$, that depends on ε, such that $a^* \in \mathcal{A}_M$.

Therefore, we have shown that for $n = 0, ..., N - 1$

$$\sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E}\big[|A(X_k) - a_k^{\mathrm{opt}}(X_k)|\big] \xrightarrow[M \to \infty]{} 0, \quad \text{with } X_k \sim \mu,$$

which is the required result stated in (4.8). $\qquad\square$

We now show (4.9) of proposition 4.1:

As stated in section 4.7 of [1]: proposition 6 in [1] shows that we can approximate a c-Lipschitz function by a γ_1-norm less than γ_M and uniform error less than $c \left(\frac{\gamma_M}{c}\right)^{-2d/(d+1)} \log \frac{\gamma_M}{c}$, and proposition 1 in [1] shows that a function with γ_1 less than γ_M may be approximated with K_M neurons with uniform error $\gamma_M K_M^{-(d+3)/(2d)}$.

Thus, given K_M and γ_M, there exists a neural network a^* in $\mathcal{V}_M$ such that

$$\|a^* - a^{\mathrm{opt}}\|_\infty \leq c \left(\frac{\gamma_M}{c}\right)^{-2d/(d+1)} \log\left(\frac{\gamma_M}{c}\right) + \gamma_M K_M^{-(d+3)/(2d)}. \qquad (A.23)$$

$\qquad\square$

A.6   Proof of Lemma 4.3

We prove Lemma 4.3 in four steps. Since the proof is very similar to the one of Lemma 4.1, we only detail the arguments that are modified.

Step 1: Symmetrization by a ghost sample. We take $\varepsilon > 0$ and show that for $M > 2\frac{\left((N-n)\|f\|_\infty + \|g\|_\infty\right)^2}{\varepsilon^2}$, it holds

$$\mathbb{P}\left[\sup_{A \in \mathcal{A}_M} \left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right] - \mathbb{E}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}\right]\right| > \varepsilon\right]$$

$$\leq 2\mathbb{P}\left[\sup_{A \in \mathcal{A}_M} \left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} - f(X_n'^{(m)}, A(X_n'^{(m)})) - \hat{Y}_{n+1}'^{(m),A}\right]\right| > \frac{\varepsilon}{2}\right],$$

$$(A.24)$$

where:

- $\left(X_n'^{(m)}\right)_{m=1}^{M}$ is a i.i.d. copy of $\left(X_n^{(m)}\right)_{m=1}^{M}$,

- $\left(\varepsilon_{n+1}'^{m}\right)_{m=1}^{M}$ is a i.i.d. copy of $\left(\varepsilon_{n+1}^{m}\right)_{m=1}^{M}$,

- we define
$$\hat{Y}_{n+1}^{(m),A} := \hat{V}_{n+1}^{M}\left(F\left(X_n^{(m)}, A\left(X_n^{(m)}\right), \varepsilon_{n+1}^{m}\right)\right),$$
and
$$\hat{Y}_{n+1}'^{(m),A} := \hat{V}_{n+1}^{M}\left(F\left(X_n'^{(m)}, A\left(X_n'^{(m)}\right), \varepsilon_{n+1}'^{m}\right)\right).$$

Proof: Since $\hat{V}_n^M$ the estimated value function at time n, for $n = 0, ..., N-1$, is bounded by construction (we truncated the estimation at the last step of the pseudo-code of the Hybrid algorithm), the proof is the same as the one in step 1 of Lemma 4.1. $\qquad\square$

Step 2: The following result holds

$$\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)},A(X_n^{(m)}))+\hat{Y}_{n+1}^{(m),A}\right]-\mathbb{E}\left[f(X_n^{(m)},A(X_n^{(m)}))+\hat{Y}_{n+1}^{(m),A}\right]\right|\right]$$

$$\leq 4\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)},A(X_n^{(m)}))+\hat{Y}_{n+1}^{(m),A}-f(X_n'^{(m)},A(X_n'^{(m)}))-\hat{Y}_{n+1}'^{(m),A}\right]\right|\right]$$

$$+\mathcal{O}\left(\frac{1}{\sqrt{M}}\right). \tag{A.25}$$

Proof: same as step 2 in the proof of Lemma 4.1. $\qquad\square$

Step 3: Introduction of additional randomness by random signs.
The following result holds:

$$\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}\left[f(X_n^{(m)},A(X_n^{(m)}))+\hat{Y}_{n+1}^{(m),A}-f(X_n'^{(m)},A(X_n'^{(m)}))-\hat{Y}_{n+1}'^{(m),A}\right]\right|\right]$$

$$\leq 4\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}r_m\left[f(X_n^{(m)},A(X_n^{(m)}))+\hat{Y}_{n+1}^{(m),A}\right]\right|\right]. \tag{A.26}$$

