



**HAL**  
open science

# Deep neural networks algorithms for stochastic control problems on finite horizon: convergence analysis

Côme Huré, Huyên Pham, Achref Bachouch, Nicolas Langrené

## ► To cite this version:

Côme Huré, Huyên Pham, Achref Bachouch, Nicolas Langrené. Deep neural networks algorithms for stochastic control problems on finite horizon: convergence analysis. *SIAM Journal on Numerical Analysis*, 2021, 59 (1), pp.525-557. 10.1137/20M1316640 . hal-01949213v2

**HAL Id: hal-01949213**

**<https://hal.science/hal-01949213v2>**

Submitted on 2 Dec 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# DEEP NEURAL NETWORKS ALGORITHMS FOR STOCHASTIC CONTROL PROBLEMS ON FINITE HORIZON: CONVERGENCE ANALYSIS

CÔME HURÉ\*, HUYÊN PHAM†, ACHREF BACHOUCH‡, AND NICOLAS LANGRENÉ§

**Abstract.** This paper develops algorithms for high-dimensional stochastic control problems based on deep learning and dynamic programming. Unlike classical approximate dynamic programming approaches, 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 dynamic programming recursion by performance or hybrid iteration, and regress now 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, and of the statistical error when estimating network function, leaving aside the optimization error. Numerical results on various applications are presented in a companion paper [3] and illustrate the performance of the proposed algorithms.

**Key words.** Deep learning, dynamic programming, performance iteration, regress now, convergence analysis, statistical risk.

**AMS subject classifications.** 65C05, 90C39, 93E35

**1. Introduction.** A large class of dynamic decision-making problems under uncertainty can be mathematically modeled as discrete-time stochastic optimal control problems in finite horizon. This paper is devoted to the analysis of novel probabilistic numerical algorithms based on neural networks for solving such problems. 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

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

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  $\sigma$ -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 with a control process  $\alpha$  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

---

\*LPSM, Université de Paris (Paris Diderot) ([hure at lpsm.paris](mailto:hure@lpsm.paris)).

†LPSM, Université de Paris (Paris Diderot) and CREST-ENSAE ([pham at lpsm.paris](mailto:pham@lpsm.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. ([achrefb at math.uio.no](mailto:achrefb@math.uio.no)). 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.

§CSIRO Data61, Melbourne, Australia ([nicolas.langrene at csiro.au](mailto:nicolas.langrene@csiro.au)).

processes  $\alpha$  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

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

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 with the controlled (homogenous) Markov chain (1.1), given by  $P^a(x, dx') := \mathbb{P}[F(x, a, \varepsilon_1) \in dx']$  and for any measurable function  $\varphi$  on  $\mathcal{X}$ :  $P^a\varphi(x) = \int \varphi(x')P^a(x, dx') = \mathbb{E}[\varphi(F(x, a, \varepsilon_1))]$ . With these notations, we have for any measurable function  $\varphi$  on  $\mathcal{X}$ , for any  $\alpha \in \mathcal{C}$ ,  $\mathbb{E}[\varphi(X_{n+1}^\alpha) | \mathcal{F}_n] = P^{\alpha_n}\varphi(X_n^\alpha)$ ,  $\forall n \in \mathbb{N}$ . The optimal value  $V_0(x_0)$  is then determined in backward induction starting from the terminal condition  $V_N(x) = g(x)$ ,  $x \in \mathcal{X}$ , and by the dynamic programming (DP) formula, for  $n = N - 1, \dots, 0$ :

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

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

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

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  $x \in \mathcal{X}$ , and any  $a \in \mathbb{A}$ , of  $P^a V_{n+1}(x)$ , for  $n = 0, \dots, N - 1$ . 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  $f(x, a) + P^a V_{n+1}(x)$  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.

The classical probabilistic numerical methods based on DP for solving the MDP are sometimes called approximate dynamic programming methods, see e.g. [6], [23], and consist basically of the two following steps:

- (i) Approximate the  $Q_n$ . 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 [21] 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 [17] for American option problem, and appears as a promising and efficient alternative in high dimension.
- (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:  $a_n(x) = A(x; \theta_n)$ ,  $n = 0, \dots, N - 1$ , for some given function  $A(\cdot, \theta_n)$  with parameters  $\theta = (\theta_0, \dots, \theta_{N-1}) \in \mathbb{R}^{q \times N}$ , and minimize over  $\theta$  the parametric functional  $\tilde{J}(\theta) = \mathbb{E} \left[ \sum_{n=0}^{N-1} f(X_n^A, A(x; \theta_n)) + g(X_N^A) \right]$ , where  $(X_n^A)_n$  denotes the controlled process with feedback control  $(A(\cdot, \theta_n))_n$ . This approach was first adopted in [19], who used EM algorithm for optimizing over the parameter  $\theta$ , and further investigated in [11], [8], [13], who considered deep neural networks (DNN) for the parametric feedback control, and stochastic gradient descent methods (SGD) for computing the optimal parameter  $\theta$ . The theoretical foundation of these DNN algorithms has been recently investigated in [12]. Let us mention that DNN approximation in stochastic control has already been explored in the context of reinforcement learning (RL) (see [6] and [24]), and called deep reinforcement learning in the artificial intelligence community [22] (see also [20] 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 [11] by DNN, but sequentially in time though DP instead of a global learning over the whole period  $0, \dots, N - 1$ . Once we get an approximation of the optimal policy, 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, rely on performance iteration (PI) or hybrid iteration (HI). Numerical results on several applications are devoted to a companion paper [3]. The theoretical contribution of the current paper is to provide

a detailed convergence analysis of our two proposed algorithms: Theorem 4.6 for the *NNContPI* algorithm based on control learning by performance iteration with DNN, and Theorem 4.13 for the *Hybrid-Now* algorithm based on control learning by DNN and then value function learning by regress-now method. We rely mainly on arguments from statistical learning and non parametric regression as developed notably in the book [10] to give estimates of approximated control and value function in terms of the universal approximation error of the neural networks, and of the statistical error in the estimation of network functions. In particular, the estimation of the optimal control using the control that minimizes the empirical loss is studied based on the uniform law of large numbers (see [10]); and the estimation of the value function in Hybrid-Now relies on Lemma H.1, proved in [17], which bounds the difference between the conditional expectation and the best approximation of the empirical conditional expectation by neural networks. In practical implementation, the computation of the approximate optimal policy in the minimization of the empirical loss function is performed by a stochastic gradient descent algorithm, which leads to the so-called *optimization error*, see e.g. [5], [15]. We do not address in the current paper this type of error in machine learning algorithms, and refer to the recent paper [4] for a full error analysis of deep neural networks training.

The paper is organized as follows. We recall in Section 2 some basic results about deep neural networks (DNN) and stochastic optimization gradient descent methods. Section 3 is devoted to the description of our two algorithms. We analyze in Section 4 the convergence of the algorithms. Finally the Appendices 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 nonlinear) 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 [18], [7] or [14]), 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 represented by functions  $x \in \mathcal{X} \mapsto \Phi(z; \theta) \in \mathbb{R}^o$ , 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.

**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

$$(2.1) \quad \inf_{\theta} \mathbb{E}[L(Z; \theta)],$$

where  $Z$  is a random variable from which the training samples  $Z^{(m)}$ ,  $m = 1, \dots, M$  are drawn, and  $L$  is a loss function involving DNN with parameters  $\theta \in \mathbb{R}^p$ , and

typically differentiable w.r.t.  $\theta$  with known gradient  $D_\theta L(Z^{(m)})$ .

Several algorithms based on stochastic gradient-descent are often used in practice for the search of infimum in (2.1); see e.g. basic SGD, AdaGrad, RMSProp, or Adam.

In the theoretical study of the algorithms proposed in this paper, we did not address the convergence of the gradient-descent algorithms, and rather assumed that the agent successfully solved the minimization problem (2.1), i.e. minimized the empirical loss  $\sum_{m=1}^M L(Z^{(m)})$  using (non stochastic) gradient-descent, a stochastic gradient-descent method with many epochs or any other methods that allow overfitting. One can then rely on statistical learning results as in [10] and [17] that are specific to neural networks. Notice that overfitting should be avoided in practice, and so our framework remains theoretical.

**3. Description of the algorithms.** 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  $\mu_n$  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, referred in short as *NNcontPI* algorithm, is designed as follows:

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

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

where  $X_n \sim \mu_n$ ,  $\hat{X}_{n+1}^\beta = F(X_n, A(X_n; \beta), \varepsilon_{n+1}) \sim P^{A(X_n; \beta)}(X_n, dx')$ , and for  $k = n + 1, \dots, N - 1$ ,  $\hat{X}_{k+1}^\beta = F(\hat{X}_k^\beta, \hat{a}_k(\hat{X}_k^\beta), \varepsilon_{k+1}) \sim P^{\hat{a}_k(\hat{X}_k^\beta)}(\hat{X}_k^\beta, dx')$ . Given estimate  $\hat{a}_k^M$  of  $\hat{a}_k$ ,  $k = n + 1, \dots, N - 1$ , the approximated policy  $\hat{a}_n$  is estimated by using a training sample  $(X_n^{(m)}, (\varepsilon_{k+1}^{(m)})_{k=n}^{N-1})$ ,  $m = 1, \dots, M$  of  $(X_n, (\varepsilon_{k+1})_{k=n}^{N-1})$  for simulating  $(X_n, (\hat{X}_{k+1}^\beta)_{k=n}^{N-1})$ , and optimizing  $\beta$  of  $A(\cdot; \beta) \in \mathcal{A}$ , by SGD (or its variants) as described in Section (2.2).

The estimated value function at time  $n$  is then given as

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

where  $\mathbb{E}_M$  is the expectation conditioned on the training set used for computing  $(\hat{a}_k^M)_k$ ; and  $(\hat{X}_k^{n,x})_{k=n}^N$  is driven by the estimated optimal controls. The dependence of the estimated value function  $\hat{V}_n^M$  upon the training samples  $X_k^{(m)}$ , for  $m = 1, \dots, M$ , used at time  $k = n, \dots, N$ , is emphasized through the exponent  $M$  in the notations.