Proof: same as step 3 in the proof of Lemma 4.1. $\qquad\square$

Step 4: We show that

$$\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}r_m\left[f(X_n^{(m)},A(X_n^{(m)}))+\hat{Y}_{n+1}^{(m),A}\right]\right|\right]\leq\frac{(N-n)\|f\|_\infty+\|g\|_\infty}{\sqrt{M}}$$

$$+([f]_L+\rho_M\gamma_M\eta_M)\frac{\gamma_M}{\sqrt{M}}$$

$$=\mathcal{O}\left(\frac{\rho_M\gamma_M^2\eta_M}{\sqrt{M}}\right),\quad\text{as }M\to+\infty. \tag{A.27}$$

Adding and removing the cost obtained by control 0 at time n yields:

$$\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}r_m\left(f(X_n^{(m)},A(X_n^{(m)}))+\hat{Y}_{n+1}^{(m),A}\right)\right|\right]\leq\mathbb{E}\left[\left|\frac{1}{M}\sum_{m=1}^{M}r_m\left(f(X_n^{(m)},0)+\hat{Y}_{n+1}^{(m),0}\right)\right|\right]$$

$$+\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\frac{1}{M}\sum_{m=1}^{M}r_m\left(f(X_n^{(m)},A(X_n^{(m)}))-f(X_n^{(m)},0)+\hat{Y}_{n+1}^{(m),A}-\hat{Y}_{n+1}^{(m),0}\right)\right|\right]. \tag{A.28}$$

The first term in the r.h.s. in (A.28) is bounded as in the proof of Lemma 4.1 by

$$\frac{(N-n)\|f\|_\infty+\|g\|_\infty}{\sqrt{M}}.$$

We use the Lipschitz-continuity of f as follows, to bound its second term:

$$\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\sum_{m=1}^{M}r_m\left(f(X_n^{(m)},A(X_n^{(m)}))-f(X_n^{(m)},0)+\hat{Y}_{n+1}^{(m),A}-\hat{Y}_{n+1}^{(m),0}\right)\right|\right]$$

$$\leq[f]_L\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\sum_{m=1}^{M}r_mA(X_n^{(m)})\right|\right]+\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\sum_{m=1}^{M}r_m\left(\hat{Y}_{n+1}^{(m),A}-\hat{Y}_{n+1}^{(m),0}\right)\right|\right],$$

$$\leq([f]_L+\rho_M\eta_M\gamma_M)\,\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\sum_{m=1}^{M}r_mA(X_n^{(m)})\right|\right]$$

where we condition by the exogenous noise, use assumption **(HF)**, and the $\eta_M\gamma_M$-Lipschitz continuity of the estimated value fonction at time $n+1$.

By using the same arguments as those presented in the proof of Lemma 4.1, we can first bound $\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\sum_{m=1}^{M}r_mA(X_n^{(m)})\right|\right]$ as follows:

$$\mathbb{E}\left[\sup_{A\in\mathcal{A}_M}\left|\sum_{m=1}^{M}r_mA(X_n^{(m)})\right|\right]\leq\gamma_M\sqrt{M}, \tag{A.29}$$

and then conclude that (A.27) holds.

Step 5: Conclusion

Combining(A.25),(A.26) and (A.27) results in the bound on the estimation error as stated in (4.23). □

A.7   Proof of Lemma 4.4

We divide the proof of Lemma 4.4 into two steps.
First write

$$\varepsilon_{\mathrm{HN},n}^{\mathrm{approx}}\leq\inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[f\left(X_n,A(X_n)\right)+\hat{V}_{n+1}^{M}\left(X_{n+1}^A\right)\right]-\mathbb{E}\left[V_n(X_n)\right]$$

$$+\mathbb{E}\left[V_n(X_n)\right]-\inf_{A\in\mathbb{A}^{\mathcal{X}}}\mathbb{E}_M\left[f\left(X_n,A(X_n)\right)+\hat{V}_{n+1}^{M}\left(X_{n+1}^A\right)\right]. \tag{A.30}$$

Step 1: We show

$$\inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[f\left(X_n,A(X_n)\right)+\hat{V}_{n+1}^{M}\left(X_{n+1}^A\right)\right]-\mathbb{E}\left[V_n(X_n)\right]$$

$$\leq([f]_L+\|V_{n+1}\|_\infty[r]_L)\inf_{A\in\mathbb{A}^{\mathcal{X}}}\mathbb{E}_M\left[\left|A(X_n)-a_n^{\mathrm{opt}}(X_n)\right|\right] \tag{A.31}$$

$$+\|r\|_\infty\mathbb{E}_M\left[\left|V_{n+1}(X_{n+1})-\hat{V}_{n+1}^{M}(X_{n+1})\right|\right].$$

Take $A\in\mathcal{A}_M$, and apply the dynamic programming principle to write

$$\mathbb{E}_M\left[f\left(X_n,A(X_n)\right)+\hat{V}_{n+1}^{M}\left(X_{n+1}^A\right)\right]-\mathbb{E}\left[V_n(X_n)\right]$$

$$\leq[f]_L\mathbb{E}_M\left[\left|A(X_n)-a_n^{\mathrm{opt}}(X_n)\right|\right]+\mathbb{E}_M\left[\mathbb{E}_M\left[\hat{V}_{n+1}^{M}(X_{n+1}^A)\right]-\mathbb{E}_M\left[V_{n+1}\left(X_{n+1}^{a_n^{\mathrm{opt}}}\right)\right]\right]$$

$$\leq([f]_L+\|V_{n+1}\|_\infty[r]_L)\,\mathbb{E}_M\left[\left|A(X_n)-a_n^{\mathrm{opt}}(X_n)\right|\right]$$

$$+\mathbb{E}_M\left[\left|\hat{V}_{n+1}^{M}(X_{n+1}^A)-V_{n+1}(X_{n+1}^A)\right|\right],$$

49

where we used **(Hd)** at the second-to-last line. By using one more time assumption **(Hd)**, we then get:

$$
\mathbb{E}_M \left[f\left(X_n, A(X_n)\right) + \hat{V}_{n+1}^M \left(X_{n+1}^A\right) \right] - \mathbb{E}\left[V_n(X_n)\right]
$$
$$
\leq \left([f]_L + \|V_{n+1}\|_\infty [r]_L\right) \mathbb{E}_M \left[\left|A(X_n) - a_n^{\mathrm{opt}}(X_n)\right|\right]
$$
$$
+ \|r\|_\infty \mathbb{E}_M \left[\left|\hat{V}_{n+1}^M(X_{n+1}) - V_{n+1}(X_{n+1})\right|\right], \quad \text{with } X_{n+1} \sim \mu,
$$

which is the result stated in (A.31).

Step 2: We show

$$
\mathbb{E}\left[V_n(X_n)\right] - \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M \left[f\left(X_n, A(X_n)\right) + \hat{V}_{n+1}^M\left(X_{n+1}^A\right)\right]
$$
$$
\leq \|r\|_\infty \mathbb{E}_M \left[\left|V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1})\right|\right]. \tag{A.32}
$$

Write

$$
\mathbb{E}\left[V_n(X_n)\right] - \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M \left[f\left(X_n, A(X_n)\right) + \hat{V}_{n+1}^M\left(X_{n+1}^A\right)\right]
$$
$$
\leq \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M \left[f\left(X_n, A(X_n)\right) + V_{n+1}\left(X_{n+1}^A\right)\right] - \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M \left[f\left(X_n, A(X_n)\right) + \hat{V}_{n+1}^M\left(X_{n+1}^A\right)\right]
$$
$$
\leq \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M \left[V_{n+1}\left(X_{n+1}^A\right) - \hat{V}_{n+1}^M\left(X_{n+1}^A\right)\right]
$$
$$
\leq \|r\|_\infty \mathbb{E}_M \left[\left|V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1})\right|\right],
$$

which completes the proof of (A.32).