*Remark 3.1.* The *NNcontPI* algorithm can be viewed as a combination of the DNN algorithm designed in [11] and dynamic programming. In the algorithm presented in [11], which totally ignores the dynamic programming principle, one learns all the optimal controls  $A(\cdot; \beta_n)$ ,  $n = 0, \dots, N - 1$  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  $A(\cdot; \beta_n)$ , for  $n = 0, \dots, N - 1$  is large (say  $\sum_{n=0}^{N-1} \ell_n \geq 100$ , where  $\ell_n$  is the number of layers in  $A(\cdot, \beta_n)$ ), then one is likely to observe vanishing or exploding gradient problems that will affect the training of the weights and bias of the first layers of the global NN (see [9] for more details).  $\square$

*Remark 3.2.* The *NNcontPI* algorithm 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 with 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.  $\square$

**3.2. Control learning by hybrid iteration.** Instead of keeping track of all the approximated optimal policies as in the *NNcontPI* algorithm, 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  $n$ . This leads to the following algorithm:

---

### Hybrid-Now algorithm

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

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

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

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

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

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

$$(3.4) \quad \hat{V}_n^M \in \operatorname{argmin}_{\Phi(\cdot; \theta) \in \mathcal{V}} \mathbb{E} \left| f(X_n, \hat{a}_n^M(X_n)) + \hat{V}_{n+1}^M(X_{n+1}^{\hat{a}_n^M}) - \Phi(X_n; \theta) \right|^2$$

using samples  $X_n^{(m)}$ ,  $X_{n+1}^{\hat{a}_n^M, (m)}$ ,  $m = 1, \dots, M$  of  $X_n \sim \mu_n$ , and  $X_{n+1}^{\hat{a}_n^M, (m)}$  of  $X_{n+1}^{\hat{a}_n^M}$

---

The approximated policy  $\hat{a}_n$  is estimated by using a training sample  $(X_n^{(m)}, \varepsilon_{n+1}^{(m)})$ ,  $m = 1, \dots, M$  of  $(X_n, \varepsilon_{n+1})$  to simulate  $(X_n, X_{n+1}^{A(\cdot; \beta)})$ , and optimizing  $A(\cdot; \beta) \in \mathcal{A}$  over the parameters  $\beta \in \mathbb{R}^l$  and the expectation in (3.3) by stochastic gradient descent

methods as described in Section (2.2). We then get an estimate  $\hat{a}_n^M = A(\cdot; \hat{\beta}_n^M)$ . The value function is represented by neural network and its optimal parameter is estimated by MC-regression in (3.4).

**3.3. Training sets design.** We discuss here the choice of the training measure  $\mu_n$  used to generate the training sets on which will be computed the estimates. 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. Separating the construction of the training set from the training procedure of the optimal control and/or value function simplifies the theoretical study of our algorithms. It is nevertheless common to merge them in practice, as discussed in Remark 3.3.

**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  $\Gamma_n$  constructed from distributions  $\mu_n$  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  $\mu_n$  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** 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,  $\Gamma_n$ , the training sets at time  $n$ , for  $n = 0, \dots, N - 1$ , can be built by forward simulation using optimal strategies estimated using algorithms developed e.g. in [11]. We are then back to the Exploitation only strategy framework. An alternative suggestion is to use the approximate policy obtained from a first iteration of the hybrid-now algorithm, in order to derive the associated approximated optimal state process,  $\hat{X}^{(1)}$ . In a second iteration of the hybrid-now algorithm, we then use the marginal distribution  $\mu_n^{(1)}$  of  $\hat{X}_n^{(1)}$ . We can even iterate this procedure in order to improve the choice of the training measure  $\mu_n$ , as proposed in [19].

*Remark 3.3.* For stationary control problems, which are the common framework in reinforcement learning, it is usual to combine exploration of the (state and action) space and learning of the value function by drawing the mini-batches under the law given by the current estimate of the  $Q$ -value plus some noise (commonly called  $\varepsilon$ -greedy policy). In practice, this method works well, and offers a flexible trade-off between exploration and exploitation. It looks however challenging to study its performance theoretically because it mixes the gradient-descent steps for the training of the value function and the optimal control. Also, note that this method is not easily adaptable to finite-horizon control problems.



**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  $\mu_n$  on  $\mathcal{X}$ , for  $n = 0, \dots, N$ , referred to as training distribution (see Section 3.3 for a discussion on the choice of  $\mu$ ). For sake of simplicity, we consider that  $\mu_n$  does not depend on  $n$ , and denote then by  $\mu$  the training distribution. We shall assume that the family of controlled transition probabilities has a density w.r.t.  $\mu$ , i.e.,

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

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.

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

*Remark 4.1.* Assumption **(Hd)** is usually satisfied when the state and control space are compact. 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 4.10), 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 Appendix A) by considering a training distribution  $\mu$  such that **(Hd)** is true and which admits a moment of order 1, i.e.  $\int |y|d\mu(y) < +\infty$ .  $\square$

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  $[g]_L$  s.t.

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

Under this boundedness condition, it is clear that the value function  $V_n$  is also bounded, and we have:  $\|V_n\|_\infty \leq (N - n)\|f\|_\infty + \|g\|_\infty$ , for  $n \in \{0, \dots, N\}$ .

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  $\mathcal{X} \times \mathbb{A}$ :

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

For  $M \in \mathbb{N}^*$ , let us define  $\rho_M = \mathbb{E} \left[ \sup_{1 \leq m \leq M} C(\varepsilon^m) \right]$ , where the  $(\varepsilon^m)_m$  is a i.i.d. sample of the noise  $\varepsilon$ . The rate of convergence of  $\rho_M$  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.  $F(x, a, \varepsilon) := b(x, a) + \sigma(x, a)\varepsilon$ , with  $b$  and  $\sigma$  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  $\sigma$ , and  $\|\cdot\|_d$  stands for the Euclidean norm in  $\mathbb{R}^d$ . Moreover, one can show that:

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

which implies in particular that  $\rho_M \underset{M \rightarrow +\infty}{=} \mathcal{O}\left(\sqrt{\log(M)}\right)$ . 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  $\epsilon_1^m$  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

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

where we used the closed-form expression of the moment generating function of the folded normal distribution<sup>a</sup> 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  $t = \sqrt{2 \log(2M')}$ :

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

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). □

*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  $n = 0, \dots, N$ , with a Lipschitz coefficient  $[V_n]_L$  that can be bounded by standard arguments as follows:

$$\begin{cases} [V_N]_L &= [g]_L \\ [V_n]_L &\leq \min\left([f]_L + \|V_n\|_\infty [r]_L, \rho_1 \frac{1 - \rho_1^{N-n}}{1 - \rho_1} + \rho_1^{N-n} [g]_L\right), \text{ for } n = 0, \dots, N-1, \end{cases}$$

using 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 algorithms described and studied in section 4.2. □

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

**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. We shall consider the following set of neural networks for approximating the optimal policy:

$$(4.3) \quad \mathcal{A}_K^\gamma := \left\{ x \in \mathcal{X} \mapsto A(x; \beta) = (A_1(x; \beta), \dots, A_q(x; \beta)) \in \mathbb{A}, \right. \\ \left. A_i(x; \beta) = \sigma_{\mathbb{A}} \left( \sum_{j=1}^K c_{ij} (a_{ij} \cdot x + b_{ij})_+ + c_{0j} \right), \quad i = 1, \dots, q, \right. \\ \left. \beta = (a_{ij}, b_{ij}, c_{ij})_{i,j}, \quad a_{ij} \in \mathbb{R}^d, \|a_{ij}\| \leq \eta, b_{ij}, c_{ij} \in \mathbb{R}, \sum_{i=0}^K |c_{ij}| \leq \gamma \right\},$$

where  $\|\cdot\|$  is the Euclidean norm in  $\mathbb{R}^d$ . Note that the considered neural networks have one hidden layer,  $K$  neurons, Relu activation functions, and  $\sigma_{\mathbb{A}}$  as output layer.  $\gamma$  and  $\eta$  are often referred to in the literature as respectively total variation and kernel. The activation function  $\sigma_{\mathbb{A}}$  is chosen depending on the form of the control space  $\mathbb{A}$ . For example, when  $\mathbb{A} = \mathbb{R}^q$ , we simply take  $\sigma_{\mathbb{A}}$  as the identity function; when  $\mathbb{A} = \mathbb{R}_+^q$ , or more generally in the form  $\mathbb{A} = \prod_{i=1}^q [a_i, \infty)$ , one can take the component-wise Relu activation function (possibly shifted and scaled); and when  $\mathbb{A} = [0, 1]^q$ , or more generally  $\mathbb{A} = \prod_{i=1}^q [a_i, b_i]$ , for  $a_i \leq b_i$ ,  $i = 1, \dots, q$ , one can take the component-wise sigmoid activation function (possibly shifted and scaled); when  $\mathbb{A}$  is a discrete space corresponding to discrete control values  $\{a_1, \dots, a_L\}$ , we may consider randomized control, hence valued in the simplex of  $\hat{\mathbb{A}}$  of  $[0, 1]^L$ , and use a softmax output layer activation function for parametrizing the discrete probabilities valued in  $\hat{\mathbb{A}}$  (such algorithm, called ClassiPI, has been developed in [3]; see also [9] on softmax output for classification problems, and [1] about softmax policies for action selection and possible improvements.)

Let  $K_M, \eta_M$  and  $\gamma_M$  be sequences of integers such that

$$(4.4) \quad K_M, \gamma_M, \eta_M \xrightarrow{M \rightarrow \infty} \infty \quad \text{and} \quad \rho_M^{N-1} \gamma_M^{N-1} \eta_M^{N-2} \sqrt{\frac{\log(M)}{M}} \xrightarrow{M \rightarrow \infty} 0.$$

We denote by  $\mathcal{A}_M := \eta_M \mathcal{A}_{k_M}^{\gamma_M}$  the class of neural network for policy as defined in (4.3) with parameters  $\eta_M, K_M$  and  $\gamma_M$  that satisfy the conditions (4.4).