Step 3 Conclusion:
We complete the proof of Lemma 4.4 by plugging (A.31) and (A.32) into (A.30). □

A.8 Some useful Lemmas for the proof of Theorem 4.2

Fix $M \in \mathbb{N}^*$, let $x_1, \ldots, x_M \in \mathbb{R}^d$, and set $x^M = (x_1, \ldots, x_M)$. Define the distance $d_2(f, g)$ between $f : \mathbb{R}^d \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}$ by

$$
d_2(f, g) = \left(\frac{1}{M} \sum_{m=1}^M |f(x_m) - g(x_m)|^2\right)^{1/2}.
$$

An ε-cover of $\mathcal{V}$ (w.r.t. the distance d_2) is a set of functions $f_1, \ldots, f_P : \mathbb{R}^d \to \mathbb{R}$ such that

$$
\min_{p=1,\ldots,P} d_2\left(f, f_p\right) < \varepsilon, \quad \text{for } f \in \mathcal{V}.
$$

Let $\mathcal{N}_2(\varepsilon, \mathcal{V}, x^M)$ denote the size of the smallest ε-cover of $\mathcal{V}$ w.r.t. the distance d_2, and set $\mathcal{N}_2(\varepsilon, \mathcal{V}, x^M) = \infty$ if there does not exist any ε-cover of $\mathcal{V}$ of finite size. $\mathcal{N}_2(\varepsilon, \mathcal{V}, x^M)$ is called L^2-ε-covering number of $\mathcal{V}$ on x^M.

Lemma A.1 *Let (X, Y) be a random variable. Assume $|Y| \leq L$ a.s. and let*

$$m(x) = \mathbb{E}[Y|X = x].$$

Assume $Y - m(X)$ is sub-Gaussian in the sense that

$$\max_{m=1,\ldots,M} c^2 \mathbb{E}\left[e^{(Y-m(X))^2/c^2} - 1 \big| X \right] \leq \sigma^2 \quad a.s.$$

for some $c, \sigma > 0$. Let $\gamma_M, L \geq 1$ and assume that the regression function is bounded by L and that $\gamma_M \xrightarrow[M \to +\infty]{} +\infty$.
Set

$$\hat{m}_M = \underset{\Phi \in \mathcal{V}_M}{\operatorname{argmin}} \frac{1}{M} \sum_{m=1}^{M} \left| \Phi(x_i) - \bar{Y}_m \right|^2$$

for some $\mathcal{V}_M$ of functions $\Phi : \mathbb{R}^d \to [-\gamma_M, \gamma_M]$ and some random variables $\bar{Y}_1, \ldots, \bar{Y}_M$ which are bounded by L. Then there exists constants $c_1, c_2 > 0$ which depend only on σ and c such that for any $\delta_M > 0$ with

$$\delta_M \xrightarrow[M \to +\infty]{} 0, \quad and \quad \frac{M\delta_M}{\gamma_M} \xrightarrow[M \to +\infty]{} +\infty$$

and

$$c_1 \frac{\sqrt{M}\delta}{\gamma_M^2} \geq \int_{c_2\delta/\gamma_M^2}^{\sqrt{\delta}} \log \left(\mathcal{N}_2 \left(\frac{u}{4\gamma_M}, \left\{ f - g : f \in \mathcal{V}_M, \frac{1}{M} \sum_{m=1}^{M} \left| f(x_m) - g(x_m) \right|^2 \leq \frac{\delta}{\gamma_M^2} \right\}, x_1^M \right) \right)^{1/2} du \tag{A.33}$$

for all $\delta \geq \delta_M$ and all $g \in \mathcal{V}_M \cup \{m\}$ we have as $M \to +\infty$:

$$\mathbb{E}\left[\left| \bar{m}_M(X) - m(X) \right|^2 \right] = \mathcal{O}_{\mathbb{P}} \left(\frac{1}{M} \sum_{m=1}^{M} \left| Y_m - \bar{Y}_m \right|^2 + \delta_M + \inf_{\Phi \in \mathcal{V}_M} \mathbb{E}\left[\left| \Phi(X) - m(X) \right|^2 \right] \right).$$

Lemma A.2 *Let $\mathcal{V}_M$ be defined as in Section 4.2. For any $\varepsilon > 0$, we have*

$$\mathcal{N}_2 \left(\varepsilon, \mathcal{V}_M, (X_n^{(m)})_{1 \leq m \leq M} \right) \leq \left(\frac{12e\gamma_M (K_M + 1)}{\varepsilon} \right)^{(4d+9)K_M + 1}.$$

References

[1] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017.