*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 get:  $\rho_M^{N-n} \xrightarrow{M \rightarrow +\infty} \mathcal{O} \left( \sqrt{\log(M)}^{N-n} \right)$ .  $\square$

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

- Compute the approximated policy at time  $n$

$$(4.5) \quad \hat{a}_n^M \in \operatorname{argmin}_{A \in \mathcal{A}_M} \frac{1}{M} \sum_{m=1}^M \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m), A} \right]$$

where

$$(4.6) \quad \hat{Y}_{n+1}^{(m),A} = \sum_{k=n+1}^{N-1} f\left(X_k^{(m),A}, \hat{a}_k^M\left(X_k^{(m),A}\right)\right) + g\left(X_N^{(m),A}\right),$$

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

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

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

*Remark 4.5.* Finding the argmin in (4.5) is highly non-trivial and impossible in real situation. It is slowly approximated by (non stochastic) gradient-descent, i.e. with one large batch of size  $M$ , for which some convergence results are known under convexity or smoothness assumptions of  $F, f, g$ ; and is approximated in practice by stochastic gradient-descent with a large enough number of epochs. We refer to [15] for error analysis of stochastic gradient descent optimization algorithms. In order to simplify the theoretical analysis, we assume that the argmin in (4.5) is exactly reached. Our main focus here is to understand how the finite size of the training set affects the convergence of our algorithms.  $\square$

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

**THEOREM 4.6.** *Assume that there exists an optimal feedback control  $(a_k^{\text{opt}})_{k=n}^{N-1}$  for the control problem with value function  $V_n$ ,  $n = 0, \dots, N$ , and let  $X_n \sim \mu$ . Then, as  $M \rightarrow \infty$ <sup>b</sup>*

$$(4.7) \quad \mathbb{E}[\hat{V}_n^M(X_n) - V_n(X_n)] = \mathcal{O}\left(\frac{\rho_M^{N-n-1} \gamma_M^{N-n-1} \eta_M^{N-n-2}}{\sqrt{M}} + \sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E}[|A(X_k) - a_k^{\text{opt}}(X_k)|]\right),$$

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  $M \rightarrow \infty$ <sup>c</sup>

$$(4.8) \quad \mathbb{E}_M[\hat{V}_n^M(X_n) - V_n(X_n)] = \mathcal{O}_{\mathbb{P}}\left(\rho_M^{N-n-1} \gamma_M^{N-n-1} \eta_M^{N-n-2} \sqrt{\frac{\log(M)}{M}} + \sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E}[|A(X_k) - a_k^{\text{opt}}(X_k)|]\right),$$

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

<sup>b</sup>The notation  $x_M = \mathcal{O}(y_M)$  as  $M \rightarrow \infty$ , means that the ratio  $|x_M|/|y_M|$  is bounded as  $M$  goes to infinity.

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

*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 (4.5). We show in section B 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 does not suffer from this problem.  $\square$

**Comment:** Since we clearly have  $V_n \leq \hat{V}_n^M$ , estimation (4.7) implies the convergence in  $L^1$  norm of the NNcontPI algorithm, under condition (4.4), and in the case where

$$\sup_{n \leq k \leq N} \inf_{A \in \mathcal{A}_M} \mathbb{E}[|A(X_k) - a_k^{\text{opt}}(X_k)|] \xrightarrow{M \rightarrow +\infty} 0$$

. This is actually the case under some regularity assumptions on the optimal controls, as stated in the following proposition.

PROPOSITION 4.8. *The two following assertions hold:*

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

$$(4.9) \quad \sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E}[|A(X_k) - a_k^{\text{opt}}(X_k)|] \xrightarrow{M \rightarrow +\infty} 0.$$

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

$$(4.10) \quad \sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E}[|A(X_k) - a_k^{\text{opt}}(X_k)|] < c \left( \frac{\gamma_M}{c} \right)^{-2d/(d+1)} \log \left( \frac{\gamma_M}{c} \right) + \gamma_M K_M^{-(d+3)/(2d)}.$$

**Proof.** The first statement of Proposition 4.8 relies essentially on the universal approximation theorem, and the second assertion is stated and proved in [2]. For sake of completeness, we provide more details in Appendix E.  $\square$

*Remark 4.9.* The consistency of NNContPI is proved in Theorem 4.6, but no results on its variance are provided. We refer e.g. to Section 3.2 in [3] to observe the small variance of NNContPI for a linear quadratic problem in dimension up to 100.

The rest of this section is devoted to the proof of Theorem 4.6. 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  $n = 0, \dots, N-1$ , and given a feedback control (policy) represented by a sequence  $(A_k)_{k=n, \dots, N-1}$ , with  $A_k$  in  $\mathbb{A}^{\mathcal{X}}$ , we denote by  $J_n^{(A_k)_{k=n}^{N-1}}$  the cost functional associated with 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 with the NNContPI algorithm by

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

with  $X_n \sim \mu$ : 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 4.10. *For  $n = 0, \dots, N-1$ , the following holds:*

$$(4.11) \quad \begin{aligned} \mathbb{E}[\varepsilon_{\text{PI},n}^{\text{esti}}] &\leq (\sqrt{2} + 16) \frac{((N-n)\|f\|_\infty + \|g\|_\infty)}{\sqrt{M}} \\ &+ \frac{16\gamma_M}{\sqrt{M}} \left\{ [f]_L \left( 1 + \rho_M \frac{1 - \rho_M^{N-n-1} (1 + \eta_M \gamma_M)^{N-n-1}}{1 - \rho_M (1 + \eta_M \gamma_M)} \right) + (1 + \eta_M \gamma_M)^{N-n-1} \rho_M^{N-n} [g]_L \right\} \\ &= \mathcal{O} \left( \frac{\rho_M^{N-n-1} \gamma_M^{N-n-1} \eta_M^{N-n-2}}{\sqrt{M}} \right), \quad \text{as } M \rightarrow \infty. \end{aligned}$$

This implies in particular that

$$(4.12) \quad \varepsilon_{\text{PI},n}^{\text{esti}} = \mathcal{O}_{\mathbb{P}} \left( \rho_M^{N-n-1} \gamma_M^{N-n-1} \eta_M^{N-n-2} \sqrt{\frac{\log(M)}{M}} \right), \quad \text{as } M \rightarrow \infty,$$

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

**Proof.** The relation (4.11) states that the estimation error cancels when  $M \rightarrow \infty$  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 [10]. 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 to Section C in the Appendix. The proof of (4.12) follows from (4.11) by a direct application of the Markov inequality.  $\square$

Let us also define the *approximation error* at time  $n$  associated with the NNCon-  
tPI algorithm by

$$(4.13) \quad \varepsilon_{\text{PI},n}^{\text{approx}} := \inf_{A \in \mathcal{A}_M} \mathbb{E}_M \left[ J_n^{A, (\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n) \right] - \inf_{A \in \mathbb{A}^{\mathcal{X}}} \mathbb{E}_M \left[ J_n^{A, (\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n) \right],$$

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  $X_n \sim \mu$ .

$\varepsilon_{\text{PI},n}^{\text{approx}}$  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 4.11. *For  $n = 0, \dots, N-1$ , the following holds as  $M \rightarrow \infty$ ,*

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

This implies in particular

$$(4.15) \quad \begin{aligned} \varepsilon_{\text{PI},n}^{\text{approx}} &= \mathcal{O}_{\mathbb{P}} \left( \rho_M^{N-n-1} \gamma_M^{N-n-1} \eta_M^{N-n-2} \sqrt{\frac{\log(M)}{M}} \right. \\ &\left. + \sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E}[|A(X_k) - a_k^{\text{opt}}(X_k)|] \right). \end{aligned}$$

**Proof.** See Appendix D for the proof of (4.14). (4.15) is a direct application of the Markov inequality.  $\square$

**Proof of Theorem 4.6.**

*Step 1.* Let us denote by  $\hat{J}_{n,M}^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}} := \frac{1}{M} \sum_{m=1}^M [f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A}]$ , the empirical cost function, from time  $n$  to  $N$ , associated with the sequence of controls  $(A, (\hat{a}_k^M)_{k=n+1}^{N-1})$  and the training set, where we recall that  $\hat{Y}_{n+1}^{(m),A}$  is defined in (4.6). We then have

$$(4.16) \quad \begin{aligned} \mathbb{E}_M[\hat{V}_n^M(X_n)] &= \mathbb{E}_M[J_n^{(\hat{a}_k^M)_{k=n}^{N-1}}(X_n)] - \hat{J}_{n,M}^{(\hat{a}_k^M)_{k=n}^{N-1}} + \hat{J}_{n,M}^{(\hat{a}_k^M)_{k=n}^{N-1}} \\ &\leq \varepsilon_{\text{PI},n}^{\text{esti}} + \hat{J}_{n,M}^{(\hat{a}_k^M)_{k=n}^{N-1}}, \end{aligned}$$

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

$$(4.17) \quad \begin{aligned} \hat{J}_{n,M}^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}} &= \hat{J}_{n,M}^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}} - \mathbb{E}_M[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)] + \mathbb{E}_M[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)] \\ &\leq \varepsilon_{\text{PI},n}^{\text{esti}} + \mathbb{E}_M[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)]. \end{aligned}$$

Recalling that  $\hat{a}_n^M = \underset{A \in \mathcal{A}_M}{\operatorname{argmin}} \hat{J}_{n,M}^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}$ , and taking the infimum over  $\mathcal{A}_M$  in the l.h.s. of (4.17) first, and in the r.h.s. secondly, we then get

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

Plugging this last inequality into (4.16) yields the following estimate

$$(4.18) \quad \mathbb{E}_M[\hat{V}_n^M(X_n)] - \inf_{A \in \mathcal{A}_M} \mathbb{E}_M[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)] \leq 2\varepsilon_{\text{PI},n}^{\text{esti}}.$$

*Step 2.* By definition (4.13) 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  $a_n^{\text{opt}}$  at time  $n$ , we have

$$\begin{aligned} &\inf_{A \in \mathcal{A}_M} \mathbb{E}_M[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)] - \mathbb{E}_M[V_n(X_n)] \\ &= \varepsilon_{\text{PI},n}^{\text{approx}} + \inf_{A \in \mathbb{A}^x} \mathbb{E}_M\{f(X_n, A(X_n)) + \mathbb{E}_n^A[J_{n+1}^{(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_{n+1})]\} \\ &\quad - \mathbb{E}_M[f(X_n, a_n^{\text{opt}}(X_n)) + \mathbb{E}_n^{a_n^{\text{opt}}} [V_{n+1}(X_{n+1})]] \\ &\leq \varepsilon_{\text{PI},n}^{\text{approx}} + \mathbb{E}_M \mathbb{E}_n^{a_n^{\text{opt}}} [J_{n+1}^{(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_{n+1}) - V_{n+1}(X_{n+1})], \end{aligned}$$

where  $\mathbb{E}_n^A[\cdot]$  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

$$(4.19) \quad \begin{aligned} &\inf_{A \in \mathcal{A}_M} \mathbb{E}_M[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)] - \mathbb{E}_M[V_n(X_n)] \\ &\leq \varepsilon_{\text{PI},n}^{\text{approx}} + \|r\|_\infty \int [J_{n+1}^{(\hat{a}_k^M)_{k=n+1}^{N-1}}(x') - V_{n+1}(x')] \mu(dx') \\ &\leq \varepsilon_{\text{PI},n}^{\text{approx}} + \|r\|_\infty \mathbb{E}_M[\hat{V}_{n+1}^M(X_{n+1}) - V_{n+1}(X_{n+1})], \text{ with } X_{n+1} \sim \mu. \end{aligned}$$