[2] Achref Bachouch, Côme Huré, Nicolas Langrené, and Huyên Pham. Deep neural networks algorithms for stochastic control problems on finite horizon, part II: numerical applications. 2018.

[3] Alessandro Balata and Jan Palczewski. Regress-later Monte-Carlo for optimal inventory control with applications in energy. *arXiv:1703.06461*, 2018.

[4] Dimitri P. Bertsekas and John Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[5] George Cybenko. Approximations by superpositions of sigmoidal functions. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.

[6] Weinan E, Jiequn Han, and Arnulf Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics 5*, 5:349–380, 2017.

[7] Aurélien Géron. *Deep Learning avec TensorFlow*. O'Reilly Media, 2017.

[8] Paul Glasserman and Bin Yu. Simulation for American options: regression now or regression later? *Monte Carlo and Quasi-Monte Carlo Methods*, pages 213–226, 2004.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[10] Siegfried Graf and Harald Luschgy. *Foundations of Quantization for Probability Distributions*, volume 1730. Springer-Verlag Berlin Heidelberg, 2000.

[11] Julien Guyon and Pierre Henry-Labordere. Uncertain volatility model: a Monte-Carlo approach. *SSRN*, 2010.

[12] Lászlo Györfi, Michael Kohler, Adam Krzyzak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer Series in Statistics, 2002.

[13] Jiequn Han and Weinan E. Deep learning approximation for stochastic control problems. *arXiv:1611.07422*, 2016.

[14] Jiequn Han and Jihao Long. Convergence of the deep BSDE method for coupled FBSDEs. *arXiv:1811.01165v1*, 2018.

[15] Pierre Henry-Labordere. Deep primal-dual algorithm for BSDEs: Applications of machine learning to CVA and IM. *SSRN:3071506*, 2017.

[16] Kurt Hornick. Approximation capabilities of multilayer feedforward networks. *Neural Networks,*, 4:251–257, 1991.

[17] Idris Kharroubi, Nicolas Langrené, and Huyên Pham. A numerical algorithm for fully nonlinear HJB equations: an approach by control randomization. *Monte Carlo Methods and Applications*, 20(2):145–165, 2014.

[18] Michael Kohler. Nonparametric regression with additional measurement errors in the dependent variable. *Journal of Statistical Planning and Inference*, 136(10):3339–3361, October 2006.

[19] Michael Kohler, Adam Krzyżak, and Nebojsa Todorovic. Pricing of high-dimensional American options by neural networks. *Mathematical Finance*, 20(3):383–410, 2010.

[20] Andrey Nikolaevich Kolmogorov. On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables. *Mathematics and Its Applications (Soviet Series)*, 25, 1991.

[21] Steven Kou, Xianhua Peng, and Xingbo Xu. EM algorithm and stochastic control. Available at SSRN: https://ssrn.com/abstract=2865124, 2016.

[22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[23] Yuxi Li. Deep reinforcement learning: an overview. *arXiv 1701.07274v3*, 2017.

[24] Francis A. Longstaff and Eduardo S. Schwartz. Valuing American options by simulation: A simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147, 2001.

[25] Michael Ludkovski and Aditya Maheshwari. Simulation methods for stochastic storage problems: A statistical learning perspective. *arXiv:1803.11309*, 2018.

[26] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, and Andrei A. Rusu. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

[27] Michael Nielsen. Neural networks and deep learning.

[28] Gilles Pagès, Huyên Pham, and Jacques Printems. Optimal quantization methods and applications to numerical problems in finance. *Handbook of computational and numerical methods in finance*, pages 253–297, 2004.

[29] Warren B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality.* Wiley & Sons, 2011.

[30] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning.* The MIT Press, 1998.

[31] Ashia C. Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017.