Step 3. From (4.18) and (4.19), we have

$$\begin{aligned}
\mathbb{E}_M [\hat{V}_n^M(X_n) - V_n(X_n)] &= \mathbb{E}_M [\hat{V}_n^M(X_n)] - \inf_{A \in \mathcal{A}_M} \mathbb{E}_M [J_n^{A, (\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)] \\
&+ \inf_{A \in \mathcal{A}_M} \mathbb{E}_M [J_n^{A, (\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n)] - \mathbb{E}_M [V_n(X_n)] \\
(4.20) \quad &\leq 2\varepsilon_{\text{PI},n}^{\text{esti}} + \varepsilon_{\text{PI},n}^{\text{approx}} + \|r\|_\infty \mathbb{E}_M [\hat{V}_{n+1}^M(X_{n+1}) - V_{n+1}(X_{n+1})].
\end{aligned}$$

By induction, this implies

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

Use the estimations (4.12) for  $\varepsilon_{\text{PI},n}^{\text{esti}}$  in Lemma 4.10, and (4.15) for  $\varepsilon_{\text{PI},n}^{\text{approx}}$  in Lemma 4.11, and observe that  $\hat{V}_n(X_n) \geq V_n(X_n)$  holds a.s., to complete the proof of (4.8). Finally, the proof of (4.7) is obtained by taking expectation in (4.20), and using estimations (4.11) and (4.14).  $\square$

**4.2. Hybrid-Now algorithm.** In this paragraph, we analyze the convergence of the hybrid-now algorithm. We shall consider the following set of neural networks for the value function approximation:

$$\begin{aligned}
\eta \mathcal{V}_K^\gamma &:= \left\{ x \in \mathcal{X} \mapsto \Phi(x; \theta) = \sum_{i=1}^K c_i \sigma(a_i \cdot x + b_i) + c_0, \right. \\
&\left. \theta = (a_i, b_i, c_i)_i, \quad \|a_i\| \leq \eta, \quad b_i \in \mathbb{R}, \quad \sum_{i=0}^K |c_i| \leq \gamma \right\}.
\end{aligned}$$

Let  $\eta_M$ ,  $K_M$  and  $\gamma_M$  be integers such that:

$$(4.21) \quad \eta_M, \gamma_M, K_M \xrightarrow{M \rightarrow \infty} \infty \text{ s.t. } \frac{\gamma_M^4 K_M \log(M)}{M}, \frac{\gamma_M^4 \rho_M^2 \eta_M^2 \log(M)}{M} \xrightarrow{M \rightarrow \infty} 0,$$

with  $\rho_M$  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  $n = 0, \dots, N-1$ , parametrized by the values  $\eta_M$ ,  $\gamma_M$  and  $K_M$  that satisfy (4.21). We also consider the class  $\mathcal{A}_M$  of neural networks for estimated feedback optimal control at time  $n = 0, \dots, N-1$ , as described in Section 4.1, with the same parameters  $\eta_M$ ,  $\gamma_M$  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  $n = N-1, \dots, 0$ , generate a training sample  $X_n^{(m)}$ ,  $m = 1, \dots, M$  from the training distribution  $\mu$ , and a training sample for the exogenous noise  $\varepsilon_{n+1}^{(m)}$ ,  $m = 1, \dots, M$ .



(i) compute the approximated policy at time  $n$

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

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

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

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

and set the truncated estimated value function at time  $n$

$$(4.22) \quad \hat{V}_n^M = \max \left( \min(\tilde{V}_n^M, \|V_n\|_\infty), -\|V_n\|_\infty \right).$$

*Remark 4.12.* Notice that we have truncated the estimated value function in (4.22) 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.  $\square$

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

**THEOREM 4.13.** *Assume that there exists an optimal feedback control  $(a_k^{\text{opt}})_{k=n}^{N-1}$  for the control problem with value function  $V_n$ ,  $n = 0, \dots, N$ , and let  $X_n \sim \mu$ . Then, as  $M \rightarrow +\infty$*

$$(4.23) \quad \begin{aligned} & \mathbb{E}_M \left[ |\hat{V}_n^M(X_n) - V_n(X_n)| \right] \\ &= \mathcal{O}_{\mathbb{P}} \left( \left( \gamma_M^4 \frac{K_M \log(M)}{M} \right)^{\frac{1}{2(N-n)}} + \left( \gamma_M^4 \frac{\rho_M^2 \eta_M^2 \log(M)}{M} \right)^{\frac{1}{4(N-n)}} \right. \\ & \quad \left. + \sup_{n \leq k \leq N} \inf_{\Phi \in \mathcal{V}_M} \left( \mathbb{E}_M \left[ |\Phi(X_k) - V_k(X_k)|^2 \right] \right)^{\frac{1}{2(N-n)}} \right. \\ & \quad \left. + \sup_{n \leq k \leq N} \inf_{A \in \mathcal{A}_M} \left( \mathbb{E} \left[ |A(X_k) - a_k^{\text{opt}}(X_k)| \right] \right)^{\frac{1}{2(N-n)}} \right), \end{aligned}$$

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 4.14.* The consistency of Hybrid-Now is proved in Theorem 4.13, but no results on its variance are provided. The stability of Hybrid-Now has nevertheless been observed numerically in [3], and we refer e.g. to its Section 3.2 to observe the small variance of Hybrid-Now in a linear-quadratic problem in dimension up to 100.

**Comment:** Theorem 4.13 states that the estimator for the value function provided by the hybrid-now algorithm converges in  $\mathbb{L}^1(\mu)$  when the size of the training set goes to infinity. Note that  $\left( \gamma_M^4 \frac{K_M \log(M)}{M} \right)^{\frac{1}{2(N-n)}}$  and  $\left( \gamma_M^4 \frac{\rho_M^2 \eta_M^2 \log(M)}{M} \right)^{\frac{1}{4(N-n)}}$  stand for the estimation error made by estimating *empirically* respectively the value functions and

the optimal control by neural networks; also:  $\sup_{n \leq k \leq N} \inf_{\Phi \in \mathcal{V}_M} \sqrt{\mathbb{E} \left[ |\Phi(X_k) - V_k(X_k)|^2 \right]}$  and  $\sup_{n \leq k \leq N} \inf_{A \in \mathcal{A}_M} \mathbb{E} \left[ |A(X_k) - a_k^{\text{opt}}(X_k)| \right]$  are the approximation error made by estimating respectively the value function and the optimal control by neural networks.

In order to prove Theorem 4.13, let us first introduce the estimation error at time  $n$  associated with the Hybrid-Now algorithm by

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

where  $\hat{Y}_{n+1}^{(m),A} = \hat{V}_{n+1}^M(X_{n+1}^{(m),A})$ , and  $X_{n+1}^{(m),A} = F(X_n^{(m)}, A(X_n^{(m)}), \varepsilon_{n+1}^m)$ . We have the following bound on this estimation error:

LEMMA 4.15. *For  $n = 0, \dots, N-1$ , the following holds:*

$$\begin{aligned} \mathbb{E} [\varepsilon_{\text{HN},n}^{\text{esti}}] &\leq \frac{(\sqrt{2} + 16)((N-n)\|f\|_\infty + \|g\|_\infty) + 16[f]_L}{\sqrt{M}} + 16 \frac{\rho_M \eta_M \gamma_M^2}{\sqrt{M}} \\ (4.24) \quad &\stackrel{M \rightarrow \infty}{=} \mathcal{O} \left( \frac{\rho_M \eta_M \gamma_M^2}{\sqrt{M}} \right). \end{aligned}$$

**Proof.** See Appendix F.  $\square$

*Remark 4.16.* The result stated by lemma 4.15 is sharper than the one stated in Lemma 4.10 for the performance iteration procedure. The main reason is that we can make use of the  $\gamma_M \eta_M$ -Lipschitz-continuity of the estimate of the value function at time  $n+1$ .  $\square$

We secondly introduce the approximation error at time  $n$  associated with the Hybrid Now algorithm by

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

where  $\hat{Y}_{n+1}^A := \hat{V}_{n+1}^M(F(X_n, A(X_n), \varepsilon_{n+1}))$ . We have the following bound on this approximation error:

LEMMA 4.17. *For  $n = 0, \dots, N-1$ , the following holds:*

$$\begin{aligned} \varepsilon_{\text{HN},n}^{\text{approx}} &\leq ([f]_L + \|V_{n+1}\|_\infty [r]_L) \inf_{A \in \mathbb{A}^\mathcal{X}} \mathbb{E}_M \left[ |A(X_n) - a_n^{\text{opt}}(X_n)| \right] \\ (4.25) \quad &+ 2\|r\|_\infty \mathbb{E}_M \left[ \left| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right| \right]. \end{aligned}$$

**Proof.** See Appendix G.  $\square$

### Proof of Theorem 4.13

Observe that not only the optimal strategy but also the value function is estimated at each time step  $n = 0, \dots, N-1$  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  $n = 0, \dots, N-1$ ,  $\bar{V}_n^M$  is the quantity estimated by  $\hat{V}_n^M$ .

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

$$(4.26) \quad 0 \leq \mathbb{E}_M \left[ \bar{V}_n^M(X_n) - \inf_{a \in A} \left\{ f(X_n, a) + \mathbb{E}_{M,n,X_n}^a \left[ \hat{V}_{n+1}^M(X_{n+1}) \right] \right\} \right] \leq 2\varepsilon_{\text{HN},n}^{\text{esti}} + \varepsilon_{\text{HN},n}^{\text{approx}},$$

and,

$$(4.27) \quad \mathbb{E}_M \left[ \left| \bar{V}_n^M(X_n) - \inf_{a \in A} \left\{ f(X_n, a) + \mathbb{E}_{M,n,X_n}^a \left[ \hat{V}_{n+1}^M(X_{n+1}) \right] \right\} \right|^2 \right] \leq 2((N-n)\|f\|_\infty + \|g\|_\infty) \left( 2\varepsilon_{\text{HN},n}^{\text{esti}} + \varepsilon_{\text{HN},n}^{\text{approx}} \right),$$

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 (4.26). 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 (4.26). Moreover, we write

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

which holds by the same arguments as those used to prove (4.18). We deduce that

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

This completes the proof of the second inequality stated in (4.26).

On the other hand, noting:

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

and using (4.26), we obtain the inequality (4.27).

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

$$(4.28) \quad \left\| \hat{V}_n^M(X_n) - \bar{V}_n^M(X_n) \right\|_{M,1} \\ = \mathcal{O}_{\mathbb{P}} \left( \gamma_M^2 \sqrt{K_M \frac{\log(M)}{M}} + \inf_{\Phi \in \mathcal{V}_M} \sqrt{\|\Phi(X_n) - V_n^M(X_n)\|_{M,1}} \right. \\ \left. + \inf_{A \in \mathcal{A}^X} \sqrt{\|A(X_n) - a_n^{\text{opt}}(X_n)\|_{M,1}} + \sqrt{\|V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1})\|_{M,1}} \right),$$

where  $\|\cdot\|_{M,p} = \left( \mathbb{E}_M [|\cdot|^p] \right)^{\frac{1}{p}}$ , i.e.  $\|\cdot\|_{M,p}$  stands for the  $\mathbb{L}^p$  norm conditioned by the training set, for  $p \in \{1, 2\}$ . The proof relies on Lemma H.1 and Lemma H.2 (see Appendix H).

Let us first show the following relation:

$$(4.29) \quad \mathbb{E}_M \left[ |\hat{V}_n^M(X_n) - \bar{V}_n^M(X_n)|^2 \right] = \mathcal{O}_{\mathbb{P}} \left( \gamma_M^4 K_M \frac{\log(M)}{M} + \inf_{\Phi \in \mathcal{V}_M} \mathbb{E} \left[ |\Phi(X_n) - \bar{V}_n^M(X_n)|^2 \right] \right).$$

For this, take  $\delta_M = \gamma_M^4 K_M \frac{\log(M)}{M}$ , let  $\delta > \delta_M$ , and denote

$$\Omega_g := \left\{ f - g : f \in \mathcal{V}_M, \frac{1}{M} \sum_{m=1}^M |f(x_m) - g(x_m)|^2 \leq \frac{\delta}{\gamma_M^2} \right\}.$$

Apply Lemma H.2 to obtain:

$$(4.30) \quad \int_{c_2 \delta / \gamma_M^2}^{\sqrt{\delta}} \log \left( \mathcal{N}_2 \left( \frac{u}{4\gamma_M}, \Omega_g, x_1^M \right) \right)^{1/2} du \\ \leq \int_{c_2 \delta / \gamma_M^2}^{\sqrt{\delta}} \log \left( \mathcal{N}_2 \left( \frac{u}{4\gamma_M}, \mathcal{V}_M, x_1^M \right) \right)^{1/2} du \\ \leq \int_{c_2 \delta / \gamma_M^2}^{\sqrt{\delta}} ((4d+9)K_M + 1)^{1/2} \left[ \log \left( \frac{48e\gamma_M^2(K_M + 1)}{u} \right) \right]^{1/2} du \\ \leq \int_{c_2 \delta / \gamma_M^2}^{\sqrt{\delta}} ((4d+9)K_M + 1)^{1/2} \left[ \log \left( 48e \frac{\gamma_M^4}{\delta} (K_M + 1) \right) \right]^{1/2} du \\ \leq \sqrt{\delta} ((4d+9)K_M + 1)^{1/2} \left[ \log (48e\gamma_M^4 M (K_M + 1)) \right]^{1/2} \\ \leq c_5 \sqrt{\delta} \sqrt{K_M} \sqrt{\log(M)},$$

where  $\mathcal{N}_2(\varepsilon, \mathcal{V}, x_1^M)$  stands for the  $\varepsilon$ -covering number of  $\mathcal{V}$  on  $x_1^M$ , which is introduced in section H, and where the last line holds since we assumed  $\frac{M\delta_M}{\gamma_M^2} \xrightarrow{M \rightarrow 0} 0$ . 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 (H.1) holds by (4.30). Therefore, by application of Lemma H.1, the following 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 (4.29).

Next, let us show

$$\begin{aligned}
& \inf_{\Phi \in \mathcal{V}_M} \|\Phi(X_n) - \bar{V}_n(X_n)\|_{M,2} \\
&= \mathcal{O} \left( \gamma_M^2 \sqrt{\frac{K_M \log(M)}{M}} + \sup_{n \leq k \leq N} \inf_{\Phi \in \mathcal{V}_M} \|\Phi(X_n) - V_n(X_n)\|_{M,2} \right. \\
(4.31) & \left. + \inf_{A \in \mathcal{A}_M} \mathbb{E}_M [ |A(X_n) - a_n^{\text{opt}}(X_n)| ] + \left\| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_{M,2} \right).
\end{aligned}$$

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

$$(4.32) \quad \inf_{\Phi \in \mathcal{V}_M} \|\Phi(X_n) - \bar{V}_n^M(X_n)\|_{M,2} \leq \inf_{\Phi \in \mathcal{V}_M} \|\Phi(X_n) - V_n(X_n)\|_{M,2}$$

$$(4.33) \quad + \left\| V_n(X_n) - \bar{V}_n^M(X_n) \right\|_{M,2}.$$

To bound the last term in the r.h.s. of (4.32), we write

$$\begin{aligned}
& \left\| V_n(X_n) - \bar{V}_n^M(X_n) \right\|_{M,2} \\
& \leq \left\| V_n(X_n) - \inf_{a \in A} \left\{ f(X_n, a) + \mathbb{E}_M^a \left[ \hat{V}_{n+1}^M(X_{n+1}) | X_n \right] \right\} \right\|_{M,2} \\
& \quad + \left\| \inf_{a \in A} \left\{ f(X_n, a) + \mathbb{E}_M^a \left[ \hat{V}_{n+1}^M(X_{n+1}) | X_n \right] \right\} - \bar{V}_n^M(X_n) \right\|_{M,2}.
\end{aligned}$$

Use the dynamic programming principle, assumption **(Hd)** and (4.27) to get:

$$\begin{aligned}
\left\| V_n(X_n) - \bar{V}_n^M(X_n) \right\|_{M,2} & \leq \|r\|_\infty \left\| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_{M,2} \\
& \quad + \sqrt{2((N-n)\|f\|_\infty + \|g\|_\infty) \left( 2\varepsilon_{\text{HN},n}^{\text{esti}} + \varepsilon_{\text{HN},n}^{\text{approx}} \right)}.
\end{aligned}$$

We then notice that

$$\begin{aligned}
& \left| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right|^2 \\
(4.34) \quad & \leq 2\|r\|_\infty ((N-n)\|f\|_\infty + \|g\|_\infty) \left| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right|
\end{aligned}$$

holds a.s., so that

$$\begin{aligned}
& \left\| V_n(X_n) - \bar{V}_n^M(X_n) \right\|_{M,2} \\
& \leq \sqrt{2\|r\|_\infty ((N-n)\|f\|_\infty + \|g\|_\infty) \left\| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right\|_{M,1}} \\
& \quad + \sqrt{2((N-n)\|f\|_\infty + \|g\|_\infty) \left( 2\varepsilon_{\text{HN},n}^{\text{esti}} + \varepsilon_{\text{HN},n}^{\text{approx}} \right)},
\end{aligned}$$

and use Lemma 4.17 to bound  $\varepsilon_{\text{HN},n}^{\text{approx}}$ . By plugging into (4.32), and using the estimations in Lemmas 4.15 and 4.17, we obtain the estimate (4.31). Together with (4.29),

this proves the required estimate(4.28). By induction, we get as  $M \rightarrow \infty$ ,

$$\begin{aligned} \mathbb{E}_M \left[ |\hat{V}_n^M(X_n) - V_n(X_n)| \right] &= \mathcal{O}_{\mathbb{P}} \left( \left( \gamma_M^4 \frac{K_M \log(M)}{M} \right)^{\frac{1}{2(N-n)}} + \left( \gamma_M^4 \frac{\rho_M^2 \eta_M^2 \log(M)}{M} \right)^{\frac{1}{4(N-n)}} \right. \\ &\quad + \sup_{n \leq k \leq N} \inf_{\Phi \in \mathcal{V}_M} \left( \mathbb{E}_M \left[ |\Phi(X_k) - V_k(X_k)|^2 \right] \right)^{\frac{1}{2(N-n)}} \\ &\quad \left. + \sup_{n \leq k \leq N} \inf_{A \in \mathcal{A}_M} \left( \mathbb{E} \left[ |A(X_k) - a_k^{\text{opt}}(X_k)| \right] \right)^{\frac{1}{2(N-n)}} \right), \end{aligned}$$

which completes the proof of Theorem 4.13.  $\square$

**5. Conclusion.** This paper develops new machine learning algorithms for high-dimensional Markov decision processes. We propose and compare two algorithms based on dynamic programming, performance/hybrid iteration and neural networks for approximating the control and value function. The main theoretical contribution is to provide a detailed convergence analysis for each of these algorithms: by using least squares neural network regression, we state error estimates in terms of the universal approximation error of neural networks, and of the statistical risk when estimating the network function. Numerical tests on various applications are presented in a companion paper [3].

#### Appendix A. 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  $\|\cdot\|_d$  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

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

for all Borelian  $O$  in  $\bar{B}_d^{\mathcal{X}}(0, R)$ , where  $\pi_R$  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  $\mu$ . 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 with the following stochastic control problem for  $(\bar{X}_n)_{n=0}^N$ :

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

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:

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

We shall assume two conditions on the measure  $\mu$ .

**(Hloc)**  $\mu$  is such that:

$$\mathbb{E}[\lvert \pi_R(X) - X \rvert] \xrightarrow{R \rightarrow \infty} 0 \quad \text{and} \quad \mathbb{P}(\lvert X \rvert > R) \xrightarrow{R \rightarrow \infty} 0, \quad \text{where } X \sim \mu.$$

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

PROPOSITION A.1. *Let  $X_n \sim \mu$ . The following holds:*

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

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  $n = 0, \dots, N$ , we get under **(Hloc)**:

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

**Comment:** Proposition A.1 states that if  $\mathcal{X}$  is not bounded, the control problem (A.1) associated with 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:

$$\begin{aligned} \mathbb{E} \left[ \left| \bar{V}_n^R(\pi_R(X_n)) - V_n(X_n) \right| \right] &\leq \mathbb{E} \left[ \left| \bar{V}_n^R(X_n) - V_n(X_n) \right| \mathbf{1}_{\lvert X_n \rvert \leq R} \right] \\ \text{(A.2)} \quad &+ \mathbb{E} \left[ \left| \bar{V}_n^R(\pi_R(X_n)) - V_n(X_n) \right| \mathbf{1}_{\lvert X_n \rvert \geq R} \right]. \end{aligned}$$

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

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

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

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

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

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

holds a.s., which implies:

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

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

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

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

### Appendix B. 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 with  $(a_n^{\mathcal{A}_M})_{n=0}^{N-1}$  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 B.1.** *Assume that there exists a sequence of optimal feedback controls  $(a_n^{\text{opt}})_{0 \leq n \leq N-1}$  for the control problem with value function  $V_n$ ,  $n = 0, \dots, N$ . Then the following holds, as  $M \rightarrow \infty$ :*

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

*Remark B.2.* Notice that there is no estimation error term in (B.1), 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.  $\square$

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



$\min_{A \in \mathcal{A}_M} J_n^A$ . Moreover:

$$\begin{aligned}
\mathbb{E} \left[ J_n^A(X_n) - V_n(X_n) \right] &\leq \mathbb{E} \left[ |f(X_n, A(X_n)) - f(X_n, a_n^{\text{opt}}(X_n))| \right] \\
&\quad + \mathbb{E} \left[ |J_{n+1}^{A_M}(F(X_n, A(X_n), \varepsilon_{n+1})) - V_{n+1}(F(X_n, a_n^{\text{opt}}(X_n), \varepsilon_{n+1}))| \right] \\
&\leq [f]_L \mathbb{E} \left[ |a_n^{\text{opt}}(X_n) - A(X_n)| \right] \\
&\quad + \mathbb{E} \left[ |V_{n+1}(F(X_n, A(X_n), \varepsilon_{n+1})) - V_{n+1}(F(X_n, a_n^{\text{opt}}(X_n), \varepsilon_{n+1}))| \right] \\
\text{(B.2)} \quad &\quad + \mathbb{E} \left[ |J_{n+1}^{A_M}(F(X_n, A(X_n), \varepsilon_{n+1})) - V_{n+1}(F(X_n, A(X_n), \varepsilon_{n+1}))| \right].
\end{aligned}$$

Applying assumption **(Hd)** to bound the last term in the r.h.s. of (B.2) yields

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

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

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

(B.1) then follows directly by induction.  $\square$

### Appendix C. Proof of Lemma 4.10.

The proof is divided into four steps.

*Step 1: Symmetrization by a ghost sample.* We take  $\varepsilon > 0$  and show that for

$M > 2 \frac{((N-n)\|f\|_\infty + \|g\|_\infty)^2}{\varepsilon^2}$ , the following holds:

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

where:

- $(X_k^{\prime(m)})_{1 \leq m \leq M, n \leq k \leq N}$  is a copy of  $(X_k^{(m)})_{1 \leq m \leq M, n \leq k \leq N}$  generated from an independent copy of the exogenous noises  $(\varepsilon_k^{\prime(m)})_{1 \leq m \leq M, n \leq k \leq N}$ , and independent copy of initial positions at time  $n$ ,  $(X_n^{\prime(m)})_{m=1}^M$ , following the same control  $\hat{a}_k^M$  at time  $k = n+1, \dots, N-1$ , and control  $A$  at time  $n$ ,
- We recall that  $Y_{n+1}^{(m),A}$  has already been defined in (4.6), and we similarly define

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



by the training set and injecting (C.2) yields

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

for  $M > 2 \frac{((N-n)\|f\|_\infty + \|g\|_\infty)^2}{\varepsilon^2}$ , where we use the definition of  $A^*$  to go from the second-to-last to the last line. The proof of (C.1) is then completed.

*Step 2:* We show that

$$\begin{aligned}
& \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] - \mathbb{E} \left[ J_n^{A, (\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n) \right] \right| \right] \\
\leq & 4\mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} - f(X_n'^{(m)}, A(X_n'^{(m)})) - \hat{Y}_{n+1}'^{(m),A} \right] \right| \right] \\
& + \mathcal{O} \left( \frac{1}{\sqrt{M}} \right).
\end{aligned} \tag{C.3}$$

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

$$\begin{aligned}
& \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] - \mathbb{E} \left[ J_n^{A, (\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n) \right] \right| \right] \\
& = \int_0^\infty \mathbb{P} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] - \mathbb{E} \left[ J_n^{A, (\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n) \right] \right| > \varepsilon \right] d\varepsilon \\
& = \int_0^{M'} \mathbb{P} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] - \mathbb{E} \left[ J_n^{A, (\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n) \right] \right| > \varepsilon \right] d\varepsilon \\
& \quad + \int_{M'}^\infty \mathbb{P} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] - \mathbb{E} \left[ J_n^{A, (\hat{a}_k^M)_{k=n+1}^{N-1}}(X_n) \right] \right| > \varepsilon \right] d\varepsilon \\
& \leq \sqrt{2} \frac{(N-n)\|f\|_\infty + \|g\|_\infty}{\sqrt{M}} \\
& \quad + 4 \int_0^\infty \mathbb{P} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] \right. \right. \\
& \quad \left. \left. - f(X_n'^{(m)}, A(X_n'^{(m)})) - \hat{Y}_{n+1}'^{(m),A} \right] \right| > \varepsilon \right] d\varepsilon.
\end{aligned} \tag{C.4}$$

The second term in the r.h.s. of (C.4) comes from (C.1). It remains to write the latter as an expectation to obtain (C.3).

*Step 3: Introduction of additional randomness by random signs.*

Let  $(r_m)_{1 \leq m \leq M}$  be i.i.d. Rademacher r.v.<sup>d</sup>. We show that:

$$\begin{aligned} & \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} - f(X_n'^{(m)}, A(X_n'^{(m)})) - \hat{Y}_{n+1}'^{(m),A} \right] \right| \right] \\ (C.5) & \leq 4 \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M r_m \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] \right| \right]. \end{aligned}$$

Since for each  $m = 1, \dots, M$  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 corresponding components. Thus, the following holds for  $\varepsilon \geq 0$ :

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

It remains to integrate on  $\mathbb{R}_+$  w.r.t.  $\varepsilon$  to get (C.5).

*Step 4:* We show that

$$\begin{aligned} & \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M r_m \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] \right| \right] \\ & \leq \frac{(N-n) \|f\|_\infty + \|g\|_\infty}{\sqrt{M}} \\ & \quad + \left( [f]_L + [f]_L \sum_{k=n+1}^{N-1} (1 + \eta_M \gamma_M)^{k-n} [F]_L^{k-n} + \eta_M^{N-n} \gamma_M^{N-n} [F]_L^{N-n} [g]_L \right) \\ (C.6) & = \mathcal{O} \left( \frac{\gamma_M^{N-n} \eta_M^{N-n}}{\sqrt{M}} \right), \quad \text{as } M \rightarrow \infty. \end{aligned}$$

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

$$\begin{aligned} & \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M r_m \left( f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right) \right| \right] \\ & \leq \mathbb{E} \left[ \left| \frac{1}{M} \sum_{m=1}^M r_m \left( f(X_n^{(m)}, 0) + \hat{Y}_{n+1}^{(m),0} \right) \right| \right] \\ (C.7) & \quad + \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M r_m \left( f(X_n^{(m)}, A(X_n^{(m)})) - f(X_n^{(m)}, 0) + \hat{Y}_{n+1}^{(m),A} - \hat{Y}_{n+1}^{(m),0} \right) \right| \right]. \end{aligned}$$

<sup>d</sup>The probability mass function of a Rademacher r.v. is by definition  $\frac{1}{2} \delta_{-1} + \frac{1}{2} \delta_1$ .

We now bound the first term of the r.h.s. of (C.7). 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

$$\begin{aligned} \mathbb{E} \left[ \left| \frac{1}{M} \sum_{m=1}^M r_m \left( f(X_n^{(m)}, 0) + \hat{Y}_{n+1}^{(m),0} \right) \right| \right] &\leq \frac{1}{M} \sqrt{\mathbb{E} \left[ \left| \sum_{m=1}^M r_m \left( f(X_n^{(m)}, 0) + \hat{Y}_{n+1}^{(m),0} \right) \right|^2 \right]} \\ (C.8) \qquad \qquad \qquad &\leq \frac{1}{\sqrt{M}} ((N-n)\|f\|_\infty + \|g\|_\infty) \end{aligned}$$

Turn now to the second term of (C.7). By the Lipschitz continuity of  $f$ , it stands:

$$\begin{aligned} \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \sum_{m=1}^M r_m \left( f(X_n^{(m)}, A(X_n^{(m)})) - f(X_n^{(m)}, 0) + \hat{Y}_{n+1}^{(m),A} - \hat{Y}_{n+1}^{(m),0} \right) \right| \right] \\ \leq [f]_L \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \sum_{m=1}^M r_m A(X_n^{(m)}) \right| \right] + \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \sum_{m=1}^M r_m \left( \hat{Y}_{n+1}^{(m),A} - \hat{Y}_{n+1}^{(m),0} \right) \right| \right] \\ \leq \left( [f]_L + [f]_L \sum_{k=n+1}^{N-1} (1 + \eta_M \gamma_M)^{k-n} \mathbb{E} \left[ \sup_{1 \leq m \leq M} \prod_{j=n+1}^k C(\varepsilon_j^m) \right] \right) \\ (C.9) \quad + [g]_L \left( 1 + \eta_M^{N-n} \gamma_M^{N-n} \right) \mathbb{E} \left[ \sup_{1 \leq m \leq M} \prod_{j=n+1}^N C(\varepsilon_j^m) \right] \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \sum_{m=1}^M r_m A(X_n^{(m)}) \right| \right] \end{aligned}$$

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  $k = n+1, \dots, N-1$ . Now, notice first that

$$(C.10) \quad \mathbb{E} \left[ \sup_{1 \leq m \leq M} \prod_{k=n+1}^N C(\varepsilon_k^m) \right] \leq \prod_{k=n+1}^N \mathbb{E} \left[ \sup_{1 \leq m \leq M} C(\varepsilon_k^m) \right] \leq \rho_M^{N-n},$$

and moreover:

$$\begin{aligned} \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \sum_{m=1}^M r_m A(X_n^{(m)}) \right| \right] &\leq \gamma_M \mathbb{E} \left[ \sup_{|v|_2 \leq 1/R} \left| \sum_{m=1}^M r_m (v^T X_n^{(m)})_+ \right| \right] \\ (C.11) \qquad \qquad \qquad &\leq \gamma_M \mathbb{E} \left[ \sup_{|v|_2 \leq 1/R} \left| \sum_{m=1}^M r_m v^T X_n^{(m)} \right| \right], \end{aligned}$$

where  $R > 0$  is a bound for the state space (see e.g. the discussion on the Frank-Wolfe step p.10 of [2] 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  $(r_m)_m$  are i.i.d. Rademacher r.v.. Plug (C.10) and (C.11) into (C.9) to obtain

$$\begin{aligned} \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \sum_{m=1}^M r_m \left( f(X_n^{(m)}, A(X_n^{(m)})) - f(X_n^{(m)}, 0) + \hat{Y}_{n+1}^{(m),A} - \hat{Y}_{n+1}^{(m),0} \right) \right| \right] \\ \leq \left( [f]_L + [f]_L \sum_{k=n+1}^{N-1} (1 + \eta_M \gamma_M)^{k-n} \rho_M^{k-n} + [g]_L \left( 1 + \eta_M^{N-n} \gamma_M^{N-n} \right) \rho_M^{N-n} \right) \gamma_M \sqrt{M}. \end{aligned} \quad (C.12)$$

Plug then (C.8) and (C.12) into (C.7) to get (C.6).

*Step 5: Conclusion.* Plug (C.6) into (C.5) and combine it with (C.3) to obtain the bound on the estimation error, as stated in (4.11) of Lemma 4.10.  $\square$

#### Appendix D. Proof of Lemma 4.11.

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

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

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  $n \in \{1, \dots, N\}$ . The following holds:

$$\begin{aligned} \varepsilon_{\text{PI},n}^{\text{approx}} &:= \inf_{A \in \mathcal{A}_M} \mathbb{E}_M [J_n^{A, (\hat{a})_{k=n+1}^{N-1}}(X_n)] - \inf_{A \in \mathbb{A}^x} \mathbb{E}_M [J_n^{A, (\hat{a})_{k=n+1}^{N-1}}(X_n)] \\ &= \inf_{A \in \mathcal{A}_M} \mathbb{E}_M [J_n^{A, (\hat{a})_{k=n+1}^{N-1}}(X_n)] - \mathbb{E}[V_n(X_n)] \\ &\quad + \mathbb{E}[V_n(X_n)] - \inf_{A \in \mathbb{A}^x} \mathbb{E}_M [J_n^{A, (\hat{a})_{k=n+1}^{N-1}}(X_n)] \\ \text{(D.1)} \quad &\leq \inf_{A \in \mathcal{A}_M} \mathbb{E}_M [J_n^{A, (\hat{a})_{k=n+1}^{N-1}}(X_n)] - \mathbb{E}[V_n(X_n)], \end{aligned}$$

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

$$\begin{aligned} \min_{A \in \mathcal{A}_M} \mathbb{E}_M [J_n^{A, (\hat{a})_{k=n+1}^{N-1}}(X_n)] - \mathbb{E}[V_n(X_n)] \\ \leq \inf_{A \in \mathcal{A}_M} \mathbb{E}_M [f(X_n, A(X_n)) + \mathbb{E}_n^A [J_{n+1}^{(\hat{a})_{k=n+1}^{N-1}}(X_{n+1})]] \\ \text{(D.2)} \quad - \mathbb{E} [f(X_n, a_n^{\text{opt}}(X_n)) + \mathbb{E}_n^{a^{\text{opt}}} [V_{n+1}(X_{n+1})]]. \end{aligned}$$

To bound the r.h.s. of (D.2), first observe that for  $A \in \mathcal{A}_M$ :

$$\begin{aligned} \mathbb{E}_M [f(X_n, A(X_n)) + \mathbb{E}_n^A [J_{n+1}^{(\hat{a})_{k=n+1}^{N-1}}(X_{n+1})]] \\ - \mathbb{E} [f(X_n, a_n^{\text{opt}}(X_n)) + \mathbb{E}_n^{a^{\text{opt}}} [V_{n+1}(X_{n+1})]] \\ \leq \mathbb{E} [|f(X_n, A(X_n)) - f(X_n, a_n^{\text{opt}}(X_n))|] \\ + \mathbb{E}_M [\mathbb{E}_n^A J_{n+1}^{(\hat{a})_{k=n+1}^{N-1}}(X_{n+1}) - \mathbb{E}_n^{a^{\text{opt}}} V_{n+1}(X_{n+1})] \\ \leq ([f]_L + \|V_{n+1}\|_\infty [r]_L) \mathbb{E} [|A(X_n) - a_n^{\text{opt}}(X_n)|] \\ \text{(D.3)} \quad + \|r\|_\infty \mathbb{E}_M [J_{n+1}^{(\hat{a})_{k=n+1}^{N-1}}(X_{n+1}) - V_{n+1}(X_{n+1})], \end{aligned}$$

where we used twice assumption **(Hd)** at the second-last line of **(D.3)**. Inject

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

into **(D.3)** to obtain:

$$\begin{aligned} & \mathbb{E}_M \left[ f(X_n, A(X_n)) + \mathbb{E}_n^A \left[ J_{n+1}^{(\hat{a})^{N-1}}(X_{n+1}) \right] \right] \\ & \quad - \mathbb{E} \left[ f(X_n, a_n^{\text{opt}}(X_n)) + \mathbb{E}_n^{a^{\text{opt}}} [V_{n+1}(X_{n+1})] \right] \\ & \leq ([f]_L + \|V_{n+1}\|_\infty [r]_L) \mathbb{E} [|A(X_n) - a_n^{\text{opt}}(X_n)|] \\ (D.4) \quad & + \|r\|_\infty \inf_{A \in \mathcal{A}_M} \mathbb{E}_M \left[ J_{n+1}^{A, (\hat{a})^{N-1}}(X_{n+1}) - V_{n+1}(X_{n+1}) \right] + 2\|r\|_\infty \varepsilon_{n+1}^{esti}. \end{aligned}$$

Plugging **(D.4)** into **(D.2)** yields

$$\begin{aligned} & \inf_{A \in \mathcal{A}_M} \mathbb{E}_M \left[ J_n^{A, (\hat{a})^{N-1}}(X_n) \right] - \mathbb{E} [V_n(X_n)] \\ & \leq \|r\|_\infty \inf_{A \in \mathcal{A}_M} \mathbb{E}_M \left[ J_{n+1}^{A, (\hat{a})^{N-1}}(X_{n+1}) - V_{n+1}(X_{n+1}) \right] + 2\|r\|_\infty \varepsilon_{n+1}^{esti} \\ & \quad + ([f]_L + \|V_{n+1}\|_\infty [r]_L) \inf_{A \in \mathcal{A}_M} \mathbb{E} [|A(X_n) - a_n^{\text{opt}}(X_n)|], \end{aligned}$$

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

$$\begin{aligned} & \mathbb{E} \left[ \inf_{A \in \mathcal{A}_M} \mathbb{E}_M \left[ J_n^{A, (\hat{a})^{N-1}}(X_n) \right] - \mathbb{E} [V_n(X_n)] \right] \\ & = \mathcal{O} \left( \sup_{n+1 \leq k \leq N-1} \mathbb{E} [\varepsilon_k^{esti}] + \sup_{n \leq k \leq N-1} \inf_{A \in \mathcal{A}_M} \mathbb{E} [|A(X_n) - a_n^{\text{opt}}(X_n)|] \right). \end{aligned}$$

We now use Lemma 4.10 to bound the expectations of the  $\varepsilon_{\text{PI},k}^{\text{esti}}$  for  $k = n+1, \dots, N-1$ , and plug the result into **(D.1)** to complete the proof of Lemma 4.11.  $\square$

### Appendix E. Function approximation by neural networks.

We assume  $a_k^{\text{opt}} \in \mathbb{L}^2(\mu)$ , and show the relation (4.9) in Proposition 4.8.

The universal approximator theorem applies for  $\mathcal{A}_\infty := \bigcup_{M=1}^\infty \mathcal{A}_M$ , and states that for all  $\varepsilon > 0$ , for all  $k$ , there exists a neural network  $a^*$  in  $\mathcal{A}_\infty$  such that:  $\|a_k^{\text{opt}} - a^*\|_\infty < \frac{\varepsilon}{\mathcal{V}_d(\mathcal{X})}$ , 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:  $\sup_{n \leq k \leq N-1} \int_{\mathcal{X}} |a_k^{\text{opt}}(x) - a^*(x)| d\mu(x) < \varepsilon$ . Also, notice that  $(\mathcal{A}_M)_{M \geq 1}$  is increasing, which implies that  $\mathcal{A}_\infty = \lim_{M \rightarrow +\infty} \mathcal{A}_M$  and gives the existence of  $M > 0$  that depends on  $\varepsilon$  such that  $a^* \in \mathcal{A}_M$ .

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

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

which is the required result stated in (4.9).  $\square$

We now show (4.10) of proposition 4.8. As stated in Proposition 6 of Section 4.7 in [2]: we can approximate a  $c$ -Lipschitz function by a  $\gamma_1$ -norm less than  $\gamma_M$  and

uniform error less than  $c \left(\frac{\gamma_M}{c}\right)^{-2d/(d+1)} \log \frac{\gamma_M}{c}$ , and proposition 1 in [2] shows that a function with  $\gamma_1$  less than  $\gamma_M$  may be approximated with  $K_M$  neurons with uniform error  $\gamma_M K_M^{-(d+3)/(2d)}$ .

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

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

### Appendix F. Proof of Lemma 4.15.

We prove Lemma 4.15 in four steps. Since the proof is very similar to the one of Lemma 4.10, we only detail the arguments that are modified.

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

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

where  $(X_n^{\prime(m)})_{m=1}^M$  is a i.i.d. copy of  $(X_n^{(m)})_{m=1}^M$ ;  $(\varepsilon_{n+1}^{\prime m})_{m=1}^M$  is a i.i.d. copy of  $(\varepsilon_{n+1}^m)_{m=1}^M$ ;  $\hat{Y}_{n+1}^{\prime(m),A} := \hat{V}_{n+1}^M \left( F \left( X_n^{\prime(m)}, A(X_n^{\prime(m)}), \varepsilon_{n+1}^{\prime m} \right) \right)$ ; and finally  $\hat{Y}_{n+1}^{(m),A} := \hat{V}_{n+1}^M \left( F \left( X_n^{(m)}, A(X_n^{(m)}), \varepsilon_{n+1}^m \right) \right)$ .

Since  $\hat{V}_n^M$  the estimated value function at time  $n$ , for  $n=0, \dots, N-1$ , is bounded by construction, (F.1) is proved the same as in step 1 of Lemma 4.10.  $\square$

*Step 2:* The following result holds

$$(F.2) \quad \begin{aligned} & \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] - \mathbb{E} \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] \right| \right] \\ & \leq 4 \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} - f(X_n^{\prime(m)}, A(X_n^{\prime(m)})) - \hat{Y}_{n+1}^{\prime(m),A} \right] \right| \right] \\ & + \mathcal{O} \left( \frac{1}{\sqrt{M}} \right), \end{aligned}$$

and it is proved the same way as step 2 in the proof of Lemma 4.10.  $\square$

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

$$(F.3) \quad \begin{aligned} & \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} - f(X_n^{\prime(m)}, A(X_n^{\prime(m)})) - \hat{Y}_{n+1}^{\prime(m),A} \right] \right| \right] \\ & \leq 4 \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M r_m \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] \right| \right], \end{aligned}$$

as proved already in step 3 in the proof of Lemma 4.10.  $\square$



*Step 4:* We show that

$$\begin{aligned}
& \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M r_m \left[ f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right] \right| \right] \\
& \leq \frac{(N-n)\|f\|_\infty + \|g\|_\infty}{\sqrt{M}} + ([f]_L + \rho_M \gamma_M \eta_M) \frac{\gamma_M}{\sqrt{M}} \\
\text{(F.4)} \quad & = \mathcal{O} \left( \frac{\rho_M \gamma_M^2 \eta_M}{\sqrt{M}} \right), \quad \text{as } M \rightarrow +\infty.
\end{aligned}$$

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

$$\begin{aligned}
\text{(F.5)} \quad & \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M r_m \left( f(X_n^{(m)}, A(X_n^{(m)})) + \hat{Y}_{n+1}^{(m),A} \right) \right| \right] \\
& \leq \mathbb{E} \left[ \left| \frac{1}{M} \sum_{m=1}^M r_m \left( f(X_n^{(m)}, 0) + \hat{Y}_{n+1}^{(m),0} \right) \right| \right] \\
& \quad + \mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \frac{1}{M} \sum_{m=1}^M r_m \left( f(X_n^{(m)}, A(X_n^{(m)})) - f(X_n^{(m)}, 0) + \hat{Y}_{n+1}^{(m),A} - \hat{Y}_{n+1}^{(m),0} \right) \right| \right].
\end{aligned}$$

The first term in the r.h.s. in (F.5) is bounded as in the proof of Lemma 4.10 by  $\frac{(N-n)\|f\|_\infty + \|g\|_\infty}{\sqrt{M}}$ . We use the Lipschitz-continuity of  $f$  as follows, to bound its second term:

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

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

By using the same arguments as those presented to prove Lemma 4.10, we have  $\mathbb{E} \left[ \sup_{A \in \mathcal{A}_M} \left| \sum_{m=1}^M r_m A(X_n^{(m)}) \right| \right] \leq \gamma_M \sqrt{M}$ , and then conclude that (F.4) holds.

*Step 5: Conclusion*

Combining (F.2), (F.3) and (F.4) results in the bound on the estimation error as stated in (4.24).  $\square$

### Appendix G. Proof of Lemma 4.17.

We divide the proof of Lemma 4.17 into two steps. First write

$$\begin{aligned}
\varepsilon_{\text{HN},n}^{\text{approx}} & \leq \inf_{A \in \mathcal{A}_M} \mathbb{E}_M \left[ f(X_n, A(X_n)) + \hat{V}_{n+1}^M(X_{n+1}^A) \right] - \mathbb{E}[V_n(X_n)] \\
\text{(G.1)} \quad & \quad + \mathbb{E}[V_n(X_n)] - \inf_{A \in \mathbb{A}^x} \mathbb{E}_M \left[ f(X_n, A(X_n)) + \hat{V}_{n+1}^M(X_{n+1}^A) \right].
\end{aligned}$$

*Step 1:* We show

$$(G.2) \quad \begin{aligned} & \inf_{A \in \mathcal{A}_M} \mathbb{E}_M \left[ f(X_n, A(X_n)) + \hat{V}_{n+1}^M(X_{n+1}^A) \right] - \mathbb{E}[V_n(X_n)] \\ & \leq ([f]_L + \|V_{n+1}\|_\infty [r]_L) \inf_{A \in \mathbb{A}^x} \mathbb{E}_M \left[ |A(X_n) - a_n^{\text{opt}}(X_n)| \right] \\ & \quad + \|r\|_\infty \mathbb{E}_M \left[ \left| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right| \right]. \end{aligned}$$

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

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

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

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

which is the result stated in **(G.2)**.

*Step 2:* We show

$$(G.3) \quad \begin{aligned} & \mathbb{E}[V_n(X_n)] - \inf_{A \in \mathbb{A}^x} \mathbb{E}_M \left[ f(X_n, A(X_n)) + \hat{V}_{n+1}^M(X_{n+1}^A) \right] \\ & \leq \|r\|_\infty \mathbb{E}_M \left[ \left| V_{n+1}(X_{n+1}) - \hat{V}_{n+1}^M(X_{n+1}) \right| \right]. \end{aligned}$$

Write

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

which completes the proof of **(G.3)**.

*Step 3 Conclusion:*

We complete the proof of Lemma 4.17 by plugging **(G.2)** and **(G.3)** into **(G.1)**.  $\square$

### Appendix H. Some useful Lemmas for the proof of Theorem 4.13.

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

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

An  $\varepsilon$ -cover of  $\mathcal{V}$  is a set of functions  $f_1, \dots, f_P : \mathbb{R}^d \rightarrow \mathbb{R}$  such that

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

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

LEMMA H.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*

$$\max_{m=1, \dots, M} c^2 \mathbb{E} \left[ e^{(Y - m(X))^2 / c^2} - 1 | X \right] \leq \sigma^2 \quad \text{a.s. for some } c, \sigma > 0. \text{ Let } \gamma_M, L \geq 1 \text{ and}$$

*assume that the regression function is bounded by  $L$  and that  $\gamma_M \xrightarrow{M \rightarrow +\infty} +\infty$ . Set*

$$\hat{m}_M := \operatorname{argmin}_{\Phi \in \mathcal{V}_M} \frac{1}{M} \sum_{m=1}^M |\Phi(x_m) - Y_m|^2$$

*for some  $\mathcal{V}_M$  of functions  $\Phi : \mathbb{R}^d \rightarrow [-\gamma_M, \gamma_M]$  and some random variables  $Y_1, \dots, Y_M$  which are bounded by  $L$ , and denote*

$$\Omega_g := \left\{ f - g : f \in \mathcal{V}_M, \frac{1}{M} \sum_{m=1}^M |f(x_m) - g(x_m)|^2 \leq \frac{\delta}{\gamma_M^2} \right\}.$$

*Then there exists constants  $c_1, c_2 > 0$  which depend only on  $\sigma$  and  $c$  such that for any  $\delta_M > 0$  with*

$$\delta_M \xrightarrow{M \rightarrow +\infty} 0, \quad \frac{M\delta_M}{\gamma_M} \xrightarrow{M \rightarrow +\infty} +\infty,$$

$$(H.1) \quad c_1 \frac{\sqrt{M}\delta}{\gamma_M^2} \geq \int_{c_2\delta/\gamma_M^2}^{\sqrt{\delta}} \log \left( \mathcal{N}_2 \left( \frac{u}{4\gamma_M}, \Omega_g, x_1^M \right) \right)^{1/2} du$$

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

$$\mathbb{E} \left[ |\hat{m}_M(X) - m(X)|^2 \right] = \mathcal{O}_{\mathbb{P}} \left( \delta_M + \inf_{\Phi \in \mathcal{V}_M} \mathbb{E} \left[ |\Phi(X) - m(X)|^2 \right] \right).$$

*Proof.* We refer to [16] (see its Theorem 3) for a proof. □

LEMMA H.2. *For any  $\varepsilon > 0$ , we have*

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

*where the class of neural networks  $\mathcal{V}_M$  is defined in Section 4.2.*

*Proof.* We refer to [17] for a proof. □

## REFERENCES

- [1] K. ASADI AND M. LITTMAN, *An alternative softmax operator for reinforcement learning*, in Proceedings of the 34th International Conference on Machine Learning, ICML'17, vol. 70, 2017, pp. 243–252.
- [2] F. BACH, *Breaking the curse of dimensionality with convex neural networks*, Journal of Machine Learning Research, 18 (2017), pp. 1–53.
- [3] A. BACHOUCH, C. HURÉ, N. LANGRENÉ, AND H. PHAM, *Deep neural networks algorithms for stochastic control problems on finite horizon: numerical applications*, Methodology and Computing in Applied Probability, (2021). To appear.
- [4] C. BECK, A. JENTZEN, AND B. KUCKUCK, *Full error analysis for the training of deep neural networks*, arXiv:1910.00121v2, (2020).
- [5] B. BERCU AND J. FORT, *Generic stochastic gradient methods*, in Wiley Encyclopedia of Operations Research and Management Science, 2011, pp. 1–8.
- [6] D. P. BERTSEKAS AND J. TSITSIKLIS, *Neuro-Dynamic Programming*, Athena Scientific, 1996.
- [7] G. CYBENKO, *Approximations by superpositions of sigmoidal functions*, Mathematics of Control, Signals, and Systems, 2 (1989), pp. 303–314.
- [8] W. E, J. HAN, AND A. 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 (2017), pp. 349–380.
- [9] A. GÉRON, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly Media, 2nd ed., 2019.
- [10] L. GYÖRFI, M. KOHLER, A. KRZYŻAK, AND H. WALK, *A Distribution-Free Theory of Nonparametric Regression*, Springer Series in Statistics, 2002.
- [11] J. HAN AND W. E, *Deep learning approximation for stochastic control problems*, in NIPS 2016, Deep Reinforcement Learning Workshop, 2016.
- [12] J. HAN AND J. LONG, *Convergence of the deep BSDE method for coupled FBSDEs*, Probability, Uncertainty and Quantitative Risk, 5 (2020), pp. 1–33.
- [13] P. HENRY-LABORDÈRE, *Deep primal-dual algorithm for BSDEs: Applications of machine learning to CVA and IM*, SSRN:3071506, (2017).
- [14] K. HORNICK, *Approximation capabilities of multilayer feedforward networks*, Neural Networks,, 4 (1991), pp. 251–257.
- [15] A. JENTZEN, B. KUCKUCK, A. NEUFELD, AND P. VON WURSTEMBERGER, *Strong error analysis for stochastic gradient descent optimization algorithms*, arXiv: 1801.09324v1, (2018).
- [16] M. KOHLER, *Nonparametric regression with additional measurement errors in the dependent variable*, Journal of Statistical Planning and Inference, 136 (2006), pp. 3339–3361.
- [17] M. KOHLER, A. KRZYŻAK, AND N. TODOROVIC, *Pricing of high-dimensional American options by neural networks*, Mathematical Finance, 20 (2010), pp. 383–410.
- [18] A. N. 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).
- [19] S. KOU, X. PENG, AND X. XU, *EM algorithm and stochastic control in economics*. SSRN:2865124, 2016.
- [20] Y. LI, *Deep reinforcement learning: an overview*, arXiv 1701.07274v3, (2017).
- [21] F. A. LONGSTAFF AND E. S. SCHWARTZ, *Valuing American options by simulation: A simple least-squares approach*, The Review of Financial Studies, 14 (2001), pp. 113–147.
- [22] V. MNIH, K. KAVUKCUOGLU, D. SILVER, AND A. A. RUSU, *Human-level control through deep reinforcement learning*, Nature, 518 (2015), pp. 529–533.
- [23] W. B. POWELL, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, Wiley & Sons, 2011.
- [24] R. S. SUTTON AND A. G. BARTO, *Reinforcement Learning*, The MIT Press, 1998.